Δ-coloring in the graph streaming model

Pankaj Kumar   Parth Mittal
Mentor: Sepehr Assadi

REU 2020, Rutgers University

Introduction

- Setting: Graph edges stream by one at a time in adversarial order.

Introduction

- Setting: Graph edges stream by one at a time in adversarial order.
- Graph coloring: a function $\mathcal{C} : V \to [k]$ which maps adjacent vertices to different values is a k-coloring.

Introduction

- Setting: Graph edges stream by one at a time in adversarial order.
- Graph coloring: a function $\mathcal{C} : V \to [k]$ which maps adjacent vertices to different values is a k-coloring.
- Want to color the graph with few colors, while using small space.

Introduction

- Setting: Graph edges stream by one at a time in adversarial order.
- Graph coloring: a function $\mathcal{C} : V \to [k]$ which maps adjacent vertices to different values is a k-coloring.
- Want to color the graph with few colors, while using small space.
- With n the number of vertices, can store entire graph in $O(n^2)$ space – would like to use $o(n^2)$ space (and ideally $\widetilde{O}(n)$).

Motivation

- All graphs admit a $(\Delta + 1)$-coloring, where Δ is the maximum degree of the graph.
  - Can be obtained greedily.

Motivation

- All graphs admit a $(\Delta + 1)$-coloring, where Δ is the maximum degree of the graph.
  - Can be obtained greedily.
- By a result of Assadi, Chen and Khanna (SODA'19) [1], can $(\Delta + 1)$-color in $\widetilde{O}(n)$ space with a randomized algorithm.

Motivation

- All graphs admit a $(\Delta+1)$-coloring, where Δ is the maximum degree of the graph.
  - Can be obtained greedily.
- By a result of Assadi, Chen and Khanna (SODA'19) [1], can $(\Delta+1)$-color in $\widetilde{O}(n)$ space with a randomized algorithm.
- Almost all graphs admit a Δ-coloring: Every connected graph which is not a clique or an odd-cycle can be Δ-colored (Brook's Theorem).

Motivation

- All graphs admit a $(\Delta + 1)$-coloring, where Δ is the maximum degree of the graph.
  - Can be obtained greedily.
- By a result of Assadi, Chen and Khanna (SODA'19) [1], can $(\Delta + 1)$-color in $\widetilde{O}(n)$ space with a randomized algorithm.
- Almost all graphs admit a Δ-coloring: Every connected graph which is not a clique or an odd-cycle can be Δ-colored (Brook's Theorem).
- Can we find a Δ-coloring in $o(n^2)$ space?

Our result

Theorem

There is an $\widetilde{O}(n^{7/4})$ space that given one pass over edges of any graph $G = (V, E)$ with maximum degree Δ, with high probability, finds a Δ-coloring of G or outputs that G does not admit a Δ-coloring.

Preliminaries

We use the Extended HSS Decomposition from [1], which for any $\varepsilon \in [0, 1)$ decomposes a graph $G(V, E)$ into:

- Sparse vertices: Neighbourhood of each sparse vertex is missing at least $\varepsilon\binom{\Delta}{2}$ edges.
- A collection of almost-cliques; each almost-clique C:
  - contains $(1 \pm \varepsilon)\Delta$ vertices.
  - every vertex in C has $\leq \varepsilon\Delta$ neighbours outside C.
  - every vertex in C has $\leq \varepsilon\Delta$ non-neighbours inside C.

Preliminaries

We use the Extended HSS Decomposition from [1], which for any $\varepsilon \in [0, 1)$ decomposes a graph $G(V, E)$ into:

- Sparse vertices: Neighbourhood of each sparse vertex is missing at least $\varepsilon\binom{\Delta}{2}$ edges.
- A collection of almost-cliques; each almost-clique C:
  - contains $(1 \pm \varepsilon)\Delta$ vertices.
  - every vertex in C has $\leq \varepsilon\Delta$ neighbours outside C.
  - every vertex in C has $\leq \varepsilon\Delta$ non-neighbours inside C.

