Coloring a Mostly Forgotten Graph

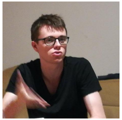
distributed coloring with your voice low and your brain small

Alexandre Nolin

CISPA Helmholtz Center for Information Security

WAND@DISC 2024 – Madrid, 01.11.2024

Based on joint work with



Maxime Flin

Mohsen Ghaffari

Magnús M. Halldórsson

Fabian Kuhn

Tigran Tonoyan

I was looking at the date today and...

I was looking at the date today and...

It is a public holiday in

I was looking at the date today and...

It is a public holiday in

1. Spain, where we are,

I was looking at the date today and...

It is a public holiday in

1. Spain, where we are,
2. Many parts of Germany, where my affiliation is and the workshop chair is from,

I was looking at the date today and...

It is a public holiday in

1. Spain, where we are,
2. Many parts of Germany, where my affiliation is and the workshop chair is from,
3. France, where I'm from,

I was looking at the date today and...

It is a public holiday in

1. Spain, where we are,
2. Many parts of Germany, where my affiliation is and the workshop chair is from,
3. France, where I'm from,
4. Italy, where WAND organizers seem to be from, and two out of three are affiliated,

I was looking at the date today and...

It is a public holiday in

1. Spain, where we are,
2. Many parts of Germany, where my affiliation is and the workshop chair is from,
3. France, where I'm from,
4. Italy, where WAND organizers seem to be from, and two out of three are affiliated,
5. Austria, where the workshop chair and the last organizer of WAND are affiliated.

I was looking at the date today and...

It is a public holiday in

1. Spain, where we are,
2. Many parts of Germany, where my affiliation is and the workshop chair is from,
3. France, where I'm from,
4. Italy, where WAND organizers seem to be from, and two out of three are affiliated,
5. Austria, where the workshop chair and the last organizer of WAND are affiliated.

And we are all gathered here

I was looking at the date today and...

It is a public holiday in

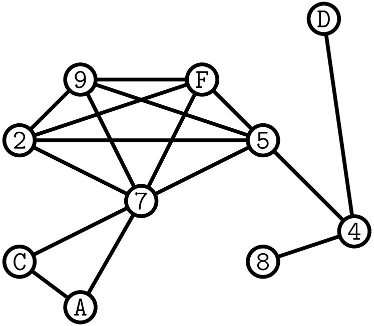
1. Spain, where we are,
2. Many parts of Germany, where my affiliation is and the workshop chair is from,
3. France, where I'm from,
4. Italy, where WAND organizers seem to be from, and two out of three are affiliated,
5. Austria, where the workshop chair and the last organizer of WAND are affiliated.

And we are all gathered here

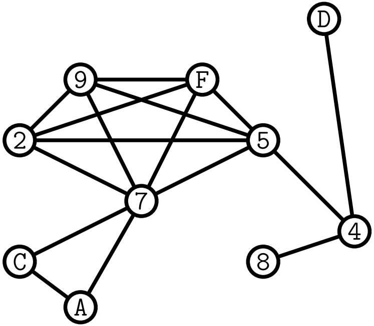
Enjoy your holiday talk ☺

LOCAL and CONGEST

LOCAL



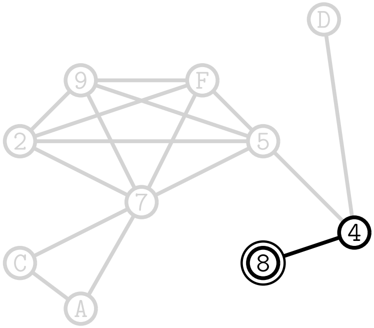
CONGEST



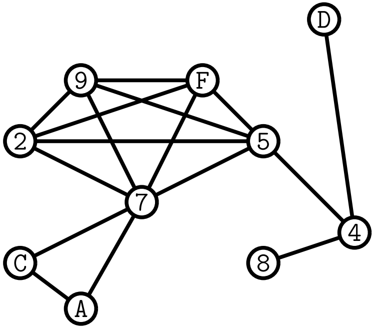
- Both: Synchronous message passing, graph models communication network.
- LOCAL: ∞-sized messages.
- CONGEST: $O(\log n)$-sized messages (n upper bound on number of nodes).

