Fast Algorithms for $(\Delta + 1)$-Edge-Coloring

Joint work with Anton Bernshteyn

Abhishek Dhawan

School of Mathematics
Georgia Institute of Technology

34th Midwestern Conference on Combinatorics and Combinatorial Computing
October 21, 2022

Table of Contents

Table of Contents

Vizing's Theorem

Recall a proper k-**edge-coloring** is a function $\varphi : E(G) \to [k]$ such that $\varphi(e) \neq \varphi(f)$ when e and f share an endpoint.
The **chromatic index** of a graph (denoted $\chi'(G)$) is the minimum value k such that G admits a proper k-edge-coloring.

Vizing's Theorem

Recall a proper k-**edge-coloring** is a function $\varphi : E(G) \to [k]$ such that $\varphi(e) \neq \varphi(f)$ when e and f share an endpoint.

The **chromatic index** of a graph (denoted $\chi'(G)$) is the minimum value k such that G admits a proper k-edge-coloring.

The following renowned result of Vizing provides a general bound on $\chi'(G)$.

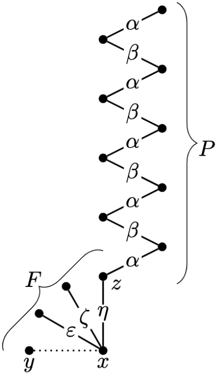
Theorem (Vizing 1964)

For any simple graph G, $\chi'(G) \in \{\Delta(G), \Delta(G) + 1\}$.

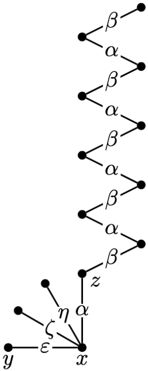
Vizing's Theorem

Theorem (Vizing 1964)

For any simple graph G, $\chi'(G) \in \{\Delta(G), \Delta(G)+1\}$.



(a) Before shifting.



(b) After shifting.

Sequential Algorithms

Theorem

- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $O(mn)$ *time (Bollobás 1984; Rao, Dijkstra 1992; Misra, Gries 1992).*

Sequential Algorithms

Theorem

- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $O(mn)$ *time (Bollobás 1984; Rao, Dijkstra 1992; Misra, Gries 1992).*
- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $O(m\sqrt{n \log n})$ *time (Gabow, Nishizeki, Kariv, Leven, Terada 1985).*

Sequential Algorithms

Theorem

- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $O(mn)$ *time (Bollobás 1984; Rao, Dijkstra 1992; Misra, Gries 1992).*
- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $O(m\sqrt{n \log n})$ *time (Gabow, Nishizeki, Kariv, Leven, Terada 1985).*
- *There is a randomized* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $O(m\sqrt{n})$ *time with probability at least* $1 - e^{-\sqrt{m}}$ *(Sinnamon 2019).*

Sequential Algorithms

Theorem

- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $O(mn)$ *time (Bollobás 1984; Rao, Dijkstra 1992; Misra, Gries 1992).*
- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $O(m\sqrt{n \log n})$ *time (Gabow, Nishizeki, Kariv, Leven, Terada 1985).*
- *There is a randomized* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $O(m\sqrt{n})$ *time with probability at least* $1 - e^{-\sqrt{m}}$ *(Sinnamon 2019).*
- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $O(m\sqrt{n})$ *time (Sinnamon 2019).*

Sequential Algorithms

Theorem

- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\, n^2$ *time (Bollobás 1984; Rao, Dijkstra 1992; Misra, Gries 1992).*
- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\, n^{3/2}\sqrt{\log n}$ *time (Gabow, Nishizeki, Kariv, Leven, Terada 1985).*
- *There is a randomized* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\, n^{3/2}$ *time with probability at least* $1 - e^{-\sqrt{n}}$ *(Sinnamon 2019).*
- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\, n^{3/2}$ *time (Sinnamon 2019).*

