Coresets Meet EDCS: Algorithms for Matching and Vertex Cover on Massive Graphs

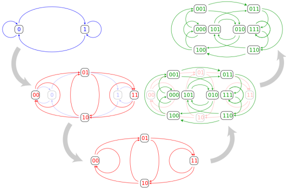
Sepehr Assadi

University of Pennsylvania

Joint work with MohammadHossein Bateni (Google), Aaron Bernstein (Rutgers), Vahab Mirrokni (Google), and Cliff Stein (Columbia)

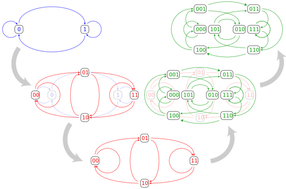
Massive Graphs

Massive graphs abound in variety of applications: web graph, social networks, biological networks, etc.



Massive Graphs

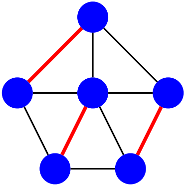
Massive graphs abound in variety of applications: web graph, social networks, biological networks, etc.



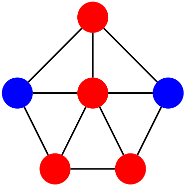
This talk: Matching and Vertex Cover problems on massive graphs.

Matchings and Vertex Covers

- Matching: A collection of vertex-disjoint edges.



- Vertex Cover: A collection of vertices containing at least one end point of every edge.



Matchings and Vertex Covers

Rich sources of inspiration for breakthrough ideas in computer science, algorithm design, and complexity theory.

Complexity class P

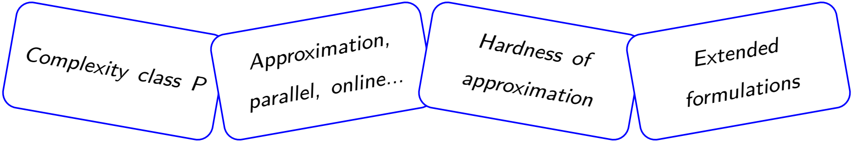
Approximation, parallel, online...

Hardness of approximation

Extended formulations

Matchings and Vertex Covers

Rich sources of inspiration for breakthrough ideas in computer science, algorithm design, and complexity theory.



This talk:

Randomized composable coresets for matching and vertex cover.

Their applications to different models including streaming, distributed, and massively parallel computation.

Randomized Composable Coresets

Definition ([A, Khanna'17])

Randomized Composable Coresets

Definition ([**A**, Khanna'17])

- Let $G^{(1)}, \ldots, G^{(k)}$ be a random partitioning of G: each edge $e \in G$ is sent to a subgraph $G^{(i)}$ uniformly at random.

Randomized Composable Coresets

Definition ([**A**, Khanna'17])

- Let $G^{(1)}, \ldots, G^{(k)}$ be a random partitioning of G: each edge $e \in G$ is sent to a subgraph $G^{(i)}$ uniformly at random.
- Consider an algorithm $\textsc{alg}$ that given $G^{(i)}$ outputs a subgraph $H^{(i)}$ of $G^{(i)}$ with s edges.

Randomized Composable Coresets

Definition ([**A**, Khanna'17])

- Let $G^{(1)}, \ldots, G^{(k)}$ be a random partitioning of G: each edge $e \in G$ is sent to a subgraph $G^{(i)}$ uniformly at random.
- Consider an algorithm $\textnormal{ALG}$ that given $G^{(i)}$ outputs a subgraph $H^{(i)}$ of $G^{(i)}$ with s edges.
- $\textnormal{ALG}$ outputs an α-approximation randomized composable coreset of size s for a problem P iff:

Randomized Composable Coresets

Definition ([**A**, Khanna'17])

- Let $G^{(1)}, \ldots, G^{(k)}$ be a random partitioning of G: each edge $e \in G$ is sent to a subgraph $G^{(i)}$ uniformly at random.
- Consider an algorithm $\textsc{alg}$ that given $G^{(i)}$ outputs a subgraph $H^{(i)}$ of $G^{(i)}$ with s edges.
- $\textsc{alg}$ outputs an α-approximation randomized composable coreset of size s for a problem P iff:

  $P(\textsc{alg}(G^{(1)}) \cup \ldots \cup \textsc{alg}(G^{(k)}))$ is an α-approximation to $P(G^{(1)} \cup \ldots \cup G^{(k)}) = P(G)$ with high probability.

Randomized Composable Coresets

Definition ([**A**, Khanna'17])

- Let $G^{(1)}, \ldots, G^{(k)}$ be a random partitioning of G: each edge $e \in G$ is sent to a subgraph $G^{(i)}$ uniformly at random.
- Consider an algorithm $\textnormal{ALG}$ that given $G^{(i)}$ outputs a subgraph $H^{(i)}$ of $G^{(i)}$ with s edges.
- $\textnormal{ALG}$ outputs an α-approximation randomized composable coreset of size s for a problem P iff:

  $P(\textnormal{ALG}(G^{(1)}) \cup \ldots \cup \textnormal{ALG}(G^{(k)}))$ is an α-approximation to $P(G^{(1)} \cup \ldots \cup G^{(k)}) = P(G)$ with high probability.