LOCAL and CONGEST

LOCAL



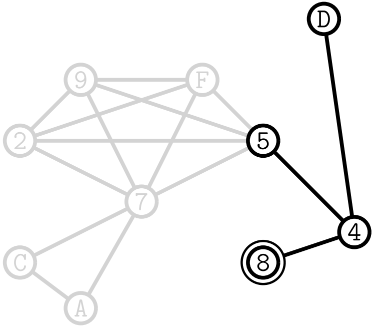
CONGEST



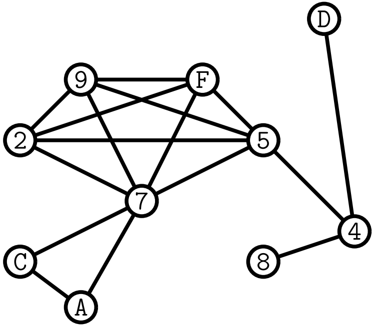
- Both: Synchronous message passing, graph models communication network.
- LOCAL: ∞-sized messages.
- CONGEST: $O(\log n)$-sized messages (n upper bound on number of nodes).

LOCAL and CONGEST

LOCAL



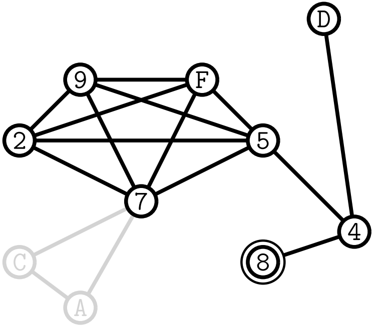
CONGEST



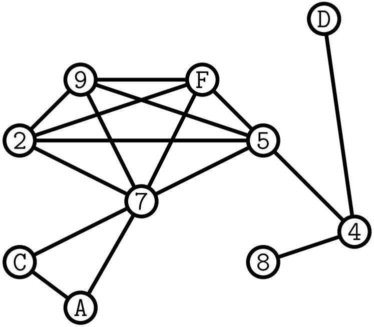
- Both: Synchronous message passing, graph models communication network.
- LOCAL: ∞-sized messages.
- CONGEST: $O(\log n)$-sized messages (n upper bound on number of nodes).

Local and Congest

Local



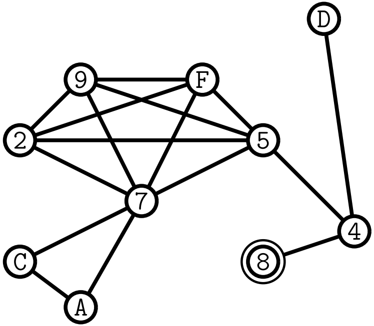
Congest



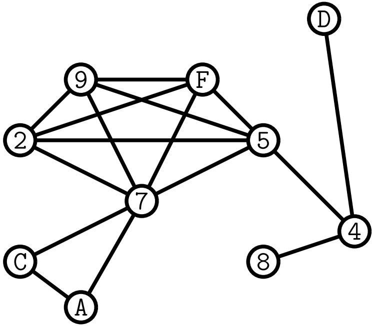
- Both: Synchronous message passing, graph models communication network.
- Local: ∞-sized messages.
- Congest: $O(\log n)$-sized messages (n upper bound on number of nodes).

Local and Congest

Local

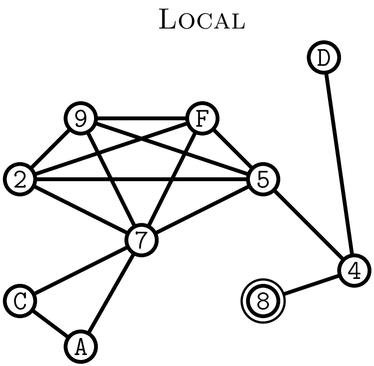


Congest



- Both: Synchronous message passing, graph models communication network.
- Local: ∞-sized messages.
- Congest: $O(\log n)$-sized messages (n upper bound on number of nodes).