Sequential Algorithms

Theorem

- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\, n^2$ *time (Bollobás 1984; Rao, Dijkstra 1992; Misra, Gries 1992).*
- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\, n^{3/2}\sqrt{\log n}$ *time (Gabow, Nishizeki, Kariv, Leven, Terada 1985).*
- *There is a randomized* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\, n \log n$ *time with probability at least* $1 - 1/\mathrm{poly}(n)$ *(Bernshteyn, D. 2022+).*
- *There is a deterministic* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\, n^{3/2}$ *time (Sinnamon 2019).*

Locality of Vizing's Theorem

Definition (Augmenting Subgraphs)

Given a partial coloring φ, we say a subgraph $H \subseteq G$ is **augmenting** if there is at least one uncolored edge in H and there is a proper coloring ψ that can be obtained by modifying φ on the edges in H such that every edge in H is now colored.

Locality of Vizing's Theorem

Definition (Augmenting Subgraphs)

Given a partial coloring φ, we say a subgraph $H \subseteq G$ is **augmenting** if there is at least one uncolored edge in H and there is a proper coloring ψ that can be obtained by modifying φ on the edges in H such that every edge in H is now colored.

For an arbitrary n-vertex graph G of maximum degree Δ and partial $(\Delta + 1)$-edge-coloring φ, how small can H be?

Locality of Vizing's Theorem

Definition (Augmenting Subgraphs)

Given a partial coloring φ, we say a subgraph $H \subseteq G$ is **augmenting** if there is at least one uncolored edge in H and there is a proper coloring ψ that can be obtained by modifying φ on the edges in H such that every edge in H is now colored.

For an arbitrary n-vertex graph G of maximum degree Δ and partial $(\Delta + 1)$-edge-coloring φ, how small can H be?

Theorem

- *There exists G, φ such that $e(H) = \Omega(\Delta \log n)$ for every augmenting subgraph H (Chang, He, Li, Pettie, Uitto 2018).*

Locality of Vizing's Theorem

Definition (Augmenting Subgraphs)

Given a partial coloring φ, we say a subgraph $H \subseteq G$ is **augmenting** if there is at least one uncolored edge in H and there is a proper coloring ψ that can be obtained by modifying φ on the edges in H such that every edge in H is now colored.

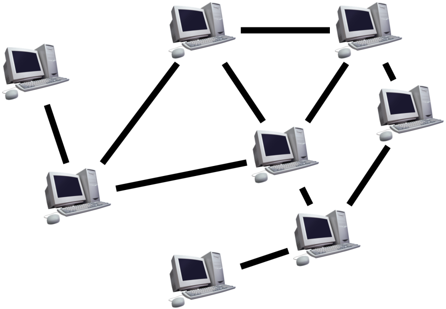
For an arbitrary n-vertex graph G of maximum degree Δ and partial $(\Delta + 1)$-edge-coloring φ, how small can H be?

Theorem

- *There exists G, φ such that $e(H) = \Omega(\Delta \log n)$ for every augmenting subgraph H (Chang, He, Li, Pettie, Uitto 2018).*
- *For every G, φ there exists an augmenting subgraph H such that $e(H) \leq \mathrm{poly}(\Delta)(\log n)^2$ (Bernshteyn 2021).*

Distributed Algorithms

A distributed algorithm in the LOCAL model occurs in rounds, where in each round, every vertex performs some local computation and then broadcasts its results to its neighbors in G.



Distributed Algorithms

Theorem

- *There is a randomised distributed* $(\Delta + \tilde{O}(\sqrt{\Delta}))$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta, \log\log n)$ *rounds (Chang, He, Li, Pettie, and Uitto 2018).*

Distributed Algorithms

Theorem

- *There is a randomised distributed* $(\Delta + \tilde{O}(\sqrt{\Delta}))$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta, \log\log n)$ *rounds (Chang, He, Li, Pettie, and Uitto 2018).*
- *There is a deterministic distributed* $\lfloor 3\Delta/2 \rfloor$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\,(\log n)^8$ *rounds (Ghaffari, Kuhn, Maus, and Uitto 2018).*

