Randomized Greedy Online Edge Coloring Succeeds for Dense and Randomly-Ordered Graphs

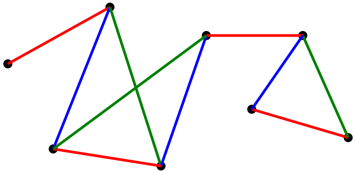
Aditi Dudeja[1] Rashmika Goswami [2] Michael Saks [2]

[1]University of Salzburg

[2]Rutgers University

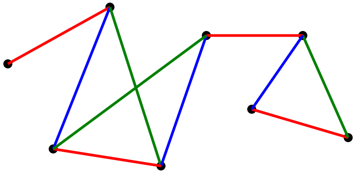
Edge Coloring

- Edge coloring: assignment of colors to edges of G so that no two edges adjacent to the same vertex have the same color.



Edge Coloring

- Edge coloring: assignment of colors to edges of G so that no two edges adjacent to the same vertex have the same color.



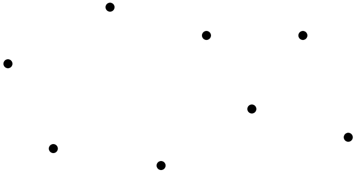
- Vizing: Graph with maximum degree Δ can be properly edge colored with $\Delta + 1$ colors (and requires at least Δ colors).

Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.

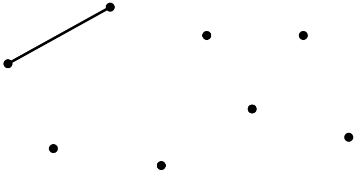
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



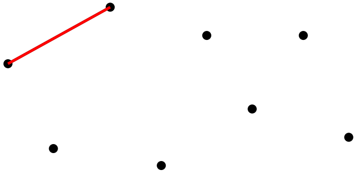
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



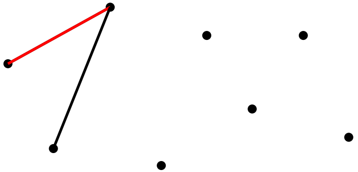
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.

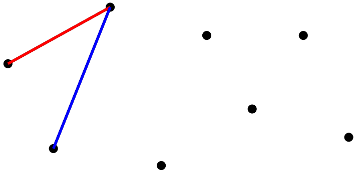


Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.

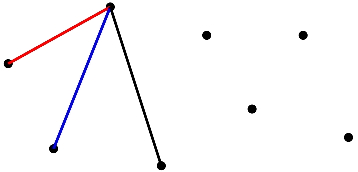


- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



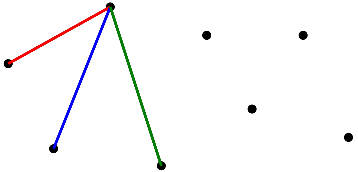
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



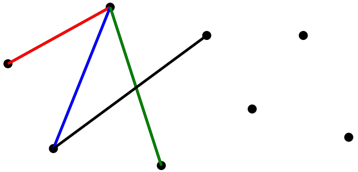
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



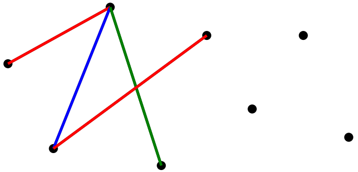
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



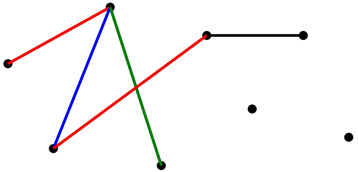
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



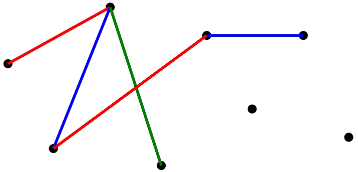
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



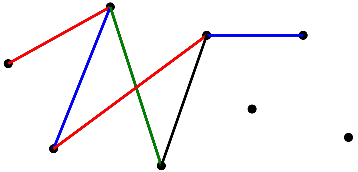
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