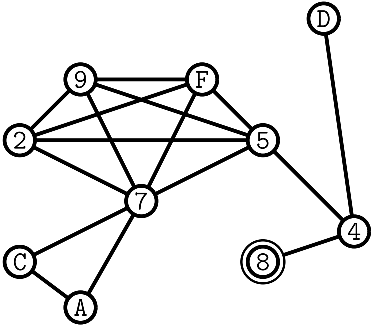
LOCAL and CONGEST



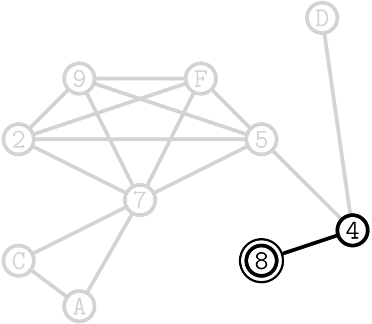
- Both: Synchronous message passing, graph models communication network.
- LOCAL: ∞-sized messages.
- CONGEST: $O(\log n)$-sized messages (n upper bound on number of nodes).

Local and Congest

Local

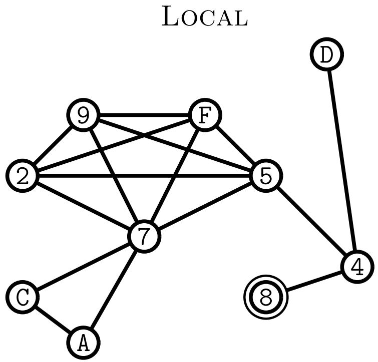


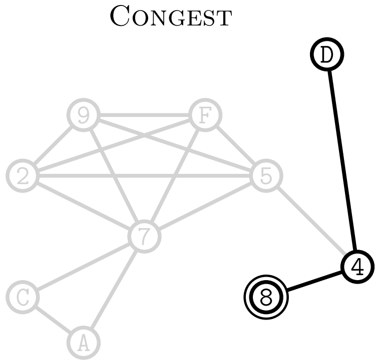
Congest



- Both: Synchronous message passing, graph models communication network.
- Local: ∞-sized messages.
- Congest: $O(\log n)$-sized messages (n upper bound on number of nodes).

Local and Congest

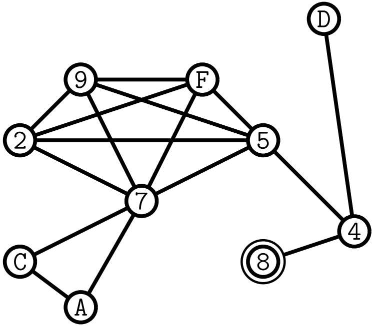




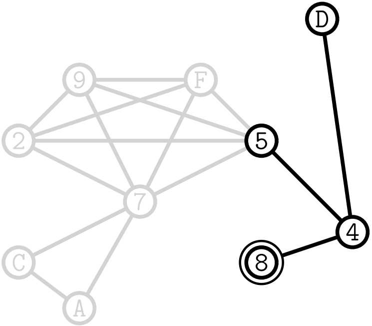
- Both: Synchronous message passing, graph models communication network.
- Local: ∞-sized messages.
- Congest: $O(\log n)$-sized messages (n upper bound on number of nodes).

LOCAL and CONGEST

LOCAL

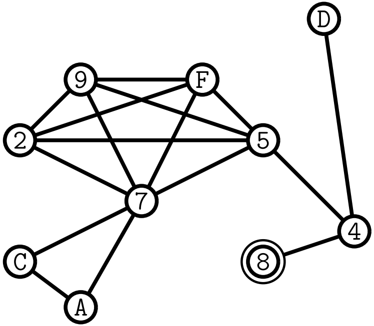


CONGEST

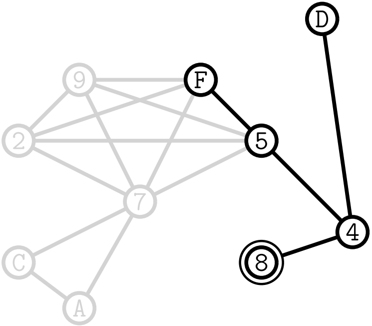


- Both: Synchronous message passing, graph models communication network.
- LOCAL: ∞-sized messages.
- CONGEST: $O(\log n)$-sized messages (n upper bound on number of nodes).