Algorithmic question. Design $\textnormal{ALG}$ with a good approximation ratio and a small size.

Randomized Composable Coresets

Definition ([**A**, Khanna'17])

- Let $G^{(1)}, \ldots, G^{(k)}$ be a random partitioning of G: each edge $e \in G$ is sent to a subgraph $G^{(i)}$ uniformly at random.
- Consider an algorithm $\textsc{alg}$ that given $G^{(i)}$ outputs a subgraph $H^{(i)}$ of $G^{(i)}$ with s edges.
- $\textsc{alg}$ outputs an α-approximation randomized composable coreset of size s for a problem P iff:

  $P(\textsc{alg}(G^{(1)}) \cup \ldots \cup \textsc{alg}(G^{(k)}))$ is an α-approximation to $P(G^{(1)} \cup \ldots \cup G^{(k)}) = P(G)$ with high probability.

Algorithmic question. Design $\textsc{alg}$ with a good approximation ratio and a small size.

Introduced first by [Mirrokni and Zadimoghaddam, 2015] for distributed submodular maximization.

Randomized Composable Coresets: Background

- Why this problem?

Randomized Composable Coresets: Background

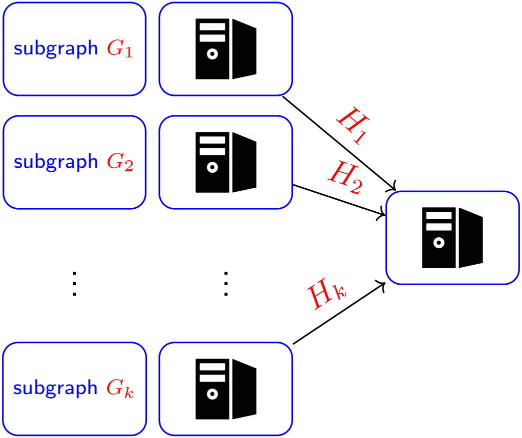
- Why this problem?
  - A natural problem that abstracts out one of the simplest approaches to large-scale optimization.

Randomized Composable Coresets: Background

- Why this problem?
  - A natural problem that abstracts out one of the simplest approaches to large-scale optimization.
  - Direct applications to distributed communication, massively parallel computation, and streaming.

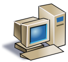
Randomized Composable Coresets: Applications

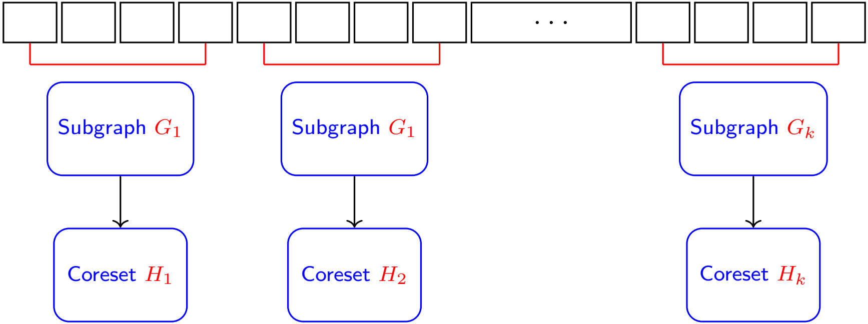
- An MPC algorithm with small memory per machine with one or two rounds of parallel computation.



Randomized Composable Coresets: Applications

- A streaming algorithm with small memory on random streams.





Randomized Composable Coresets: Background

- Why this problem?
  - Abstract out one of the simplest approach to large-scale optimization.
  - Applications to distributed, massively parallel computation, and streaming.
- Why random partitioning?

Randomized Composable Coresets: Background

- Why this problem?
  - Abstract out one of the simplest approach to large-scale optimization.
  - Applications to distributed, massively parallel computation, and streaming.
- Why random partitioning?
  - Adversarial partitions do not admit non-trivial solutions for matching and vertex cover [**A**, Khanna, Li, Yaroslavtsev'16].
    - ⋆ $n^{o(1)}$-approximation requires $n^{2-o(1)}$ space.

Randomized Composable Coresets: Background

- Why this problem?
  - Abstract out one of the simplest approach to large-scale optimization.
  - Applications to distributed, massively parallel computation, and streaming.
- Why random partitioning?
  - Adversarial partitions do not admit non-trivial solutions for matching and vertex cover [**A**, Khanna, Li, Yaroslavtsev'16].
    - $n^{o(1)}$-approximation requires $n^{2-o(1)}$ space.
  - Randomized composable coresets were suggested in [**A**, Khanna'17] to bypass these impossibility results.