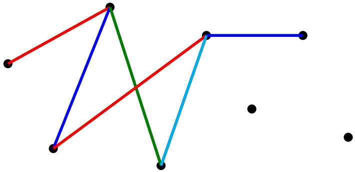
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



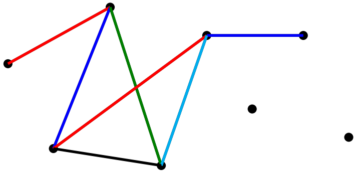
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



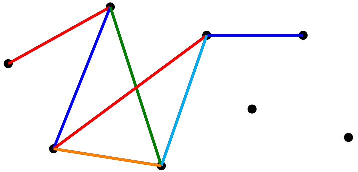
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



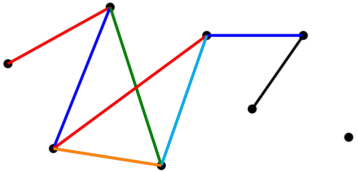
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



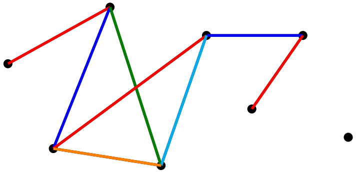
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



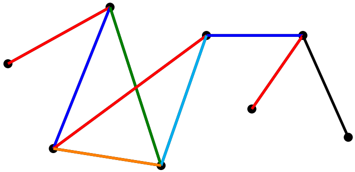
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



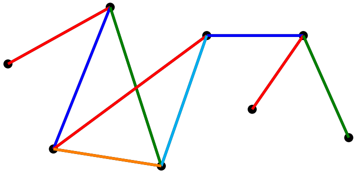
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



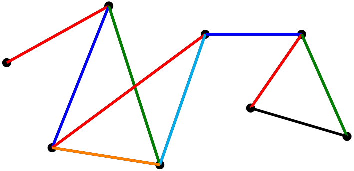
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



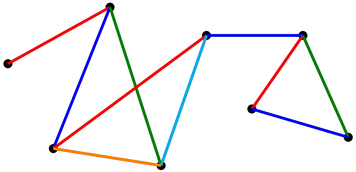
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



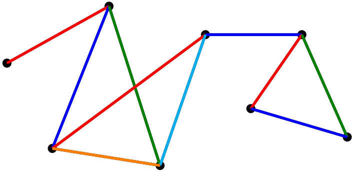
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



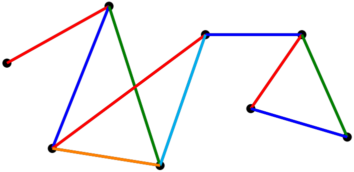
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



Online Algorithms

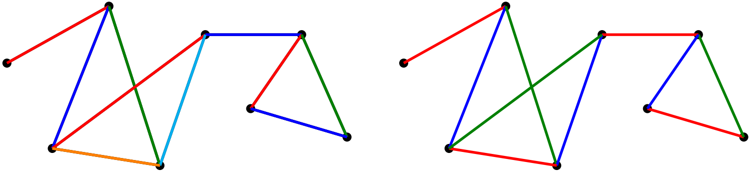
- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



- The greedy algorithm is an online algorithm that can $2\Delta - 1$ color a graph with maximum degree Δ.

Online Algorithms

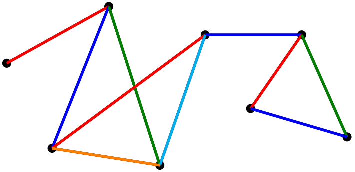
- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



- The greedy algorithm is an online algorithm that can $2\Delta - 1$ color a graph with maximum degree Δ.
- Bar-Noy, Motwani, and Naor 1992: This is optimal.