Distributed Algorithms

Theorem

- *There is a randomised distributed* $(\Delta + \tilde{O}(\sqrt{\Delta}))$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta, \log\log n)$ *rounds (Chang, He, Li, Pettie, and Uitto 2018).*
- *There is a deterministic distributed* $\lfloor 3\Delta/2 \rfloor$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\,(\log n)^8$ *rounds (Ghaffari, Kuhn, Maus, and Uitto 2018).*
- *There is a randomised distributed* $(\Delta + 2)$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\,(\log n)^3$ *rounds (Su, Vu 2019).*

Distributed Algorithms

Theorem

- *There is a randomised distributed* $(\Delta + \tilde{O}(\sqrt{\Delta}))$*-edge-coloring algorithm that runs in* $\text{poly}(\Delta, \log\log n)$ *rounds (Chang, He, Li, Pettie, and Uitto 2018).*
- *There is a deterministic distributed* $\lfloor 3\Delta/2 \rfloor$*-edge-coloring algorithm that runs in* $\text{poly}(\Delta)\,(\log n)^8$ *rounds (Ghaffari, Kuhn, Maus, and Uitto 2018).*
- *There is a randomised distributed* $(\Delta + 2)$*-edge-coloring algorithm that runs in* $\text{poly}(\Delta)\,(\log n)^3$ *rounds (Su, Vu 2019).*
- *There is a randomised distributed* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $\text{poly}(\Delta)\,(\log n)^5$ *rounds (Bernshteyn 2021).*

Distributed Algorithms

Theorem

- *There is a randomised distributed* $(\Delta + \tilde{O}(\sqrt{\Delta}))$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta, \log\log n)$ *rounds (Chang, He, Li, Pettie, and Uitto 2018).*
- *There is a deterministic distributed* $\lfloor 3\Delta/2 \rfloor$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\,(\log n)^8$ *rounds (Ghaffari, Kuhn, Maus, and Uitto 2018).*
- *There is a randomised distributed* $(\Delta + 2)$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\,(\log n)^3$ *rounds (Su, Vu 2019).*
- *There is a randomised distributed* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta)\,(\log n)^5$ *rounds (Bernshteyn 2021).*
- *There is a deterministic distributed* $(\Delta + 1)$*-edge-coloring algorithm that runs in* $\mathrm{poly}(\Delta, \log\log n)\,(\log n)^{11}$ *rounds (Bernshteyn 2021).*

Our Results

Theorem (Bernshteyn, D. 2022+)

Let G be an n-vertex graph of maximum degree Δ and let φ be a partial $(\Delta + 1)$-edge-coloring. For any uncolored edge e, there is an augmenting subgraph H containing e such that $e(H) \leq \text{poly}(\Delta) \log n$.

Our Results

Theorem (Bernshteyn, D. 2022+)

There exists a randomized (resp. deterministic) distributed algorithm in the LOCAL model that computes a proper $(\Delta + 1)$*-edge-coloring of an* n*-vertex graph of maximum degree* Δ *in* $\mathrm{poly}(\Delta)(\log n)^3$ *rounds (resp.* $\mathrm{poly}(\Delta, \log\log n)(\log n)^6$*).*

Theorem (Bernshteyn, D. 2022+)

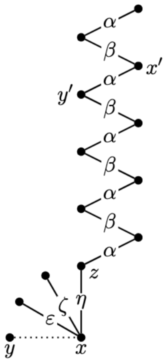
There exists a randomized sequential algorithm that computes a proper $(\Delta + 1)$*-edge-coloring of an* n*-vertex graph of maximum degree* Δ *in poly*$(\Delta)\, n$ *time with probability at least* $1 - 1/\Delta^n$*.*

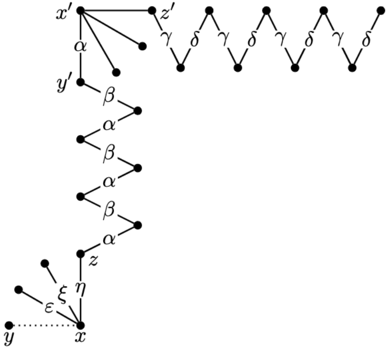
Table of Contents

Non-Intersecting Vizing Chains

Idea (Grebík, Pikhurko 2020; Bernshteyn 2021)

- If P is too long, consider a random initial segment of P, shift and try again.