State-of-the-Art

[**A**, Khanna'17]: There are $\widetilde{O}(n)$ size randomized composable coresets with:

- $O(1)$ approximation for matching, and
- $O(\log n)$ approximation for vertex cover.

State-of-the-Art

[**A**, Khanna'17]: There are $\widetilde{O}(n)$ size randomized composable coresets with:

- $O(1)$ approximation for matching, and
- $O(\log n)$ approximation for vertex cover.

[**A**, Khanna'17] used this to obtain improved distributed and MPC algorithms.

Motivating Question

The randomized composable coresets in [**A**, Khanna'17]:

- bypassed the impossibility results for previous techniques;
- gave a unified approach across multiple models.

Motivating Question

The randomized composable coresets in [**A**, Khanna'17]:

- bypassed the impossibility results for previous techniques;
- gave a unified approach across multiple models.

However, these randomized coresets

- had large approximation factors;
- could not compete with model-specific solutions in each model.

Motivating Question

The randomized composable coresets in [**A**, Khanna'17]:

- bypassed the impossibility results for previous techniques;
- gave a unified approach across multiple models.

However, these randomized coresets

- had large approximation factors;
- could not compete with model-specific solutions in each model.

Questions.

- Improved randomized composable coresets?
- Compete with model-specific solutions using this general technique?

Our Results

Our Results

We give significantly improved randomized composable coresets for matching and vertex cover.

Main Result. Randomized coresets of size $\widetilde{O}(n)$ with:

- $(1.5 + \varepsilon)$-approximation for matching, and
- $(2 + \varepsilon)$-approximation for vertex cover.

Our Results

We give significantly improved randomized composable coresets for matching and vertex cover.

Main Result. Randomized coresets of size $\widetilde{O}(n)$ with:

- $(1.5 + \varepsilon)$-approximation for matching, and
- $(2 + \varepsilon)$-approximation for vertex cover.

Size of these coresets are essentially optimal [**A**, Khanna'17].

Our Results

We give significantly improved randomized composable coresets for matching and vertex cover.

Main Result. Randomized coresets of size $\widetilde{O}(n)$ with:

- $(1.5 + \varepsilon)$-approximation for matching, and
- $(2 + \varepsilon)$-approximation for vertex cover.

Size of these coresets are essentially optimal [**A**, Khanna'17].

Improve upon state-of-the-art in streaming, distributed, and MPC model in one or all parameters involved.

Direct Applications of Our Main Result

Corollary (Streaming)

A single-pass streaming algorithm on random arrival streams for $(1.5 + \varepsilon)$-approximation of matching in $\widetilde{O}(n\sqrt{n})$ space.

Direct Applications of Our Main Result

Corollary (Streaming)

A single-pass streaming algorithm on random arrival streams for $(1.5+\varepsilon)$-approximation of matching in $\widetilde{O}(n\sqrt{n})$ space.

Previously,

- Getting better than 2-approximation with $o(n^2)$ space in adversarial streams is a big open question.
- Better than $\frac{e}{e-1} \approx 1.58$ approximation in adversarial streams requires $n^{1+\Omega(1/\log\log n)}$ space [Kapralov, 2013].

Direct Applications of Our Main Result

Corollary (Streaming)

A single-pass streaming algorithm on random arrival streams for $(1.5+\varepsilon)$-approximation of matching in $\widetilde{O}(n\sqrt{n})$ space.

Previously,

- Getting better than 2-approximation with $o(n^2)$ space in adversarial streams is a big open question.
- Better than $\frac{e}{e-1} \approx 1.58$ approximation in adversarial streams requires $n^{1+\Omega(1/\log\log n)}$ space [Kapralov, 2013].
- [Konrad et al., 2012]: a 1.98-approximation to matching in random arrival streams with $\widetilde{O}(n)$ space.
- [Konrad, 2018]: improved approximation to 1.85 (following [Esfandiari et al., 2016, Kale and Tirodkar, 2017]).

Our Randomized Composable Coresets for Matching and Vertex Cover

Our Main Result

Randomized composable coresets of size $\widetilde{O}(n)$ with:

- $(1.5 + \varepsilon)$-approximation for matching, and
- $(2 + \varepsilon)$-approximation for vertex cover.

Our Main Result

Randomized composable coresets of size $\widetilde{O}(n)$ with:

- $(1.5 + \varepsilon)$-approximation for matching, and
- $(2 + \varepsilon)$-approximation for vertex cover.

We mostly focus on maximum matching in this talk.

High Level Approach

The goal in randomized composable coresets:

High Level Approach

The goal in randomized composable coresets:

- Find a subgraph $H^{(i)}$ of each $G^{(i)}$ so that $H^{(1)} \cup \ldots \cup H^{(k)}$ contains a large matching of $G^{(1)} \cup \ldots \cup G^{(k)}$.