Online Algorithms

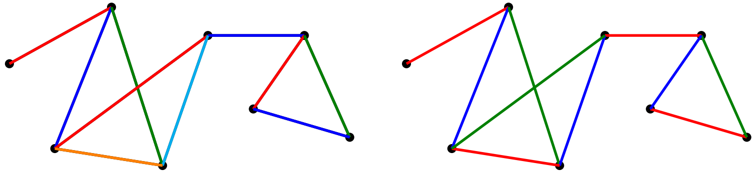
- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.



- The greedy algorithm is an online algorithm that can $2\Delta - 1$ color a graph with maximum degree Δ.
- Bar-Noy, Motwani, and Naor 1992: This is optimal... for $\Delta = O(\log n)$. Can we do better if $\Delta = \omega(\log n)$?

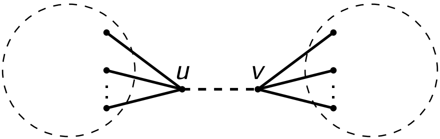
Online Algorithms

- Online setting: edges of the graph arrive one by one and are assigned colors upon arrival.

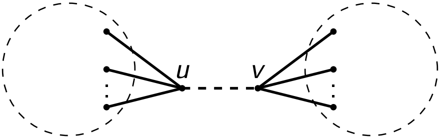


- The greedy algorithm is an online algorithm that can $2\Delta - 1$ color a graph with maximum degree Δ.
- Bar-Noy, Motwani, and Naor 1992: This is optimal... for $\Delta = O(\log n)$. Can we do better if $\Delta = \omega(\log n)$?
- Random greedy algorithm: start with a color set of size $(1 + \epsilon)\Delta$ and choose the color for each edge uniformly at random from the valid remaining colors upon arrival.

Intuition: Trees [Feder, Motwani, and Panigrahy n.d.]

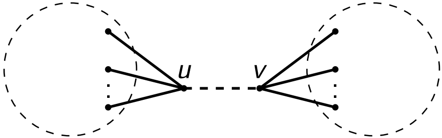


Intuition: Trees [Feder, Motwani, and Panigrahy n.d.]

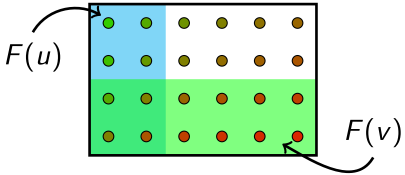


- Let $F(u)$ and $F(v)$ be the set of free (unused) colors from the palette at u and v respectively when edge (u, v) arrives.

Intuition: Trees [Feder, Motwani, and Panigrahy n.d.]



- Let $F(u)$ and $F(v)$ be the set of free (unused) colors from the palette at u and v respectively when edge (u, v) arrives.
- We expect



$$|F(u) \cap F(v)| \approx \frac{|F(u)||F(v)|}{(1+\epsilon)\Delta} \geq \frac{\epsilon^2 \Delta}{1+\epsilon}.$$

Results

Adversarial Settings:

- Random Order Arrival
- Oblivious Adversary
- Adaptive Adversary

Results

Adversarial Settings:

- Random Order Arrival
- Oblivious Adversary
- Adaptive Adversary

Theorem (Random Order Case)

For all ϵ, there exists a constant N s.t. if the edges of a graph G with maximum degree Δ on $n \leq 2^{\frac{\Delta}{N}}$ vertices arrive in a random order, then with high probability, the random greedy algorithm using $(1+\epsilon)\Delta$ colors succeeds.

Results

Adversarial Settings:

- Random Order Arrival
- Oblivious Adversary
- Adaptive Adversary

Theorem (Dense Case)

For all ϵ, M and $n \leq M\Delta$, the random greedy algorithm using $(1+\epsilon)\Delta$ colors succeeds with high probability on graphs with maximum degree Δ on n vertices, even if the edges of G are chosen by an adaptive adversary.

