Towards a Unified Theory of Sparsification for Matching Problems

Sepehr Assadi

University of Pennsylvania

Joint work with Aaron Bernstein (Rutgers)

Sparsifiers

A powerful tool when dealing with large graphs is sparsification.

Sparsifiers

A powerful tool when dealing with large graphs is sparsification.

A sparsifier of a graph G is a subgraph H that preserves certain properties of G while having a smaller number of edges.

Sparsifiers

A powerful tool when dealing with large graphs is sparsification.

A sparsifier of a graph G is a subgraph H that preserves certain properties of G while having a smaller number of edges.

Canonical examples:

- Cut sparsifiers: preserve cut-value between bi-partitions [Karger, 1994, Benczúr and Karger, 1996, Fung et al., 2011];
- Spectral sparsifiers: preserve Laplacian spectrum of the graph [Spielman and Teng, 2011, Batson et al., 2009];
- Spanners: preserve pairwise distances [Awerbuch, 1985, Peleg and Schäffer, 1989];
- ...

Sparsifiers

A powerful tool when dealing with large graphs is sparsification.

A sparsifier of a graph G is a subgraph H that preserves certain properties of G while having a smaller number of edges.

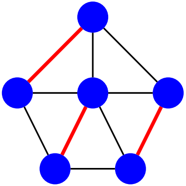
Canonical examples:

- Cut sparsifiers: preserve cut-value between bi-partitions [Karger, 1994, Benczúr and Karger, 1996, Fung et al., 2011];
- Spectral sparsifiers: preserve Laplacian spectrum of the graph [Spielman and Teng, 2011, Batson et al., 2009];
- Spanners: preserve pairwise distances [Awerbuch, 1985, Peleg and Schäffer, 1989];
- ...

This talk: Are there efficient matching sparsifiers?

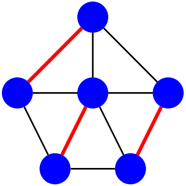
Matching

- Matching: A collection of vertex-disjoint edges.



Matching

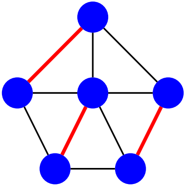
- Matching: A collection of vertex-disjoint edges.



Maximum Matching problem: Find a matching with a largest number of edges.

Matching

- Matching: A collection of vertex-disjoint edges.



Maximum Matching problem: Find a matching with a largest number of edges.

$\mu(G)$: size of a maximum matching in G.

Matching Sparsifier

What is a good definition for a matching sparsifier?

Matching Sparsifier

What is a good definition for a matching sparsifier?

- Preserves the largest matching?
- Preserves large matchings for all subsets of vertices?
- Preserves large matchings over a given set of edges?
- . . .

Matching Sparsifier

What is a good definition for a matching sparsifier?

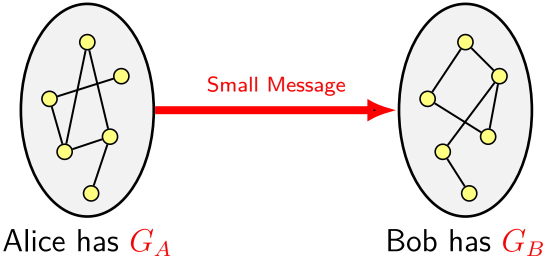
- Preserves the largest matching?
- Preserves large matchings for all subsets of vertices?
- Preserves large matchings over a given set of edges?
- ...

Let us instead consider examples of what we expect from a "good" matching sparsifier in the context of known matching problems.

One-Way Communication Complexity of Matching

Consider the following problem:

- Alice and Bob are given graphs $G_A(V, E_A)$ and $G_B(V, E_B)$.
- Alice wants to send a single message to Bob so Bob can compute a maximum matching of $G_A \cup G_B$.
- What is the tradeoff between length of the message and approximation ratio?



One-Way Communication Complexity of Matching

Consider the following problem:

- Alice and Bob are given graphs $G_A(V, E_A)$ and $G_B(V, E_B)$.
- Alice wants to send a single message to Bob so Bob can compute a maximum matching of $G_A \cup G_B$.
- What is the tradeoff between length of the message and approximation ratio?

Studied by [Goel et al., 2012, Lee and Singla, 2017] owing to its close connection to streaming and online batch-arrival algorithms for maximum matching.

One-Way Communication Complexity of Matching

How does a good sparsifier help here?

One-Way Communication Complexity of Matching

How does a good sparsifier help here?

What would have happened if we were instead interested in communication complexity of minimum or maximum cut problem?

One-Way Communication Complexity of Matching

How does a good sparsifier help here?