Can be found in a single pass using $O(n/\varepsilon^2)$ space ([1]).

The big ideas

- Recover all edges incident on C_i (in addition to the decomposition).

The big ideas

- Recover all edges incident on C_i (in addition to the decomposition).
- List color (with lists of size $\frac{\log n}{\varepsilon^2}$) the vertices of $V_\star^{\text{sparse}}$

The big ideas

- Recover all edges incident on C_i (in addition to the decomposition).
- List color (with lists of size $\frac{\log n}{\varepsilon^2}$) the vertices of $V_\star^{\text{sparse}}$
- Extend the partial coloring to almost cliques.

Sparse recovery

Lemma

For any $n > 0$ *and* $k \leq n$*, there exists a set of* $m = O(k \log \frac{n}{k})$ *measurements* $A \in \mathbb{F}_2^{m \times n}$ *for recovering any* k*-sparse vector* $x \in \mathbb{F}_2^n$*. Moreover* A *chosen randomly has this property with high-probability.*

Sparse recovery

Lemma

For any $n > 0$ and $k \leq n$, there exists a set of $m = O(k \log \frac{n}{k})$ measurements $A \in \mathbb{F}_2^{m \times n}$ for recovering any k-sparse vector $x \in \mathbb{F}_2^n$. Moreover A chosen randomly has this property with high-probability.

- Want to find y^v (adjacency vector) for each $v \in C_i$ for some i.

Sparse recovery

Lemma

For any $n > 0$ and $k \leq n$, there exists a set of $m = O(k \log \frac{n}{k})$ measurements $A \in \mathbb{F}_2^{m \times n}$ for recovering any k-sparse vector $x \in \mathbb{F}_2^n$. Moreover A chosen randomly has this property with high-probability.

- Want to find y^v (adjacency vector) for each $v \in C_i$ for some i.
- y^v only $\underline{\Delta}$-sparse, instead find $x^v = y^v - z^C$, where z^C is characteristic vector of almost-clique C containing v ($O(\varepsilon\Delta)$-sparse!!)

Sparse recovery

Lemma

For any $n > 0$ *and* $k \leq n$*, there exists a set of* $m = O(k \log \frac{n}{k})$ *measurements* $A \in \mathbb{F}_2^{m \times n}$ *for recovering any* k*-sparse vector* $x \in \mathbb{F}_2^n$*. Moreover* A *chosen randomly has this property with high-probability.*

- Want to find y^v (adjacency vector) for each $v \in C_i$ for some i.
- y^v only $\underline{\Delta}$-sparse, instead find $x^v = y^v - z^C$, where z^C is characteristic vector of almost-clique C containing v ($O(\varepsilon\Delta)$-sparse!!)
- $A \cdot y^v$ easy to update while streaming, can compute $A \cdot z^C$ from decomposition at end of stream.

Sparse recovery

Lemma

For any $n > 0$ and $k \leq n$, there exists a set of $m = O(k \log \frac{n}{k})$ measurements $A \in \mathbb{F}_2^{m \times n}$ for recovering any k-sparse vector $x \in \mathbb{F}_2^n$. Moreover A chosen randomly has this property with high-probability.

- Want to find y^v (adjacency vector) for each $v \in C_i$ for some i.
- y^v only $\underline{\Delta}$-sparse, instead find $x^v = y^v - z^C$, where z^C is characteristic vector of almost-clique C containing v ($O(\varepsilon\Delta)$-sparse!!)
- $A \cdot y^v$ easy to update while streaming, can compute $A \cdot z^C$ from decomposition at end of stream.
- Can find all edges incident on almost cliques in $\widetilde{O}(n\varepsilon\Delta)$ space (as opposed to $O(n\Delta)$ which is trivial).

Saving Colors on $V_\star^{\mathsf{sparse}}$

- Cannot extend arbitrary Δ-coloring outside almost-cliques to them.

Saving Colors on $V_\star^{\mathsf{sparse}}$

- Cannot extend arbitrary Δ-coloring outside almost-cliques to them.
- If we "save" some colors on $V_\star^{\mathsf{sparse}}$, can extend (sketch later).