Results

Adversarial Settings:

- Random Order Arrival
- Oblivious Adversary
- Adaptive Adversary

Corollary (Deterministic Case)

For all ϵ, M there is a deterministic algorithm that, for Δ sufficiently large, will $(1+\epsilon)\Delta$ color graphs with maximum degree Δ on $n \leq M\Delta$ vertices.

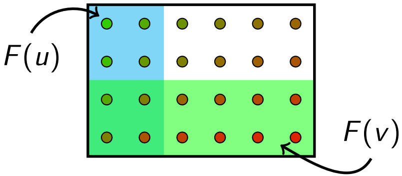
Related Work

- Bar-Noy, Motwani, and Naor 1992: No random online algorithm using $< 2\Delta - 1$ colors can guarantee success with probability $\geq \frac{1}{e}$.
- Bhattacharya, Grandoni, and Wajc 2021: Online $(1 + o(1))\Delta$ coloring algorithm that succeeds whp in random order $\Delta = \omega(\log n)$ setting.
- Blikstad et al. 2024b: Provides an online $(1 + o(1))\Delta$ coloring algorithm that succeeds with high probability when $\Delta = \omega(\log n)$.
- Blikstad et al. 2024a: Provides a deterministic $(\frac{e}{e-1} + o(1))$-competitive online bipartite edge-coloring algorithm under one-sided vertex arrivals when $\Delta = \omega(\log n)$.

Key Idea

- Intuition from tree case: for all pairs u, v, we want the sets $F(u)$, $F(v)$ to look independent. That is, we would like

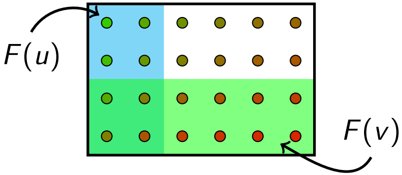
$$|F(u) \cap F(v)| \approx \frac{|F(u)||F(v)|}{(1+\epsilon)\Delta}$$



Key Idea

- Intuition from tree case: for all pairs u, v, we want the sets $F(u)$, $F(v)$ to look independent. That is, we would like

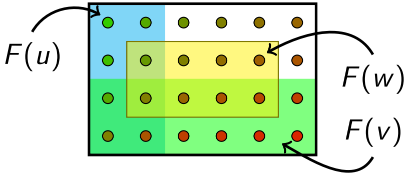
$$|F(u) \cap F(v)| \approx \frac{|F(u)||F(v)|}{(1+\epsilon)\Delta}$$



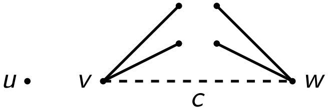
- It suffices if at each step, the color c assigned to an edge of v satisfies

$$\Pr[c \in F(u)] \approx \frac{|F(u) \cap F(v)|}{|F(v)|}$$

Key Idea

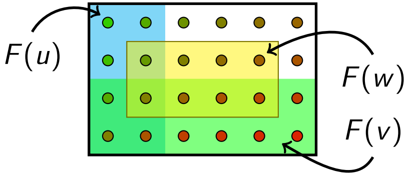


- Reality:



$$\Pr[c \in F(u)] = \frac{|F(u) \cap F(w) \cap F(v)|}{|F(v) \cap F(w)|}$$

Key Idea



- Goal: for all v, u, w, show:

$$\frac{|F(u) \cap F(w) \cap F(v)|}{|F(v) \cap F(w)|} \approx \frac{|F(u) \cap F(v)|}{|F(v)|}.$$

Key Idea

- Goal: for all v, u, w, show:

$$\frac{|F(u) \cap F(w) \cap F(v)|}{|F(v) \cap F(w)|} \approx \frac{|F(u) \cap F(v)|}{|F(v)|}.$$

- Note: it is enough to show that for all colors sets S and vertices w,

$$\frac{|F(w) \cap S|}{|F(w)|} \approx \frac{|S|}{(1+\epsilon)\Delta}. \qquad (*)$$

Key Idea

- Goal: for all v, u, w, show:

$$\frac{|F(u) \cap F(w) \cap F(v)|}{|F(v) \cap F(w)|} \approx \frac{|F(u) \cap F(v)|}{|F(v)|}.$$

- Note: it is enough to show that for all colors sets S and vertices w,

$$\frac{|F(w) \cap S|}{|F(w)|} \approx \frac{|S|}{(1+\epsilon)\Delta}. \qquad (*)$$

  - Then take $S' = F(v)$, $S'' = F(u) \cap F(v)$ to get above expression.