What would have happened if we were instead interested in communication complexity of minimum or maximum cut problem?

- Alice sends a cut sparsifier with $\widetilde{O}(n)$ communication.
- Bob outputs a $(1+\varepsilon)$-approximation.

One-Way Communication Complexity of Matching

How does a good sparsifier help here?

What would have happened if we were instead interested in communication complexity of minimum or maximum cut problem?

- Alice sends a cut sparsifier with $\widetilde{O}(n)$ communication.
- Bob outputs a $(1+\varepsilon)$-approximation.

For original maximum matching problem, we would like Alice to be able to send a matching sparsifier!

Other Examples

Stochastic matching problem

Compute a sparse subgraph H of G such that random subgraphs of H have a large matching compared to random subgraphs of G.

Other Examples

Stochastic matching problem

Compute a sparse subgraph H of G such that random subgraphs of H have a large matching compared to random subgraphs of G.

Studied primarily owing to its connection to kidney exchange problem [Blum et al., 2015, Assadi et al., 2016, Assadi et al., 2017, Behnezhad and Reyhani, 2018, Yamaguchi and Maehara, 2018, Behnezhad et al., 2019]

Other Examples

Stochastic matching problem

Compute a sparse subgraph H of G such that random subgraphs of H have a large matching compared to random subgraphs of G.

Studied primarily owing to its connection to kidney exchange problem [Blum et al., 2015, Assadi et al., 2016, Assadi et al., 2017, Behnezhad and Reyhani, 2018, Yamaguchi and Maehara, 2018, Behnezhad et al., 2019]

Fault tolerant matching problem

Fault-tolerant subgraphs studied extensively for spanners and distance preservers. [Chechik et al., 2009, Peleg, 2009, Baswana et al., 2016, Bodwin et al., 2017, Bodwin et al., 2018] $\cdots$

Our Proposed Matching Sparsifier

Our Proposed Matching Sparsifier

We propose edge degree constrained subgraphs as a candidate for a matching sparsifier.

Our Proposed Matching Sparsifier

We propose edge degree constrained subgraphs as a candidate for a matching sparsifier.

Introduced originally by [Bernstein and Stein, 2015] [Bernstein and Stein, 2016] in context of dynamic graph algorithms.

Our Proposed Matching Sparsifier

We propose edge degree constrained subgraphs as a candidate for a matching sparsifier.

Introduced originally by [Bernstein and Stein, 2015] [Bernstein and Stein, 2016] in context of dynamic graph algorithms.

Very recently also used to design randomized composable coresets for matching [Assadi et al., 2019].

Edge Degree Constrained Subgraphs

Definition ([Bernstein and Stein, 2015])

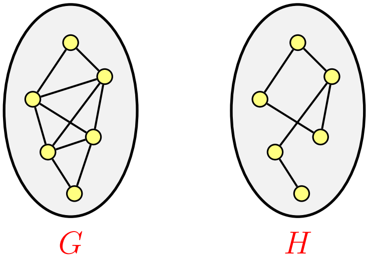
For any $\varepsilon \in (0, 1)$ and $\beta \geq 1$,

Edge Degree Constrained Subgraphs

Definition ([Bernstein and Stein, 2015])

For any $\varepsilon \in (0,1)$ and $\beta \geq 1$,

A subgraph H of G is called a (β, ε)-EDCS of G:



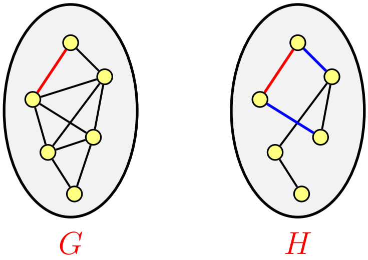
Edge Degree Constrained Subgraphs

Definition ([Bernstein and Stein, 2015])

For any $\varepsilon \in (0,1)$ and $\beta \geq 1$,

A subgraph H of G is called a (β, ε)-EDCS of G:

1. $\forall (u,v) \in H \qquad d_H(u) + d_H(v) \leq \beta$,



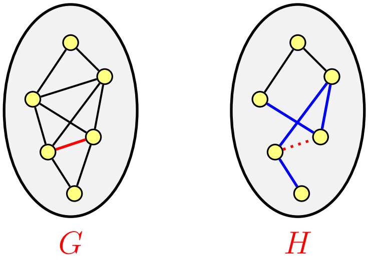
Edge Degree Constrained Subgraphs

Definition ([Bernstein and Stein, 2015])

For any $\varepsilon \in (0,1)$ and $\beta \geq 1$,

A subgraph H of G is called a (β,ε)-EDCS of G:

1. $\forall (u,v) \in H \qquad d_H(u) + d_H(v) \leq \beta$,
2. $\forall (u,v) \in G \setminus H \qquad d_H(u) + d_H(v) \geq (1-\varepsilon) \cdot \beta$.