Saving Colors on $V_\star^{\mathsf{sparse}}$

- Cannot extend arbitrary Δ-coloring outside almost-cliques to them.
- If we "save" some colors on $V_\star^{\mathsf{sparse}}$, can extend (sketch later).
- We will list-color $V_\star^{\mathsf{sparse}}$ using $(1-\delta)\Delta$ colors, where $\delta = \frac{\varepsilon^2}{100}$.

Saving Colors on $V_\star^{\text{sparse}}$

- Cannot extend arbitrary Δ-coloring outside almost-cliques to them.
- If we "save" some colors on $V_\star^{\text{sparse}}$, can extend (sketch later).
- We will list-color $V_\star^{\text{sparse}}$ using $(1-\delta)\Delta$ colors, where $\delta = \frac{\varepsilon^2}{100}$.

Before streaming edges: for each vertex, pick list of colors $L(v) \subset [(1-\delta)\Delta]$, with each color independently in $L(v)$ with probability $p := \frac{\alpha \log n}{3\varepsilon^2(1-\delta)\Delta}$.

Saving Colors on $V_\star^{\text{sparse}}$

- Cannot extend arbitrary Δ-coloring outside almost-cliques to them.
- If we "save" some colors on $V_\star^{\text{sparse}}$, can extend (sketch later).
- We will list-color $V_\star^{\text{sparse}}$ using $(1-\delta)\Delta$ colors, where $\delta = \frac{\varepsilon^2}{100}$.

Before streaming edges: for each vertex, pick list of colors $L(v) \subset [(1-\delta)\Delta]$, with each color independently in $L(v)$ with probability $p := \frac{\alpha \log n}{3\varepsilon^2(1-\delta)\Delta}$.

Lemma

With high probability, there exists a partial coloring function $\mathcal{C} : V \to [(1-\delta)\Delta] \cup \{\perp\}$ such that for all vertices $v \in V_\star^{\text{sparse}}, \mathcal{C}(v) \in L(v)$.

Extending the coloring to almost-cliques

This lemma is our main contribution:

Lemma

Given a partial coloring Φ *of* G *which colors only* $V_{\star}^{\mathsf{sparse}}$ *using* $(1-\varepsilon^2/100)\Delta$ *colors, where* $(\varepsilon^2/100)\Delta > 8n/\Delta$, *and all the edges incident on almost-cliques of* G, *we can find a proper* $\Delta-$*coloring of* G.

Proof Sketch

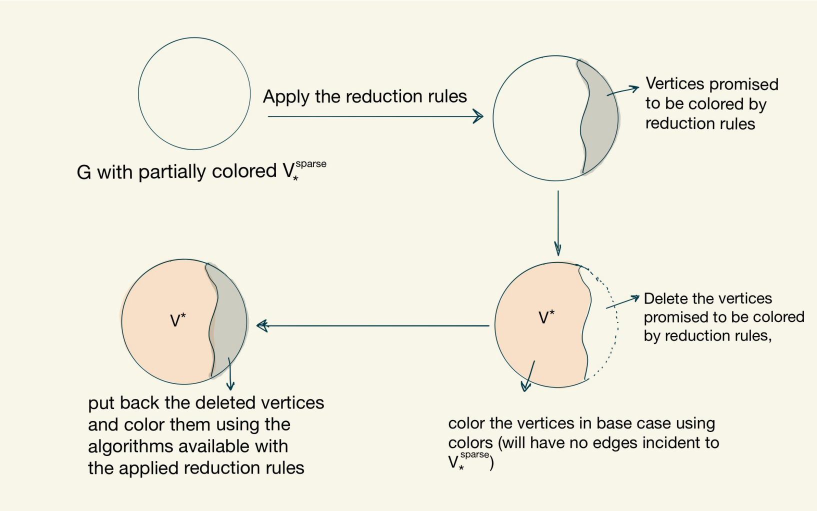

Figure: Overall idea

Proof Sketch (continued)

Common Theme: in all the cases we color a rooted spanning tree where the root has $< \Delta$ colors in its (fully colored) neighbourhood by either the structure of the graph, or by recoloring a vertex with a previously saved color (inspired by Lovász's proof of Brook's Theorem [2]).