High Level Approach

The goal in randomized composable coresets:

- Find a subgraph $H^{(i)}$ of each $G^{(i)}$ so that $H^{(1)} \cup \ldots \cup H^{(k)}$ contains a large matching of $G^{(1)} \cup \ldots \cup G^{(k)}$.
- Each $H^{(i)}$ should be a "good" representative of "large" matchings in $G^{(i)}$.

High Level Approach

The goal in randomized composable coresets:

- Find a subgraph $H^{(i)}$ of each $G^{(i)}$ so that $H^{(1)} \cup \ldots \cup H^{(k)}$ contains a large matching of $G^{(1)} \cup \ldots \cup G^{(k)}$.
- Each $H^{(i)}$ should be a "good" representative of "large" matchings in $G^{(i)}$.

[**A**, Khanna'17] used maximum matching as coresets.

High Level Approach

The goal in randomized composable coresets:

- Find a subgraph $H^{(i)}$ of each $G^{(i)}$ so that $H^{(1)} \cup \ldots \cup H^{(k)}$ contains a large matching of $G^{(1)} \cup \ldots \cup G^{(k)}$.
- Each $H^{(i)}$ should be a "good" representative of "large" matchings in $G^{(i)}$.

[**A**, Khanna'17] used maximum matching as coresets.

Maximum matchings do not seem to be robust enough representation of all large matchings.

High Level Approach

The goal in randomized composable coresets:

- Find a subgraph $H^{(i)}$ of each $G^{(i)}$ so that $H^{(1)} \cup \ldots \cup H^{(k)}$ contains a large matching of $G^{(1)} \cup \ldots \cup G^{(k)}$.
- Each $H^{(i)}$ should be a "good" representative of "large" matchings in $G^{(i)}$.

[**A**, Khanna'17] used maximum matching as coresets.

Maximum matchings do not seem to be robust enough representation of all large matchings.

In particular, using maximum matchings as coresets cannot yield a better than 2 approximation.

High Level Approach

The goal in randomized composable coresets:

- Find a subgraph $H^{(i)}$ of each $G^{(i)}$ so that $H^{(1)} \cup \ldots \cup H^{(k)}$ contains a large matching of $G^{(1)} \cup \ldots \cup G^{(k)}$.
- Each $H^{(i)}$ should be a "good" representative of "large" matchings in $G^{(i)}$.

[**A**, Khanna'17] used maximum matching as coresets.

Maximum matchings do not seem to be robust enough representation of all large matchings.

In particular, using maximum matchings as coresets cannot yield a better than 2 approximation.

We instead use edge degree constrained subgraphs to represent large matchings.

Edge Degree Constrained Subgraphs

Definition ([Bernstein and Stein, 2015])

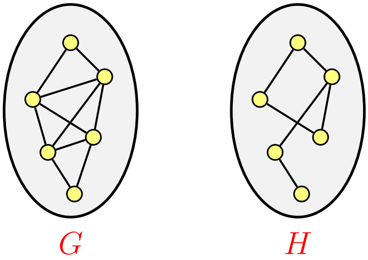
For any $\varepsilon \in (0,1)$ and $\beta \geq 1$,

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:



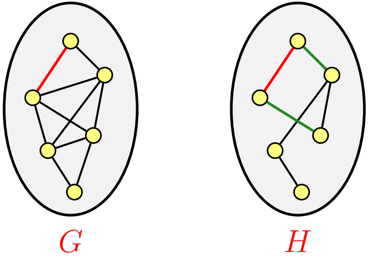
Edge Degree Constrained Subgraphs

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



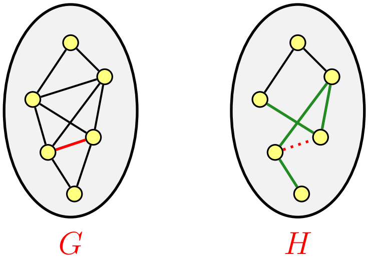
Edge Degree Constrained Subgraphs

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



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

Previously used in the context of dynamic graph algorithms in [Bernstein and Stein, 2015, Bernstein and Stein, 2016].

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

Previously used in the context of dynamic graph algorithms in [Bernstein and Stein, 2015, Bernstein and Stein, 2016].

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 $\beta > 1/\varepsilon$.

Edge Degree Constrained Subgraphs

What is special about an EDCS in general?

Edge Degree Constrained Subgraphs

What is special about an EDCS in general?

- [Bernstein and Stein, 2016]: A (β, ε)-EDCS always contains a $(1.5 + \varepsilon)$-approximate matching for $\beta > 1/\varepsilon^3$.
- [this work]: A (β, ε)-EDCS can always be used to recover a $(2 + \varepsilon)$-approximate vertex cover for $\beta > 1/\varepsilon$.

Edge Degree Constrained Subgraphs

What is special about an EDCS in general?

- [Bernstein and Stein, 2016]: A (β, ε)-EDCS always contains a $(1.5 + \varepsilon)$-approximate matching for $\beta > 1/\varepsilon^3$.
- [this work]: A (β, ε)-EDCS can always be used to recover a $(2 + \varepsilon)$-approximate vertex cover for $\beta > 1/\varepsilon$.