EDCS as a Matching Sparsifier

Basic properties:

- A (β, ε)-EDCS has $O(n\beta)$ edges.
- Every graph admits a (β, ε)-EDCS for all $\varepsilon \in (0, 1)$ and $\beta > 1/\varepsilon$.

EDCS as a Matching Sparsifier

Basic properties:

- A (β, ε)-EDCS has $O(n\beta)$ edges.
- Every graph admits a (β, ε)-EDCS for all $\varepsilon \in (0, 1)$ and $\beta > 1/\varepsilon$.

EDCS contains a large matching:

A (β, ε)-EDCS contains a $(1.5 + \varepsilon)$-approximate matching for $\beta > \frac{1}{\varepsilon^3}$ [Bernstein and Stein, 2016].

EDCS as a Matching Sparsifier

Basic properties:

- A (β, ε)-EDCS has $O(n\beta)$ edges.
- Every graph admits a (β, ε)-EDCS for all $\varepsilon \in (0,1)$ and $\beta > 1/\varepsilon$.

EDCS contains a large matching:

A (β, ε)-EDCS contains a $(1.5+\varepsilon)$-approximate matching for $\beta > \frac{1}{\varepsilon^3}$ [Bernstein and Stein, 2016].

[This work]:

> *An EDCS can act as a robust matching sparsifier under different notions of sparsification.*

Our Results

One-way communication complexity of matching:
$(1.5 + \varepsilon)$-approximation with $O_\varepsilon(n)$ communication.

Our Results

One-way communication complexity of matching:
$(1.5+\varepsilon)$-approximation with $O_\varepsilon(n)$ communication.

- 1.5-approximation for bipartite graphs [Goel et al., 2012].
- 1.66-approximation for general graphs [Lee and Singla, 2017].

Our Results

One-way communication complexity of matching:
$(1.5 + \varepsilon)$-approximation with $O_\varepsilon(n)$ communication.

- 1.5-approximation for bipartite graphs [Goel et al., 2012].
- 1.66-approximation for general graphs [Lee and Singla, 2017].

Stochastic matching problem:
$(1.5 + \varepsilon)$-approximation with max-degree $O_\varepsilon\left(\frac{\log(1/p)}{p}\right)$.

Our Results

One-way communication complexity of matching:
$(1.5 + \varepsilon)$-approximation with $O_\varepsilon(n)$ communication.

- 1.5-approximation for bipartite graphs [Goel et al., 2012].
- 1.66-approximation for general graphs [Lee and Singla, 2017].

Stochastic matching problem:
$(1.5 + \varepsilon)$-approximation with max-degree $O_\varepsilon\left(\frac{\log(1/p)}{p}\right)$.

- $(2 + \varepsilon)$-approximation [Blum et al., 2015, Assadi et al., 2016].
- 1.999-approximation [Assadi et al., 2017].
- 1.52-approximation [Behnezhad et al., 2019] (parallel work).

Our Results

One-way communication complexity of matching:
$(1.5+\varepsilon)$-approximation with $O_\varepsilon(n)$ communication.

- 1.5-approximation for bipartite graphs [Goel et al., 2012].
- 1.66-approximation for general graphs [Lee and Singla, 2017].

Stochastic matching problem:
$(1.5+\varepsilon)$-approximation with max-degree $O_\varepsilon\left(\frac{\log(1/p)}{p}\right)$.

- $(2+\varepsilon)$-approximation [Blum et al., 2015, Assadi et al., 2016].
- 1.999-approximation [Assadi et al., 2017].
- 1.52-approximation [Behnezhad et al., 2019] (parallel work).

f-fault-tolerant matching problem:
$(1.5+\varepsilon)$-approximation with $O_\varepsilon(n+f)$ edges.

Our Results

Problem? the proof of matching preserving property of EDCS (for general graphs) in [Bernstein and Stein, 2016] is not simple.

Our Results

Problem? the proof of matching preserving property of EDCS (for general graphs) in [Bernstein and Stein, 2016] is not simple.

We give a significantly simpler proof of this property with even slightly improved parameters.

Our Results

Problem? the proof of matching preserving property of EDCS (for general graphs) in [Bernstein and Stein, 2016] is not simple.