Key Idea

- Goal: for all v, u, w, show:

$$\frac{|F(u) \cap F(w) \cap F(v)|}{|F(v) \cap F(w)|} \approx \frac{|F(u) \cap F(v)|}{|F(v)|}.$$

- Note: it is enough to show that for all colors sets S and vertices w,

$$\frac{|F(w) \cap S|}{|F(w)|} \approx \frac{|S|}{(1+\epsilon)\Delta}. \qquad (*)$$

  - Then take $S' = F(v)$, $S'' = F(u) \cap F(v)$ to get above expression.
  - Unfortunately, this is clearly impossible to show (consider $S = F(w)$.)

Key Idea

- Goal: for all v, u, w, show:

$$\frac{|F(u) \cap F(w) \cap F(v)|}{|F(v) \cap F(w)|} \approx \frac{|F(u) \cap F(v)|}{|F(v)|}.$$

- Note: it is enough to show that for all colors sets S and vertices w,

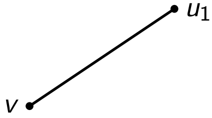
$$\frac{|F(w) \cap S|}{|F(w)|} \approx \frac{|S|}{(1+\epsilon)\Delta}. \qquad (*)$$

  - Then take $S' = F(v)$, $S'' = F(u) \cap F(v)$ to get above expression.
  - Unfortunately, this is clearly impossible to show (consider $S = F(w)$.)
- What we can show: with high probability, for all S, all vertices w excepting constantly many satisfy $(*)$.

Tracking $|F(v) \cap S|$

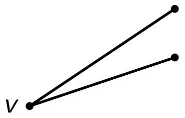
$v \bullet$

Tracking $|F(v) \cap S|$



$\Pr[c_1 \in S] = \frac{|F_0(u_1) \cap F_0(v) \cap S|}{|F_0(u_1) \cap F_0(v)|} = \frac{|F_0(v) \cap S|}{|F_0(v)|} \pm \widehat{\epsilon}_1$

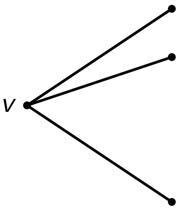
Tracking $|F(v) \cap S|$



$$u_1 \quad \Pr[c_1 \in S] = \frac{|F_0(u_1) \cap F_0(v) \cap S|}{|F_0(u_1) \cap F_0(v)|} = \frac{|F_0(v) \cap S|}{|F_0(v)|} \pm \widehat{\epsilon}_1$$

$$u_2 \quad \Pr[c_2 \in S] = \frac{|F_1(u_2) \cap F_1(v) \cap S|}{|F_1(u_2) \cap F_1(v)|} = \frac{|F_1(v) \cap S|}{|F_1(v)|} \pm \widehat{\epsilon}_2$$

Tracking $|F(v) \cap S|$



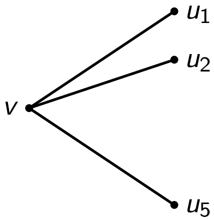
u_1 $\Pr[c_1 \in S] = \frac{|F_0(u_1) \cap F_0(v) \cap S|}{|F_0(u_1) \cap F_0(v)|} = \frac{|F_0(v) \cap S|}{|F_0(v)|} \pm \widehat{\epsilon}_1$

u_2 $\Pr[c_2 \in S] = \frac{|F_1(u_2) \cap F_1(v) \cap S|}{|F_1(u_2) \cap F_1(v)|} = \frac{|F_1(v) \cap S|}{|F_1(v)|} \pm \widehat{\epsilon}_2$