Proof Sketch (continued)

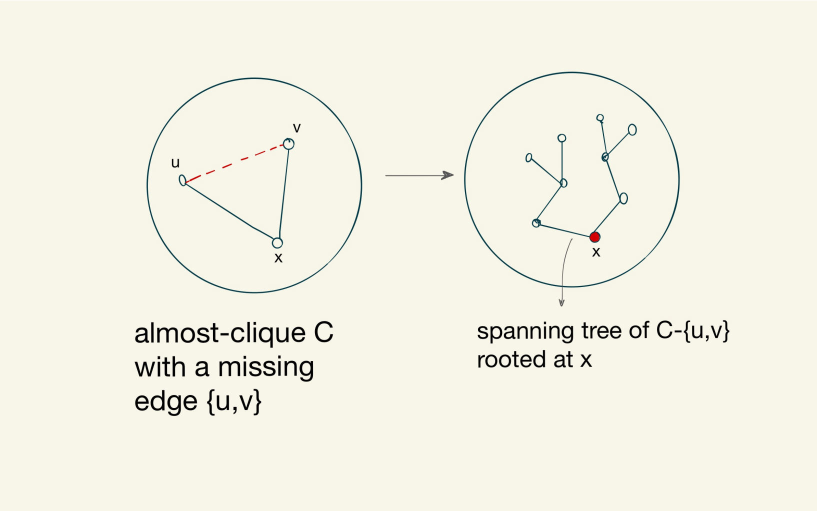

Figure: Rule I. Can color almost-cliques with an edge missing inside.

Proof Sketch (continued)

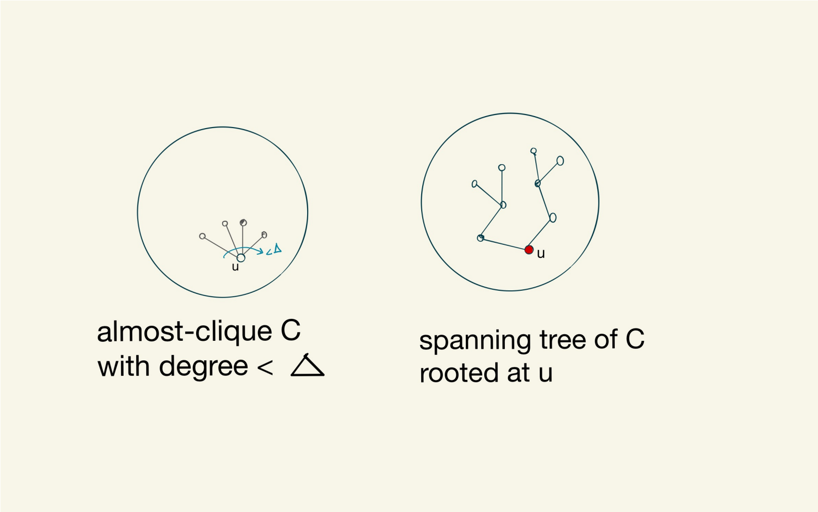

Figure: Rule II. Can color almost-cliques with a vertex of degree $< \Delta$

Proof Sketch (continued)