What is special about an EDCS for randomized composable coresets?

Edge Degree Constrained Subgraphs

What is special about an EDCS in general?

- [Bernstein and Stein, 2016]: A (β, ε)-EDCS always contains a $(1.5 + \varepsilon)$-approximate matching for $\beta > 1/\varepsilon^3$.
- [this work]: A (β, ε)-EDCS can always be used to recover a $(2 + \varepsilon)$-approximate vertex cover for $\beta > 1/\varepsilon$.

What is special about an EDCS for randomized composable coresets?

[this work]: W.h.p. on random partitions:

$$\text{EDCS}(G^{(1)}) \cup \ldots \cup \text{EDCS}(G^{(k)}) \approx \text{EDCS}(G^{(1)} \cup \ldots \cup G^{(k)}).$$

EDCS as a Randomized Coreset

Our main technical result:

Let $G^{(1)}, \ldots, G^{(k)}$ be a random partitioning of G.
Let $H^{(i)}$ be an arbitrary (β, ε)-EDCS of $G^{(i)}$.
Then $H^{(1)} \cup \ldots \cup H^{(k)}$ is a $\left(k\beta, \widetilde{\Theta}(\varepsilon)\right)$-EDCS of G w.h.p.

EDCS as a Randomized Coreset

Our main technical result:

Let $G^{(1)}, \ldots, G^{(k)}$ be a random partitioning of G.
Let $H^{(i)}$ be an arbitrary (β, ε)-EDCS of $G^{(i)}$.
Then $H^{(1)} \cup \ldots \cup H^{(k)}$ is a $\left(k\beta, \widetilde{\Theta}(\varepsilon)\right)$-EDCS of G w.h.p.

Randomized Composable Coreset:

Let the randomized coreset be an arbitrary $\left(\widetilde{\Theta}(1), \widetilde{\Theta}(\varepsilon)\right)$-EDCS.

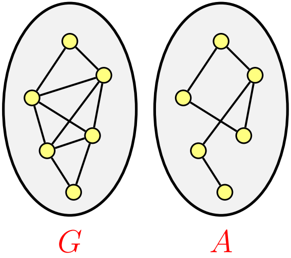
Size of each coreset is $\widetilde{O}(n)$.

Approximation follows from general properties of EDCS.

Proof Sketch of the Main Technical Result

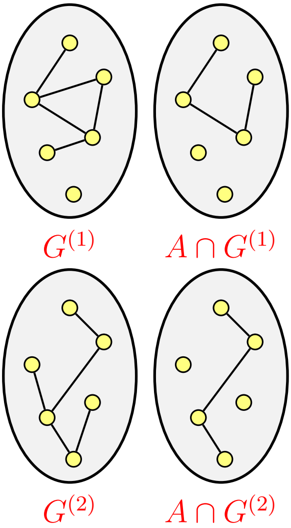
Proof Sketch of the Main Technical Result

- Fix a $\left(k\beta, \widetilde{\Theta}(\varepsilon)\right)$-EDCS A of the input graph G.



Proof Sketch of the Main Technical Result

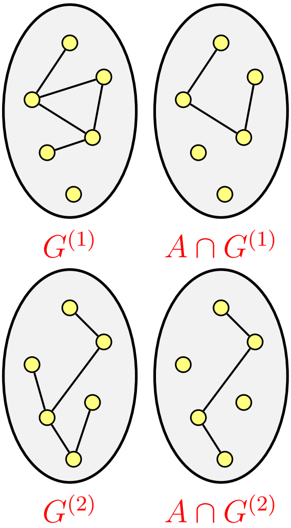
- Fix a $\left(k\beta, \widetilde{\Theta}(\varepsilon)\right)$-EDCS A of the input graph G.
- $A \cap G^{(i)}$ is w.h.p. a (β, ε)-EDCS of $G^{(i)}$.



Proof Sketch of the Main Technical Result

- Fix a $\left(k\beta, \widetilde{\Theta}(\varepsilon)\right)$-EDCS A of the input graph G.
- $A \cap G^{(i)}$ is w.h.p. a (β, ε)-EDCS of $G^{(i)}$.

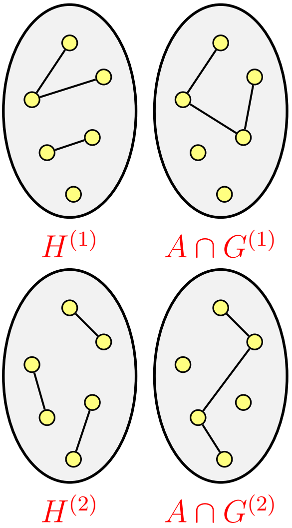
  (Proof: random partitioning preserves degrees after scaling by k)



Proof Sketch of the Main Technical Result

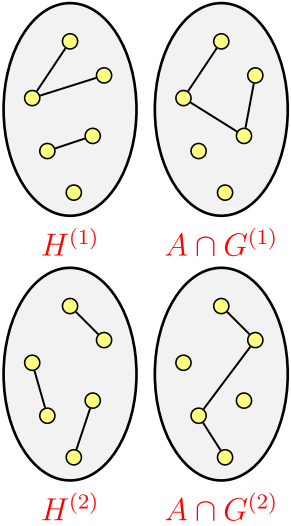
- Fix a $\left(k\beta, \widetilde{\Theta}(\varepsilon)\right)$-EDCS A of the input graph G.
- $A \cap G^{(i)}$ is w.h.p. a (β, ε)-EDCS of $G^{(i)}$.