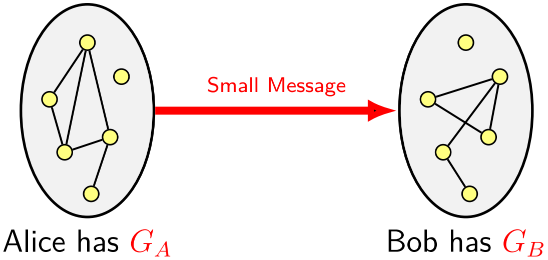
We give a significantly simpler proof of this property with even slightly improved parameters.

(β, ε)-EDCS contains a $(1.5+\varepsilon)$-approximate matching for $\beta \gtrsim \frac{1}{\varepsilon^2}$.

EDCS as a Matching Sparsifier

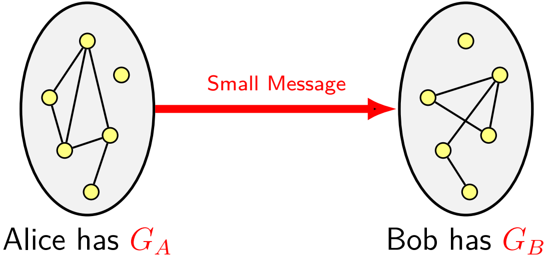
One-Way Communication Complexity of Matching

- Alice and Bob are given graphs $G_A(V, E_A)$ and $G_B(V, E_B)$.
- Alice wants to send a single message to Bob so Bob can compute a maximum matching of $G_A \cup G_B$.



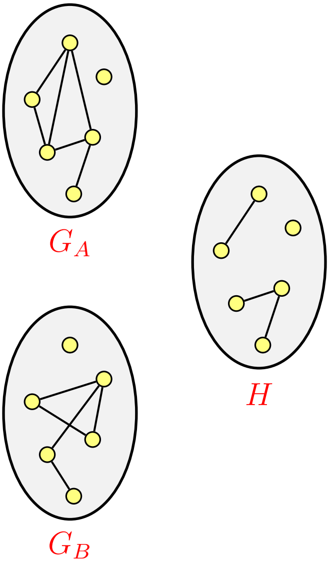
One-Way Communication Complexity of Matching

- Alice and Bob are given graphs $G_A(V, E_A)$ and $G_B(V, E_B)$.
- Alice wants to send a single message to Bob so Bob can compute a maximum matching of $G_A \cup G_B$.



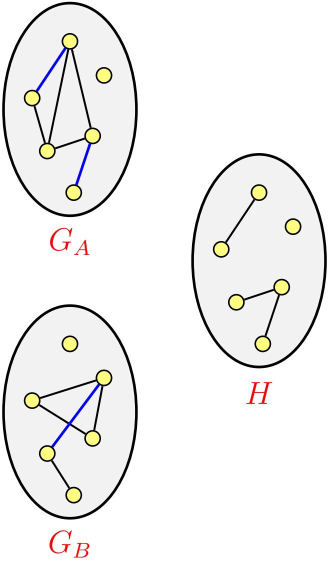
Our solution: Alice sends a $(1/\varepsilon^2, \varepsilon)$-EDCS H of G_A to Bob.

Proof of Correctness



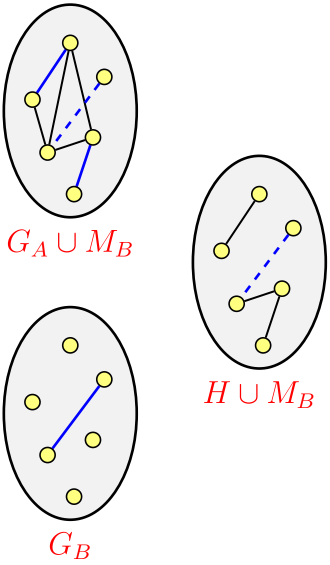
Proof of Correctness

- Fix a maximum matching $M_A \cup M_B$ of $G_A \cup G_B$.



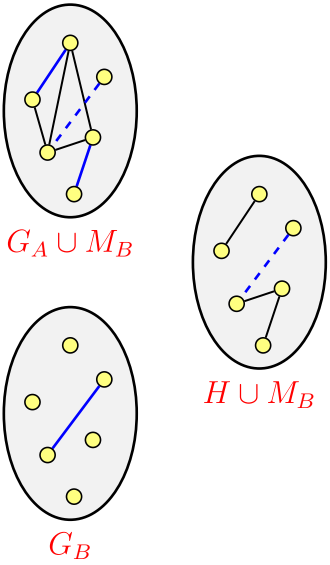
Proof of Correctness

- Fix a maximum matching $M_A \cup M_B$ of $G_A \cup G_B$.
- $H \cup M_B$ contains a $(\beta + 2, 2\varepsilon)$-EDCS of $G_A \cup M_B$:



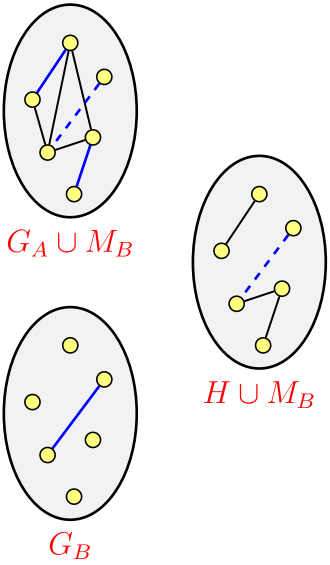
Proof of Correctness

- Fix a maximum matching $M_A \cup M_B$ of $G_A \cup G_B$.
- $H \cup M_B$ contains a $(\beta + 2, 2\varepsilon)$-EDCS of $G_A \cup M_B$:
- **Proof.** Add any edge (u, v) in M_B to H iff $d_H(u) + d_H(v) \leq \beta$.



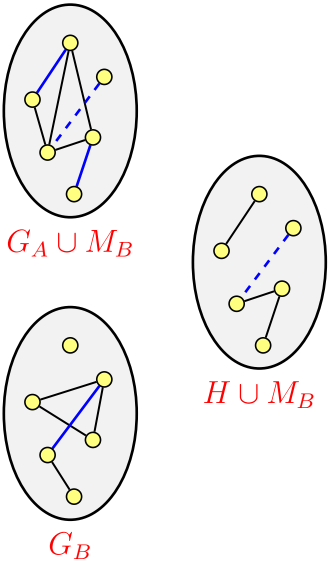
Proof of Correctness

- Fix a maximum matching $M_A \cup M_B$ of $G_A \cup G_B$.
- $H \cup M_B$ contains a $(\beta + 2, 2\varepsilon)$-EDCS of $G_A \cup M_B$:
- **Proof.** Add any edge (u, v) in M_B to H iff $d_H(u) + d_H(v) \leq \beta$.
- So $\mu(H \cup M_B) \geq (2/3 - \varepsilon) \cdot \mu(G_A \cup G_B)$.



Proof of Correctness

- Fix a maximum matching $M_A \cup M_B$ of $G_A \cup G_B$.
- $H \cup M_B$ contains a $(\beta + 2, 2\varepsilon)$-EDCS of $G_A \cup M_B$:
- **Proof.** Add any edge (u, v) in M_B to H iff $d_H(u) + d_H(v) \leq \beta$.
- So $\mu(H \cup M_B) \geq (2/3 - \varepsilon) \cdot \mu(G_A \cup G_B)$.
- $H \cup M_B \subseteq H \cup G_B$ known to Bob.



One-Way Communication Complexity of Matching

One-way communication complexity of matching:
$(1.5 + \varepsilon)$-approximation with $O_{\varepsilon}(n)$ communication.

One-Way Communication Complexity of Matching

One-way communication complexity of matching:
$(1.5 + \varepsilon)$-approximation with $O_\varepsilon(n)$ communication.

Previously,

- 1.5-approximation for bipartite graphs [Goel et al., 2012].
- 1.66-approximation for general graphs [Lee and Singla, 2017].

EDCS Contains a Large Matching

Matching Preserving Property of EDCS

(β, ε)-EDCS contains a $(1.5+\varepsilon)$-approximate matching for $\beta \gtrsim \frac{1}{\varepsilon^2}$.

Matching Preserving Property of EDCS

(β, ε)-EDCS contains a $(1.5+\varepsilon)$-approximate matching for $\beta \gtrsim \frac{1}{\varepsilon^2}$.

Part one: Prove the result for bipartite graphs.

- A simple argument based on Hall's theorem.
- Similar to [Bernstein and Stein, 2015].

Matching Preserving Property of EDCS

(β, ε)-EDCS contains a $(1.5+\varepsilon)$-approximate matching for $\beta \gtrsim \frac{1}{\varepsilon^2}$.

Part one: Prove the result for bipartite graphs.

- A simple argument based on Hall's theorem.
- Similar to [Bernstein and Stein, 2015].

Part two: Reduce the general case to bipartite graphs.

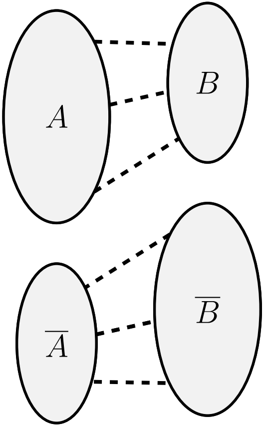
- Uses robustness properties of EDCS that we prove in this paper with a simple application of Lovasz Local Lemma.
- Entirely different from and significantly simpler than [Bernstein and Stein, 2016].