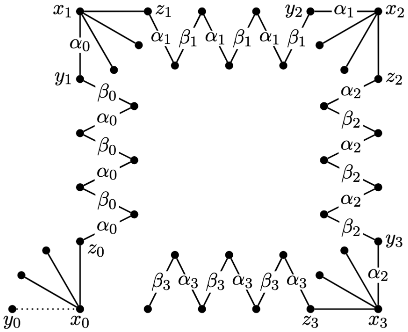




Non-Intersecting Vizing Chains

Idea (Grebík, Pikhurko 2020; Bernshteyn 2021)

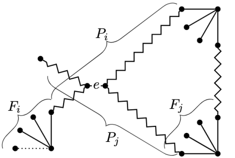
- If P is too long, consider a random initial segment of P, shift and try again.
- Repeat until the final path is short.



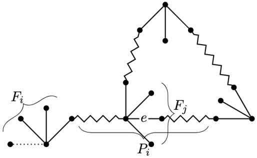
Non-Intersecting Vizing Chains

Idea (Grebík, Pikhurko 2020; Bernshteyn 2021)

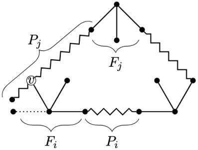
- If P is too long, consider a random initial segment of P, shift and try again.
- Repeat until the final path is short, while guaranteeing the chain is non-intersecting.



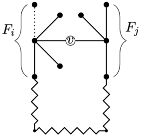
(a) Intersection between P_i, P_j at an edge e



(b) Intersection between P_i, F_j at an edge e



(c) Intersection between F_i, P_j at a vertex v.



(d) Intersection between F_i, F_j at a vertex v

The Multi-Step Vizing Algorithm

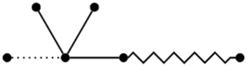
- Input: A graph G, a partial coloring φ, an uncolored edge $e = xy$, a vertex $x \in e$ and a parameter $\ell \in \mathbb{N}$.
- Output: A non-intersecting augmenting multi-step Vizing chain C such that $\mathsf{Start}(C) = xy$.

The Multi-Step Vizing Algorithm

- Input: A graph G, a partial coloring φ, an uncolored edge $e = xy$, a vertex $x \in e$ and a parameter $\ell \in \mathbb{N}$.
- Output: A non-intersecting augmenting multi-step Vizing chain C such that $\mathsf{Start}(C) = xy$.

Step 1: First Chain

Find the first Vizing Chain $F + P$ on C.

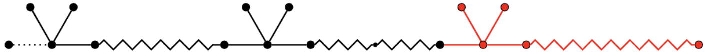


The Multi-Step Vizing Algorithm

- Input: A graph G, a partial coloring φ, an uncolored edge $e = xy$, a vertex $x \in e$ and a parameter $\ell \in \mathbb{N}$.
- Output: A non-intersecting augmenting multi-step Vizing chain C such that $\mathsf{Start}(C) = xy$.

Step 2: Iterate

Let C (in black) and $F + P$ (in red) be the current chain and candidate chain at the start of the iteration.



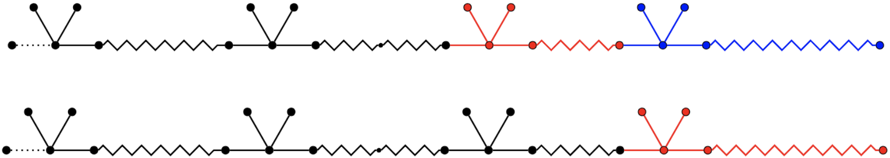
The Multi-Step Vizing Algorithm

- Input: A graph G, a partial coloring φ, an uncolored edge $e = xy$, a vertex $x \in e$ and a parameter $\ell \in \mathbb{N}$.
- Output: A non-intersecting augmenting multi-step Vizing chain C such that $\mathsf{Start}(C) = xy$.

Step 2: Iterate

Let C (in black) and $F + P$ (in red) be the current chain and candidate chain after truncation. Let $\tilde{F} + \tilde{P}$ (in blue) be the next step chain.