  (Proof: random partitioning preserves degrees after scaling by k)
- Each $H^{(i)}$ is also another (β, ε)-EDCS of $G^{(i)}$ by construction.



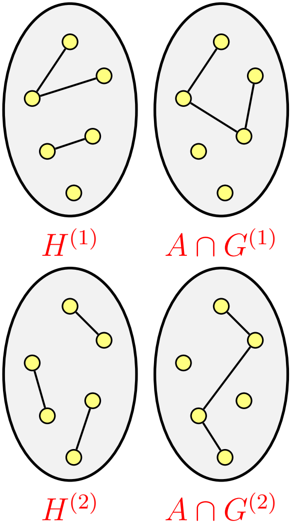
Proof Sketch of the Main Technical Result

- Ideal Scenario? $H^{(i)} = A \cap G^{(i)}$ for all $i \in [k]$.



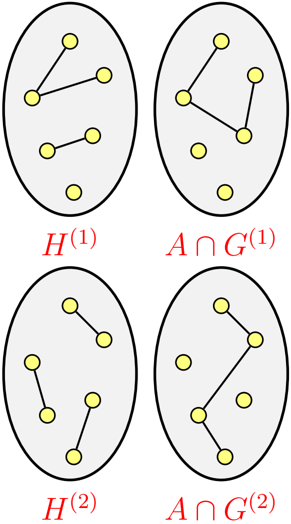
Proof Sketch of the Main Technical Result

- Ideal Scenario? $H^{(i)} = A \cap G^{(i)}$ for all $i \in [k]$.
  ($H^{(1)} \cup \ldots \cup H^{(k)}$ equals A, an $\left(k\beta, \widetilde{\Theta}(\varepsilon)\right)$-EDCS).



Proof Sketch of the Main Technical Result

- Ideal Scenario? $H^{(i)} = A \cap G^{(i)}$ for all $i \in [k]$.
  ($H^{(1)} \cup \ldots \cup H^{(k)}$ equals A, an $\left(k\beta, \widetilde{\Theta}(\varepsilon)\right)$-EDCS).
- This requires (β, ε)-EDCS to be unique.

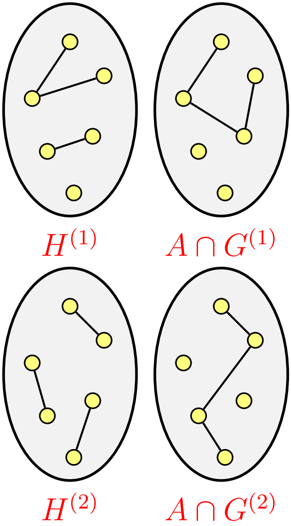


Proof Sketch of the Main Technical Result

- Ideal Scenario? $H^{(i)} = A \cap G^{(i)}$ for all $i \in [k]$.

  ($H^{(1)} \cup \ldots \cup H^{(k)}$ equals A, an $\left(k\beta, \widetilde{\Theta}(\varepsilon)\right)$-EDCS).
- This requires (β, ε)-EDCS to be unique.

  (this is not the case in general).

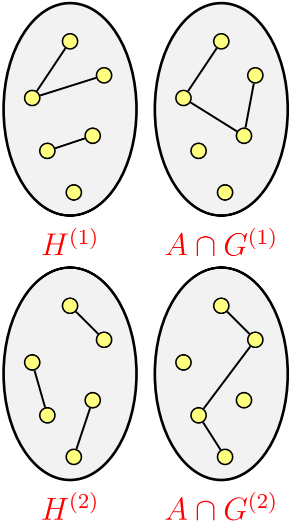


Proof Sketch of the Main Technical Result

- Ideal Scenario? $H^{(i)} = A \cap G^{(i)}$ for all $i \in [k]$.

  ($H^{(1)} \cup \ldots \cup H^{(k)}$ equals A, an $\left(k\beta, \widetilde{\Theta}(\varepsilon)\right)$-EDCS).
- This requires (β, ε)-EDCS to be unique.

  (this is not the case in general).
- Any fix?



Proof Sketch of the Main Technical Result

We prove that degree-distribution of a (β, ε)-EDCS is almost unique.

Proof Sketch of the Main Technical Result

We prove that degree-distribution of a (β, ε)-EDCS is almost unique.