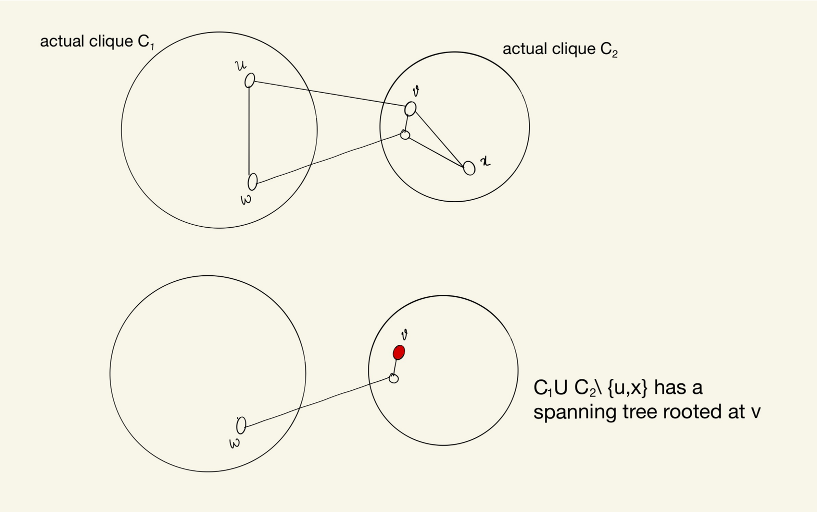

Figure: Rule III. Can color two cliques with two edges between them

Proof Sketch (continued)

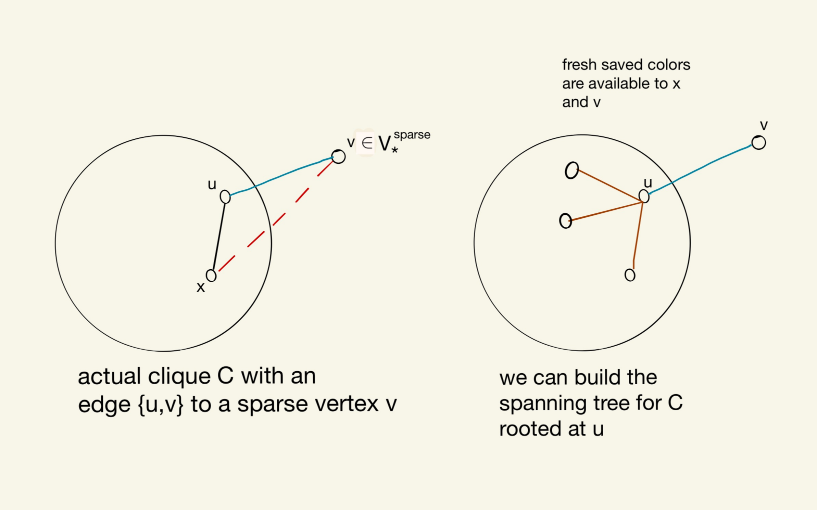

Figure: Rule IV. 1 saved color per remaining clique with an edge to a sparse vertex

Tying things up

- Storing edges incident on almost cliques takes $\widetilde{O}(n\varepsilon\Delta)$ space.
- Extended HSS decomposition takes $\widetilde{O}(n/\varepsilon^2)$ space.
- $(1-\delta)\Delta$-coloring sparse vertices only works if $\varepsilon \geq \frac{\log \Delta}{\Delta^{1/4}}$.
- Extending coloring to almost-cliques only works if $\varepsilon > \frac{\sqrt{n}}{\Delta}$.

Tying things up

- Storing edges incident on almost cliques takes $\widetilde{O}(n\varepsilon\Delta)$ space.
- Extended HSS decomposition takes $\widetilde{O}(n/\varepsilon^2)$ space.
- $(1-\delta)\Delta$-coloring sparse vertices only works if $\varepsilon \geq \frac{\log \Delta}{\Delta^{1/4}}$.
- Extending coloring to almost-cliques only works if $\varepsilon > \frac{\sqrt{n}}{\Delta}$.

With $\varepsilon = \frac{\log \Delta}{\Delta^{1/4}}$, we get $\widetilde{O}(n^{7/4})$ algorithm.

References

S. Assadi, Y. Chen, and S. Khanna.
Sublinear algorithms for $(\Delta + 1)$ vertex coloring.
In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 767–786. SIAM, 2019.

L. Lovász.
Three short proofs in graph theory.
J. Combinatorial Theory Ser. B, 19(3):269–271, 1975.

Acknowledgements

We would like to thank:

- Rutgers University, for organising this program.
- CoSP for supporting our participation (under European Union's Horizon 2020 research and innovation programme, grant agreement No. 823748)