Case 1: Non-intersecting.



The Multi-Step Vizing Algorithm

- Input: A graph G, a partial coloring φ, an uncolored edge $e = xy$, a vertex $x \in e$ and a parameter $\ell \in \mathbb{N}$.
- Output: A non-intersecting augmenting multi-step Vizing chain C such that $\mathsf{Start}(C) = xy$.

Step 2: Iterate

Let C (in black) and $F + P$ (in red) be the current chain and candidate chain after truncation. Let $\tilde{F} + \tilde{P}$ (in blue) be the next step chain.

Case 2: Intersecting.

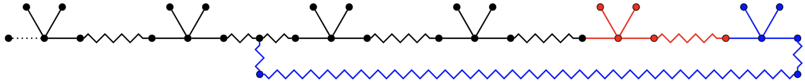


Table of Contents

Termination of the Multi-Step Vizing Algorithm

Theorem (Bernshteyn, D. 2022+)

For $\ell = \text{poly}(\Delta)$ *of large enough degree, the Multi-Step Vizing Algorithm terminates in* $O(\log n)$ *steps with probability at least* $1 - 1/\text{poly}(n)$*.*

Termination of the Multi-Step Vizing Algorithm

We utilise the entropy compression argument developed by Moser and Tardos in 2010.

Termination of the Multi-Step Vizing Algorithm

We utilise the entropy compression argument developed by Moser and Tardos in 2010.

Some Notable Applications

- For every $n \geq 1$ and a sequence of sets $L_1, \ldots, L_n$ of size at least 4, there is a non-repetitive sequence chosen from $L_1, \ldots, L_n$ (Grytczuk, Kozik, Micek 2011).
- For every graph G of maximum degree Δ, there is a non-repetitive coloring using at most $(1 + o(1))\Delta^2$ colors (Dujmović, Joret, Kozik, Wood 2014; Rosenfeld 2020).
- The list chromatic number of triangle free graphs is at most $(1 + o(1))\Delta / \log \Delta$ asymptotically (Molloy 2017).

Termination of the Multi-Step Vizing Algorithm

- Suppose your process runs for t steps making random choices $\omega_1, \ldots, \omega_t$.

Termination of the Multi-Step Vizing Algorithm

- Suppose your process runs for t steps making random choices $\omega_1, \dots, \omega_t$.
- Define a record for each step $r_1, \dots, r_t$ such that given the record and the output $\mathcal{O}_t$ after t steps, we can recover $\omega_1, \dots, \omega_t$ as well as the outputs $\mathcal{O}_j$ for each $1 \leq j < t$.

Termination of the Multi-Step Vizing Algorithm

- Suppose your process runs for t steps making random choices $\omega_1, \ldots, \omega_t$.
- Define a record for each step $r_1, \ldots, r_t$ such that given the record and the output $\mathcal{O}_t$ after t steps, we can recover $\omega_1, \ldots, \omega_t$ as well as the outputs $\mathcal{O}_j$ for each $1 \leq j < t$.
- We have

$$\mathbb{P}[\text{Failure in } t \text{ steps}] \leq \frac{\#(\text{record,output})\text{-pairs}}{\text{Amount of randomness generated}}$$

Termination of the Multi-Step Vizing Algorithm

- Suppose your process runs for t steps making random choices $\omega_1, \dots, \omega_t$.
- Define a record for each step $r_1, \dots, r_t$ such that given the record and the output $\mathcal{O}_t$ after t steps, we can recover $\omega_1, \dots, \omega_t$ as well as the outputs $\mathcal{O}_j$ for each $1 \le j < t$.
- We have

$$\mathbb{P}[\text{Failure in } t \text{ steps}] \le \frac{\#\text{(record,output)-pairs}}{\ell^t}$$