Let A and B be two (β, ε)-EDCS of a graph G. For all $v \in V(G)$:

$$d_A(v) = d_B(v) \pm \widetilde{\Theta}(\varepsilon\beta).$$

Proof Sketch of the Main Technical Result

We prove that degree-distribution of a (β, ε)-EDCS is almost unique.

Let A and B be two (β, ε)-EDCS of a graph G. For all $v \in V(G)$:

$$d_A(v) = d_B(v) \pm \widetilde{\Theta}(\varepsilon\beta).$$

Enough to conclude that $H^{(1)} \cup \ldots \cup H^{(k)}$ is a $\left(k\beta, \widetilde{\Theta}(\varepsilon)\right)$-EDCS of G by the previous argument.

Wrap-Up

We proved,

Let $G^{(1)}, \ldots, G^{(k)}$ be a random partitioning of G.
Let $H^{(i)}$ be an arbitrary (β, ε)-EDCS of $G^{(i)}$.
Then $H^{(1)} \cup \ldots \cup H^{(k)}$ is a $\left(k\beta, \widetilde{\Theta}(\varepsilon)\right)$-EDCS of G w.h.p.

Wrap-Up

We proved,

Let $G^{(1)}, \ldots, G^{(k)}$ be a random partitioning of G.
Let $H^{(i)}$ be an arbitrary (β, ε)-EDCS of $G^{(i)}$.
Then $H^{(1)} \cup \ldots \cup H^{(k)}$ is a $\left(k\beta, \widetilde{\Theta}(\varepsilon)\right)$-EDCS of G w.h.p.

Combined with general properties of EDCS, this implies:

Randomized composable coresets of size $\widetilde{O}(n)$ with:

- $(1.5 + \varepsilon)$-approximation for matching, and
- $(2 + \varepsilon)$-approximation for vertex cover.

Concluding Remarks

Distributed Sparsification

Randomized composable coresets can be viewed as a distributed sparsification method:

Distributed Sparsification

Randomized composable coresets can be viewed as a distributed sparsification method:

1. Distribute the graph randomly across multiple machines.
2. Compute the coreset on each machine separately.
3. The union of the coreset is a sparser graph.
4. Solve the problem locally on this sparser graph.

Distributed Sparsification

Randomized composable coresets can be viewed as a distributed sparsification method:

1. Distribute the graph randomly across multiple machines.
2. Compute the coreset on each machine separately.
3. The union of the coreset is a sparser graph.
4. Solve the problem locally on this sparser graph.

We take this view to the next step for MPC algorithms.

Further Application to MPC

1. Distribute the graph randomly across multiple machines.
2. Compute the coreset on each machine separately.
3. The union of the coreset is a sparser graph.
4. Solve the problem locally on this sparser graph.

Further Application to MPC

1. Distribute the graph randomly across multiple machines.
2. Compute the coreset on each machine separately.
3. The union of the coreset is a sparser graph.
4. ~~Solve the problem locally on this sparser graph.~~
   Recurse on this sparser graph.

Further Application to MPC

1. Distribute the graph randomly across multiple machines.
2. Compute the coreset on each machine separately.
3. The union of the coreset is a sparser graph.
4. ~~Solve the problem locally on this sparser graph.~~
   Recurse on this sparser graph.

To make this work:

- Vertex-based partitioning approach of [Czumaj et al., 2018].
- Additional care to not blow up approximation due to recursion.

Further Application to MPC

Corollary (MPC with low-memory per-machine)

An $O(\log\log n)$*-round MPC algorithm with* $O(1)$*-approximation to both matching and vertex cover and only* $O(n)$ *memory per-machine.*

Further Application to MPC

Corollary (MPC with low-memory per-machine)

An $O(\log\log n)$-round MPC algorithm with $O(1)$-approximation to both matching and vertex cover and only $O(n)$ memory per-machine.

- Can also give $(1+\varepsilon)$-approximation to maximum matching.
- Memory can be reduced to $O(n/\mathrm{polylog}(n))$.

Further Applications

Corollary (MPC with low-memory per-machine)

An $O(\log\log n)$*-round MPC algorithm with* $O(1)$*-approximation to both matching and vertex cover and only* $O(n)$ *memory per-machine.*

Previously,

Further Applications

Corollary (MPC with low-memory per-machine)

An $O(\log\log n)$-round MPC algorithm with $O(1)$-approximation to both matching and vertex cover and only $O(n)$ memory per-machine.

Previously,

- [Lattanzi et al., 2011]: $O(\log n)$ rounds; 2-approximation to both problems; $O(n)$ memory.

Further Applications

Corollary (MPC with low-memory per-machine)

An $O(\log\log n)$*-round MPC algorithm with* $O(1)$*-approximation to both matching and vertex cover and only* $O(n)$ *memory per-machine.*

Previously,

- [Lattanzi et al., 2011]: $O(\log n)$ rounds; 2-approximation to both problems; $O(n)$ memory.
- [Czumaj et al., 2018]: $O((\log\log n)^2)$ rounds; $O(1)$-approximation only to matching; $O(n)$ memory.