Bipartite Graphs

- H is a (β, ε)-EDCS of G with maximum matching size $\mu(H)$.

Bipartite Graphs

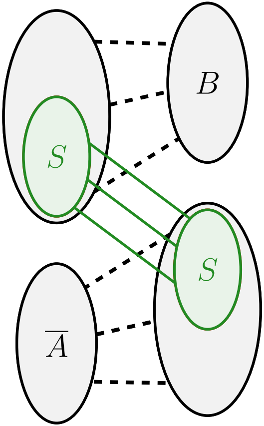
- H is a (β, ε)-EDCS of G with maximum matching size $\mu(H)$.
- Hall's theorem:
  No A to $\overline{B}$ edge;
  $\left|\overline{A} \cup B\right| = \mu(H)$.



EDCS H

Bipartite Graphs

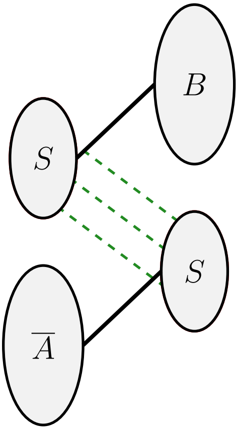
- H is a (β, ε)-EDCS of G with maximum matching size $\mu(H)$.
- Hall's theorem: No A to $\overline{B}$ edge; $\left|\overline{A} \cup B\right| = \mu(H)$.
- G has a matching of size $\mu(G) \geq \mu(H)$: $|S| = 2(\mu(G) - \mu(H))$.



Matching M in G.

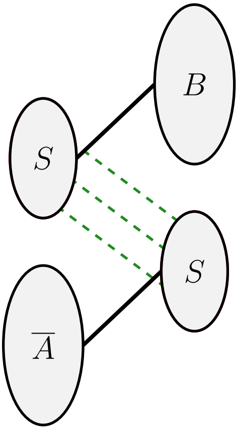
Bipartite Graphs

- H is an EDCS: average degree of S is $\gtrsim_\varepsilon \beta/2$.



Bipartite Graphs

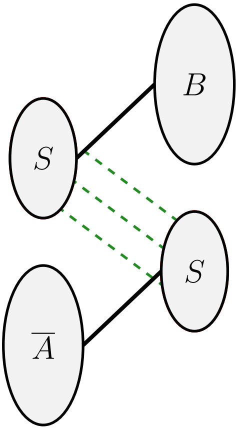
- H is an EDCS: average degree of S is $\gtrsim_\varepsilon \beta/2$.
- H is an EDCS: average degree of $\overline{A} \cup B$ is $\lesssim_\varepsilon \beta/2$.



EDCS H

Bipartite Graphs

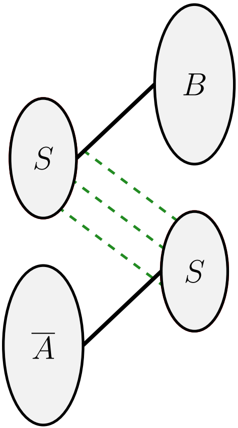
- H is an EDCS: average degree of S is $\gtrsim_\varepsilon \beta/2$.
- H is an EDCS: average degree of $\overline{A} \cup B$ is $\lesssim_\varepsilon \beta/2$.
- $|S| \lesssim_\varepsilon \left|\overline{A} \cup B\right|$: $2(\mu(G) - \mu(H)) \lesssim_\varepsilon \mu(H)$.



EDCS H

Bipartite Graphs

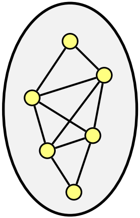
- H is an EDCS: average degree of S is $\gtrsim_\varepsilon \beta/2$.
- H is an EDCS: average degree of $\overline{A} \cup B$ is $\lesssim_\varepsilon \beta/2$.
- $|S| \lesssim_\varepsilon \left|\overline{A} \cup B\right|$: $2(\mu(G) - \mu(H)) \lesssim_\varepsilon \mu(H)$.
- $\mu(H) \gtrsim_\varepsilon 2/3 \cdot \mu(G)$.



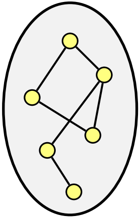
EDCS H

General Graphs

- Fix a (β, ε)-EDCS H of G.