$\vdots$

u_5 $\Pr[c_5 \in S] = \frac{|F_4(u_5) \cap F_4(v) \cap S|}{|F_4(u_5) \cap F_4(v)|} = \frac{|F_4(v) \cap S|}{|F_4(v)|} \pm \widehat{\epsilon}_5$

Tracking $|F(v) \cap S|$



$$\Pr[c_1 \in S] = \frac{|F_0(v) \cap S|}{|F_0(v)|} \pm \widehat{\epsilon}_1$$

$$\Pr[c_2 \in S] = \frac{|F_1(v) \cap S|}{|F_1(v)|} \pm \widehat{\epsilon}_2$$

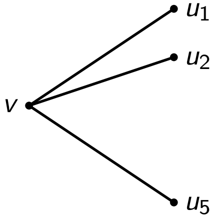
$$\vdots$$

$$\Pr[c_5 \in S] = \frac{|F_4(v) \cap S|}{|F_4(v)|} \pm \widehat{\epsilon}_5$$

- Let $X_i(S)$ indicate whether $c_i \in S$ and $p_i(S) = \Pr[c_i \in S]$

$$|S \cap \overline{F(v)}| = \sum X_i(S) \quad = \sum (X_i(S) - p_i(S)) + \sum \frac{|F_{i-1}(v) \cap S|}{|F_{i-1}(v)|} + \sum \widehat{\epsilon}_i$$

Tracking $|F(v) \cap S|$



$$\Pr[c_1 \in S] = \frac{|F_0(v) \cap S|}{|F_0(v)|} \pm \widehat{\epsilon}_1$$

$$\Pr[c_2 \in S] = \frac{|F_1(v) \cap S|}{|F_1(v)|} \pm \widehat{\epsilon}_2$$

$$\vdots$$

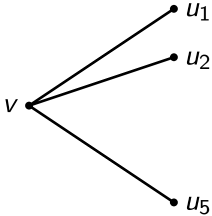
$$\Pr[c_5 \in S] = \frac{|F_4(v) \cap S|}{|F_4(v)|} \pm \widehat{\epsilon}_5$$

- Let $X_i(S)$ indicate whether $c_i \in S$ and $p_i(S) = \Pr[c_i \in S]$

$$|S \cap \overline{F(v)}| = \sum X_i(S) \quad = \sum (X_i(S) - p_i(S)) + \sum \frac{|F_{i-1}(v) \cap S|}{|F_{i-1}(v)|} + \sum \widehat{\epsilon}_i$$

- Two sources of divergence in our calculation of $|S \cap F(v)|$:

Tracking $|F(v) \cap S|$



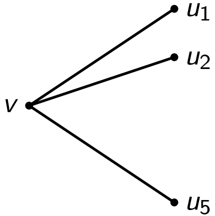
$$\begin{aligned}
\Pr[c_1 \in S] &= \tfrac{|F_0(v) \cap S|}{|F_0(v)|} \pm \widehat{\epsilon}_1 \\
\Pr[c_2 \in S] &= \tfrac{|F_1(v) \cap S|}{|F_1(v)|} \pm \widehat{\epsilon}_2 \\
&\vdots \\
\Pr[c_5 \in S] &= \tfrac{|F_4(v) \cap S|}{|F_4(v)|} \pm \widehat{\epsilon}_5
\end{aligned}$$

- Let $X_i(S)$ indicate whether $c_i \in S$ and $p_i(S) = \Pr[c_i \in S]$

$$|S \cap \overline{F(v)}| = \sum X_i(S) \quad = \sum (X_i(S) - p_i(S)) + \sum \frac{|F_{i-1}(v) \cap S|}{|F_{i-1}(v)|} + \sum \widehat{\epsilon}_i$$