LOCAL

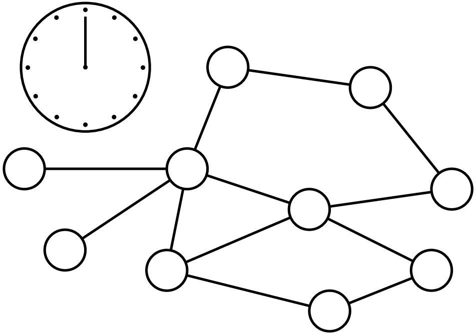


CONGEST



- Both: Synchronous message passing, graph models communication network.
- LOCAL: ∞-sized messages.
- CONGEST: $O(\log n)$-sized messages (n upper bound on number of nodes).

Coloring problem



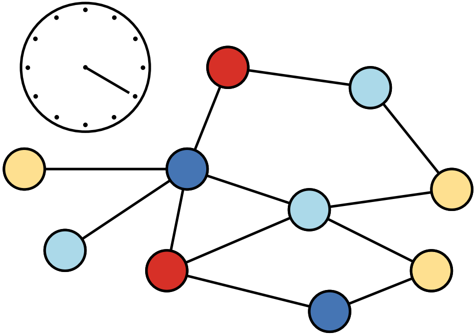
Goal: starting from an initially uncolored graph $G = (V, E)$, assign a color to each node s.t. adjacent nodes receive distinct colors.

Formally: compute an assignment φ giving colors to the nodes

$$\varphi : V \to \mathcal{C}$$

such that $\varphi(u) \neq \varphi(v)$ for each edge $uv \in E$.

Coloring problem



Goal: starting from an initially uncolored graph $G = (V, E)$, assign a color to each node s.t. adjacent nodes receive distinct colors.

Formally: compute an assignment φ giving colors to the nodes

$$\varphi : V \to \mathcal{C}$$

such that $\varphi(u) \neq \varphi(v)$ for each edge $uv \in E$.

Palette Sparsification: the general idea

Theorems of the form:

- Let each node $v \in V$ independently sample a list of colors $L(v) \subseteq \mathcal{C}$ according to some distribution $\mathcal{D}_v$.
- Then with probability at least p, we get a list-coloring instance with a solution, as is: G is colorable with the new constraints that $\varphi(v) \in L(v)$ for each $v \in V$.

Palette Sparsification: the general idea

Theorems of the form:

- Let each node $v \in V$ independently sample a list of colors $L(v) \subseteq \mathcal{C}$ according to some distribution $\mathcal{D}_v$.

- Then with probability at least p, we get a list-coloring instance with a solution, as is: G is colorable with the new constraints that $\varphi(v) \in L(v)$ for each $v \in V$.

Trivial versions (amuse-bouches):

- If the distributions are $\Pr[\mathcal{D}_v = \{1, ..., \Delta + 1\}] = 1$, then the graph is colorable with the lists $L(v)$ with probability 1.
- If the distributions are $\Pr[\mathcal{D}_v = \{i\}] = 1/(\deg(v) + 1)$ for each $i \in \{1, ..., \deg(v) + 1\}$, then the graph is colorable with the lists $L(v)$ with probability $> \Delta^{-n}$.

Palette Sparsification: the general idea

Theorems of the form:

- Let each node $v \in V$ independently sample a list of colors $L(v) \subseteq \mathcal{C}$ according to some distribution $\mathcal{D}_v$.
- Then with probability at least p, we get a list-coloring instance with a solution, as is: G is colorable with the new constraints that $\varphi(v) \in L(v)$ for each $v \in V$.

Trivial versions (amuse-bouches):

- If the distributions are $\Pr[\mathcal{D}_v = \{1, ..., \Delta + 1\}] = 1$, then the graph is colorable with the lists $L(v)$ with probability 1.
- If the distributions are $\Pr[\mathcal{D}_v = \{i\}] = 1/(\deg(v) + 1)$ for each $i \in \{1, ..., \deg(v) + 1\}$, then the graph is colorable with the lists $L(v)$ with probability $> \Delta^{-n}$.