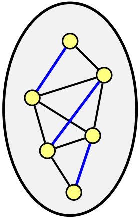
G



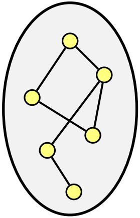
H

General Graphs

- Fix a (β, ε)-EDCS H of G.
- Consider a maximum matching in G.



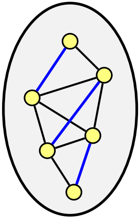
G



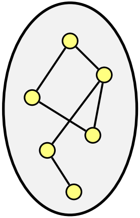
H

General Graphs

- Fix a (β, ε)-EDCS H of G.
- Consider a maximum matching in G.
- Randomly partition vertices of G and H along this matching.



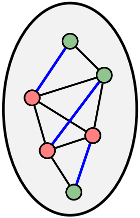
G



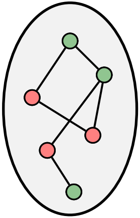
H

General Graphs

- Fix a (β, ε)-EDCS H of G.
- Consider a maximum matching in G.
- Randomly partition vertices of G and H along this matching.



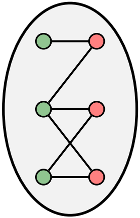
G



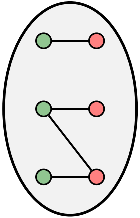
H

General Graphs

- Fix a (β, ε)-EDCS H of G.
- Consider a maximum matching in G.
- Randomly partition vertices of G and H along this matching.



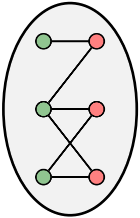
G'



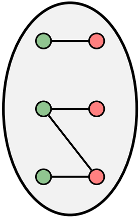
H'

General Graphs

- Fix a (β, ε)-EDCS H of G.
- Consider a maximum matching in G.
- Randomly partition vertices of G and H along this matching.
- $d_{H'}(v) \approx_{\varepsilon} d_H(v)/2$ with constant probability.



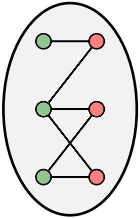
G'



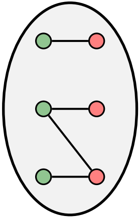
H'

General Graphs

- Fix a (β, ε)-EDCS H of G.
- Consider a maximum matching in G.
- Randomly partition vertices of G and H along this matching.
- $d_{H'}(v) \approx_\varepsilon d_H(v)/2$ with constant probability.
- LLL $\implies$ $\exists$ a random partitioning s.t. $d_{H'}(v) \approx_\varepsilon d_H(v)/2$ for all vertices.



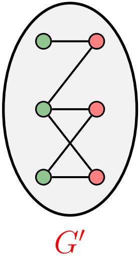
G'

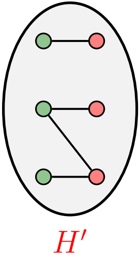


H'

General Graphs

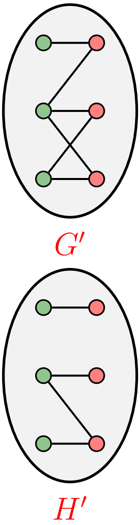
- Fix a (β, ε)-EDCS H of G.
- Consider a maximum matching in G.
- Randomly partition vertices of G and H along this matching.
- $d_{H'}(v) \approx_\varepsilon d_H(v)/2$ with constant probability.
- LLL $\implies$ $\exists$ a random partitioning s.t. $d_{H'}(v) \approx_\varepsilon d_H(v)/2$ for all vertices.
- H' becomes a $(\beta/2, \varepsilon)$-EDCS of G'!





General Graphs

- Fix a (β, ε)-EDCS H of G.
- Consider a maximum matching in G.
- Randomly partition vertices of G and H along this matching.
- $d_{H'}(v) \approx_\varepsilon d_H(v)/2$ with constant probability.
- LLL $\implies$ $\exists$ a random partitioning s.t. $d_{H'}(v) \approx_\varepsilon d_H(v)/2$ for all vertices.
- H' becomes a $(\beta/2, \varepsilon)$-EDCS of G'!
- $(1.5 + \varepsilon)$-approximation by the result on bipartite graphs.



Concluding Remarks

Concluding Remarks

EDCS can act as a matching sparsifier under different notions of sparsification.

Concluding Remarks

EDCS can act as a matching sparsifier under different notions of sparsification.

This gives extremely simple and non-technical proofs for various matching problems in a unified way.

Concluding Remarks

EDCS can act as a matching sparsifier under different notions of sparsification.

This gives extremely simple and non-technical proofs for various matching problems in a unified way.

One-way communication complexity of matching:
$(1.5 + \varepsilon)$-approximation with $O_\varepsilon(n)$ communication.