- Two sources of divergence in our calculation of $|S \cap F(v)|$:
    1. The natural divergence between $X_i(S)$ and $p_i(S)$

Tracking $|F(v) \cap S|$



$$\Pr[c_1 \in S] = \frac{|F_0(v) \cap S|}{|F_0(v)|} \pm \widehat{\epsilon}_1$$
$$\Pr[c_2 \in S] = \frac{|F_1(v) \cap S|}{|F_1(v)|} \pm \widehat{\epsilon}_2$$
$$\vdots$$
$$\Pr[c_5 \in S] = \frac{|F_4(v) \cap S|}{|F_4(v)|} \pm \widehat{\epsilon}_5$$

- Let $X_i(S)$ indicate whether $c_i \in S$ and $p_i(S) = \Pr[c_i \in S]$

$$|S \cap \overline{F(v)}| = \sum X_i(S) \quad = \sum (X_i(S) - p_i(S)) + \sum \frac{|F_{i-1}(v) \cap S|}{|F_{i-1}(v)|} + \sum \widehat{\epsilon}_i$$

- Two sources of divergence in our calculation of $|S \cap F(v)|$:
  1. The natural divergence between $X_i(S)$ and $p_i(S)$
  2. The $\widehat{\epsilon}$ error terms from each neighbor

Future Directions

- Generalizing the dense case:
    - Current proof of the error bound for the dense case allows for worst case directions of the errors of all neighbors.
    - Can we improve our estimate of the collective error of sets of vertices in order to weaken the density requirement?

Future Directions

- Generalizing the dense case:
  - Current proof of the error bound for the dense case allows for worst case directions of the errors of all neighbors.
  - Can we improve our estimate of the collective error of sets of vertices in order to weaken the density requirement?
- Dynamic setting:
  - Adversary can insert or delete edges to graph
  - Want to minimize the number of changes made to the coloring after each insertion/deletion, in expectation

Future Directions

- Generalizing the dense case:
  - Current proof of the error bound for the dense case allows for worst case directions of the errors of all neighbors.
  - Can we improve our estimate of the collective error of sets of vertices in order to weaken the density requirement?
- Dynamic setting:
  - Adversary can insert or delete edges to graph
  - Want to minimize the number of changes made to the coloring after each insertion/deletion, in expectation
- Multigraphs:
  - In the case of multigraphs, instead of Vizing's theorem, we have Shannon's theorem giving an upper bound of $\frac{3\Delta}{2}$ for $\chi'(G)$.
  - How close can the random greedy algorithm get to this bound in the multigraph case?

Thank You

Bibliography I

Bar-Noy, Amotz, Rajeev Motwani, and Joseph Naor (1992). "The Greedy Algorithm is Optimal for On-Line Edge Coloring". In: Inf. Process. Lett. 44.5, pp. 251–253. DOI: 10.1016/0020-0190(92)90209-E. URL: https://doi.org/10.1016/0020-0190(92)90209-E.

Bhattacharya, Sayan, Fabrizio Grandoni, and David Wajc (2021). "Online Edge Coloring Algorithms via the Nibble Method". In: Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, S Ed. by Dániel Marx. SIAM, pp. 2830–2842. DOI: 10.1137/1.9781611976465.168.

Blikstad, Joakim et al. (2024a). "Deterministic Online Bipartite Edge Coloring". In: arXiv preprint arXiv:2406.13000. arXiv: 2408.03661 [cs.DS].

– (2024b). "Online Edge Coloring is (Nearly) as Easy as Offline". In: CoRR abs/2402.18339. DOI: 10.48550/ARXIV.2402.18339. arXiv: 2402.18339.

Bibliography II

Feder, Tomás, Rajeev Motwani, and Rina Panigrahy (n.d.). "Online Algorithms for Edge Coloring". In: (). Available at: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e300284ac75ed317c6550bea907ee9ef862d2c19.