Further Applications

Corollary (MPC with low-memory per-machine)

An $O(\log\log n)$-round MPC algorithm with $O(1)$-approximation to both matching and vertex cover and only $O(n)$ memory per-machine.

Previously,

- [Lattanzi et al., 2011]: $O(\log n)$ rounds; 2-approximation to both problems; $O(n)$ memory.
- [Czumaj et al., 2018]: $O((\log\log n)^2)$ rounds; $O(1)$-approximation only to matching; $O(n)$ memory.

Subsequently,

Further Applications

Corollary (MPC with low-memory per-machine)

An $O(\log\log n)$-round MPC algorithm with $O(1)$-approximation to both matching and vertex cover and only $O(n)$ memory per-machine.

Previously,

- [Lattanzi et al., 2011]: $O(\log n)$ rounds; 2-approximation to both problems; $O(n)$ memory.
- [Czumaj et al., 2018]: $O((\log\log n)^2)$ rounds; $O(1)$-approximation only to matching; $O(n)$ memory.

Subsequently,

- [Ghaffari et al., 2018]: $O(\log\log n)$ rounds; $(2+\varepsilon)$-approximation to both problems; $O(n)$ memory.

Concluding Remarks

Randomized composable coresets:

- A unified approach for algorithm design in different models.
- A distributed sparsification method particularly useful for MPC.

Concluding Remarks

Randomized composable coresets:

- A unified approach for algorithm design in different models.
- A distributed sparsification method particularly useful for MPC.

Randomized composable coresets of size $\widetilde{O}(n)$ with $(1.5+\varepsilon)$- and $(2+\varepsilon)$-approximation to matching and vertex cover.

Concluding Remarks

Randomized composable coresets:

- A unified approach for algorithm design in different models.
- A distributed sparsification method particularly useful for MPC.

Randomized composable coresets of size $\widetilde{O}(n)$ with $(1.5+\varepsilon)$- and $(2+\varepsilon)$-approximation to matching and vertex cover.

Some key applications:

- A random arrival streaming $(1.5+\varepsilon)$-approximation to matching.
- An $O(\log\log n)$-round MPC $(1+\varepsilon)$-approximation and $O(1)$-approximation to matching and vertex cover with $O(n/\text{poly}\log(n))$ memory.

Concluding Remarks

Randomized composable coresets:

- A unified approach for algorithm design in different models.
- A distributed sparsification method particularly useful for MPC.

Randomized composable coresets of size $\widetilde{O}(n)$ with $(1.5+\varepsilon)$- and $(2+\varepsilon)$-approximation to matching and vertex cover.

Some key applications:

- A random arrival streaming $(1.5+\varepsilon)$-approximation to matching.
- An $O(\log\log n)$-round MPC $(1+\varepsilon)$-approximation and $O(1)$-approximation to matching and vertex cover with $O(n/\mathrm{poly}\log(n))$ memory.

Thank you!

Ahn, K. J. and Guha, S. (2015).
Access to data and number of iterations: Dual primal algorithms for maximum matching under resource constraints.
In *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*, pages 202–211.

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.

Czumaj, A., Lacki, J., Madry, A., Mitrovic, S., Onak, K., and Sankowski, P. (2018).
Round compression for parallel matching algorithms.
In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 471–484.

Esfandiari, H., Hajiaghayi, M., and Monemizadeh, M. (2016).
Finding large matchings in semi-streaming.
In *IEEE International Conference on Data Mining Workshops, ICDM Workshops 2016, December 12-15, 2016, Barcelona, Spain.*, pages 608–614.

Ghaffari, M., Gouleakis, T., Konrad, C., Mitrovic, S., and Rubinfeld, R. (2018).
Improved massively parallel computation algorithms for mis, matching, and vertex cover.

In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018*, pages 129–138.

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.

Kale, S. and Tirodkar, S. (2017).
Maximum matching in two, three, and a few more passes over graph streams.
In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, pages 15:1–15:21.

Kapralov, M. (2013).

Better bounds for matchings in the streaming model.
In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1679–1697.

Konrad, C. (2018).
A simple augmentation method for matchings with applications to streaming algorithms.
In *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, pages 74:1–74:16.

Konrad, C., Magniez, F., and Mathieu, C. (2012).
Maximum matching in semi-streaming with few passes.
In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, pages 231–242.

Lattanzi, S., Moseley, B., Suri, S., and Vassilvitskii, S. (2011).
Filtering: a method for solving graph problems in mapreduce.
In *SPAA 2011: Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Jose, CA, USA, June 4-6, 2011 (Co-located with FCRC 2011)*, pages 85–94.

Mirrokni, V. S. and Zadimoghaddam, M. (2015).
Randomized composable core-sets for distributed submodular maximization.
In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 153–162.