Stochastic matching problem:
$(1.5 + \varepsilon)$-approximation with max-degree $O_\varepsilon\left(\frac{\log(1/p)}{p}\right)$.

f-fault-tolerant matching problem:
$(1.5 + \varepsilon)$-approximation with $O_\varepsilon(n + f)$ edges.

Concluding Remarks

EDCS can act as a matching sparsifier under different notions of sparsification.

This gives extremely simple and non-technical proofs for various matching problems in a unified way.

One-way communication complexity of matching:
$(1.5 + \varepsilon)$-approximation with $O_\varepsilon(n)$ communication.

Stochastic matching problem:
$(1.5 + \varepsilon)$-approximation with max-degree $O_\varepsilon\left(\frac{\log(1/p)}{p}\right)$.

f-fault-tolerant matching problem:
$(1.5 + \varepsilon)$-approximation with $O_\varepsilon(n + f)$ edges.

Thank you!

Assadi, S., Bateni, M., Bernstein, A., Mirrokni, V. S., and Stein, C. (2019).
Coresets meet EDCS: algorithms for matching and vertex cover on massive graphs.
In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*.

Assadi, S., Khanna, S., and Li, Y. (2016).
The stochastic matching problem with (very) few queries.
In *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 43–60.

Assadi, S., Khanna, S., and Li, Y. (2017).
The stochastic matching problem: Beating half with a non-adaptive algorithm.

In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 99–116.

Awerbuch, B. (1985).
Complexity of network synchronization.
J. ACM, 32(4):804–823.

Baswana, S., Choudhary, K., and Roditty, L. (2016).
Fault tolerant subgraph for single source reachability: generic and optimal.
In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 509–518.

Batson, J. D., Spielman, D. A., and Srivastava, N. (2009).
Twice-ramanujan sparsifiers.

In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 255–262.

Behnezhad, S., Farhadi, A., Hajiaghayi, M., and Reyhani, N. (2019).
Stochastic matching with few queries: New algorithms and tools.
In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*.

Behnezhad, S. and Reyhani, N. (2018).
Almost optimal stochastic weighted matching with few queries.
In *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, pages 235–249.

Benczúr, A. A. and Karger, D. R. (1996).
Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time.

In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 47–55.

Bernstein, A. and Stein, C. (2015).
Fully dynamic matching in bipartite graphs.
In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 167–179.

Bernstein, A. and Stein, C. (2016).
Faster fully dynamic matchings with small approximation ratios.
In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 692–711.

Blum, A., Dickerson, J. P., Haghtalab, N., Procaccia, A. D., Sandholm, T., and Sharma, A. (2015).

Ignorance is almost bliss: Near-optimal stochastic matching with few queries.
In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 325–342.

Bodwin, G., Dinitz, M., Parter, M., and Williams, V. V. (2018).
Optimal vertex fault tolerant spanners (for fixed stretch).
In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1884–1900.

Bodwin, G., Grandoni, F., Parter, M., and Williams, V. V. (2017).
Preserving distances in very faulty graphs.
In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 73:1–73:14.

Chechik, S., Langberg, M., Peleg, D., and Roditty, L. (2009).
Fault-tolerant spanners for general graphs.
In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 435–444.

Fung, W. S., Hariharan, R., Harvey, N. J. A., and Panigrahi, D. (2011).
A general framework for graph sparsification.
In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 71–80.

Goel, A., Kapralov, M., and Khanna, S. (2012).
On the communication and streaming complexity of maximum bipartite matching.

In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 468–485. SIAM.

Karger, D. R. (1994).
Random sampling in cut, flow, and network design problems.
In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 648–657.

Lee, E. and Singla, S. (2017).
Maximum matching in the online batch-arrival model.
In *Integer Programming and Combinatorial Optimization - 19th International Conference, IPCO 2017, Waterloo, ON, Canada, June 26-28, 2017, Proceedings*, pages 355–367.

Peleg, D. (2009).
As good as it gets: Competitive fault tolerance in network structures.

In *Stabilization, Safety, and Security of Distributed Systems, 11th International Symposium, SSS 2009, Lyon, France, November 3-6, 2009. Proceedings*, pages 35–46.

Peleg, D. and Schäffer, A. A. (1989).
Graph spanners.
Journal of Graph Theory, 13(1):99–116.

Spielman, D. A. and Teng, S. (2011).
Spectral sparsification of graphs.
SIAM J. Comput., 40(4):981–1025.

Yamaguchi, Y. and Maehara, T. (2018).
Stochastic packing integer programs with few queries.
In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 293–310.