Termination of the Multi-Step Vizing Algorithm

The Record

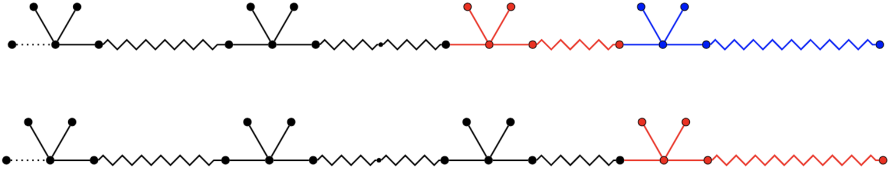
Each entry of our record is of the form (d_i, k_i). Assuming the process doesn't terminate, we define the i-th entry as follows:

Termination of the Multi-Step Vizing Algorithm

The Record

Each entry of our record is of the form (d_i, k_i). Assuming the process doesn't terminate, we define the i-th entry as follows:

1. If the second step chain found is non-intersecting, we let $d_i = 1$, $k_i =$ blank.

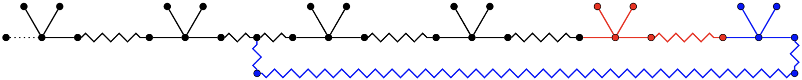


Termination of the Multi-Step Vizing Algorithm

The Record

Each entry of our record is of the form (d_i, k_i). Assuming the process doesn't terminate, we define the i-th entry as follows:

1. If the second step chain found is non-intersecting, we let $d_i = 1$, $k_i =$ blank.
2. If the second step chain found intersects C, we let $d_i = -k$ and let k_i denote the $(k+1)$-step chain causing the truncation.



Termination of the Multi-Step Vizing Algorithm

A subtle counting argument shows that the number of k-step chains ending at a vertex x is at most $\mathrm{poly}(\Delta)^k$. This observation helps in bounding the number of (record, output)-pairs.

Lemma

Assuming the process doesn't terminate, there are at most $n(\mathrm{poly}(\Delta)\,\ell)^{t/2}$ (record, output)-pairs after t steps of the Multi-Step Vizing Algorithm.

Termination of the Multi-Step Vizing Algorithm

A subtle counting argument shows that the number of k-step chains ending at a vertex x is at most $\mathrm{poly}(\Delta)^k$. This observation helps in bounding the number of (record, output)-pairs.

Lemma

Assuming the process doesn't terminate, there are at most $n(\mathrm{poly}(\Delta)\,\ell)^{t/2}$ (record, output)-pairs after t steps of the Multi-Step Vizing Algorithm.

It follows that

$$\mathbb{P}[\mathrm{Failure}] \leq \frac{n(\mathrm{poly}(\Delta)\,\ell)^{t/2}}{\ell^t}.$$

For $\ell \sim \mathrm{poly}(\Delta)$ and $t \sim \log n$, the above is at most $1/\mathrm{poly}(n)$.

Table of Contents

Key Takeaways

1. The upper bound on the size of an augmenting subgraph matches the lower bound asymptotically.

Key Takeaways

1. The upper bound on the size of an augmenting subgraph matches the lower bound asymptotically.
2. Runtime of the distributed algorithms matches those for $(\Delta + 2)$-edge-coloring.

Key Takeaways

1. The upper bound on the size of an augmenting subgraph matches the lower bound asymptotically.
2. Runtime of the distributed algorithms matches those for $(\Delta + 2)$-edge-coloring.
3. The sequential coloring algorithm matches the lower bound asymptotically.

Key Takeaways

1. The upper bound on the size of an augmenting subgraph matches the lower bound asymptotically.
2. Runtime of the distributed algorithms matches those for $(\Delta + 2)$-edge-coloring.
3. The sequential coloring algorithm matches the lower bound asymptotically.
4. Moreover, the result implies that the average size of an augmenting subgraph found using the Multi-Step Vizing Algorithm is independent of n.

Key Takeaways

1. The upper bound on the size of an augmenting subgraph matches the lower bound asymptotically.
2. Runtime of the distributed algorithms matches those for $(\Delta + 2)$-edge-coloring.
3. The sequential coloring algorithm matches the lower bound asymptotically.
4. Moreover, the result implies that the average size of an augmenting subgraph found using the Multi-Step Vizing Algorithm is independent of n.
5. Unconventional use of the entropy compression argument.

Thank you!