Interesting zone is between those two extremes.

Examples of palette sparsification results

- Assadi, Chen, and Khanna [ACK19]:
  - $\Theta(\log n)$ colors from $\{1, \ldots, \Delta + 1\}$.
- Alon and Assadi [AA20]:
  - $\Theta(\varepsilon^{-1.5}\sqrt{\log n})$ colors from $\{1, \ldots, (1+\varepsilon)\Delta\}$
  - $\Theta(\log n)$ colors from $\{1, \ldots, \deg(v) + 1\}$.
  - $\Theta(\varepsilon^{-1} \log n)$ colors from arbitrary palettes $\Psi(v)$ of size $(1+\varepsilon)\deg(v)$.
  - $\Theta(\Delta^\gamma + \sqrt{\log n})$ colors from $\{1, \ldots, \frac{9\Delta}{\gamma \ln \Delta})\}$ (triangle-free graphs)
- Halldórsson, Kuhn, N., and Tonoyan [HKNT22]
  - $\Theta(\log^2 n)$ colors from arbitrary palettes $\Psi(v)$ of size $\deg(v) + 1$.
- Dhawan [Dha24]
  - $\Theta(\Delta^\gamma + \sqrt{\log n})$ colors from $\{1, \ldots, \Theta(\frac{\Delta}{\log(\Delta^\gamma/\sqrt{k})}))\}$

    (k-locally sparse graphs for $k \ll \Delta^{2\gamma}$, i.e., $\max_v |G[N(v)]| \leq k \ll \Delta^{2\gamma}$)

Examples of palette sparsification results

- Assadi, Chen, and Khanna [ACK19]:
  - $\Theta(\log n)$ colors from $\{1, ..., \Delta + 1\}$.
- Alon and Assadi [AA20]:
  - $\Theta(\varepsilon^{-1.5}\sqrt{\log n})$ colors from $\{1, ..., (1+\varepsilon)\Delta\}$
  - $\Theta(\log n)$ colors from $\{1, ..., \deg(v) + 1\}$.
  - $\Theta(\varepsilon^{-1} \log n)$ colors from arbitrary palettes $\Psi(v)$ of size $(1+\varepsilon)\deg(v)$.
  - $\Theta(\Delta^\gamma + \sqrt{\log n})$ colors from $\{1, ..., \frac{9\Delta}{\gamma \ln \Delta})\}$ (triangle-free graphs)
- Halldórsson, Kuhn, N., and Tonoyan [HKNT22]
  - $\Theta(\log^2 n)$ colors from arbitrary palettes $\Psi(v)$ of size $\deg(v) + 1$.
- Dhawan [Dha24]
  - $\Theta(\Delta^\gamma + \sqrt{\log n})$ colors from $\{1, ..., \Theta(\frac{\Delta}{\log(\Delta^\gamma/\sqrt{k})}))\}$

    (k-locally sparse graphs for $k \ll \Delta^{2\gamma}$, i.e., $\max_v |G[N(v)]| \leq k \ll \Delta^{2\gamma}$)

Also theorems of similar flavor which I wasn't aware of in the graph theory literature, see [Dha24].

Why sparsify palettes?

Most salient use in memory-bound models (especially streaming, but also MPC).

Basic idea: we can forget about all edges s.t. $L(u) \cap L(v) = \emptyset$.

Why sparsify palettes?

Most salient use in memory-bound models (especially streaming, but also MPC).

Basic idea: we can forget about all edges s.t. $L(u) \cap L(v) = \emptyset$.

Definition (Sparsified graph)

The *sparsified graph* $\widetilde{G} = (V, E')$ of G is defined as $E' = \{uv \in E : L(u) \cap L(v) \neq \emptyset\}$.

Lemma

With every node sampling $\Theta(\log^d n)$ *colors out of a space of size* $\Theta(\Delta)$ *with* $d \geq 1/2$*, the maximum degree is* $O(\log^{2d} n)$ *w.h.p.*

Why sparsify palettes?

Most salient use in memory-bound models (especially streaming, but also MPC).

Basic idea: we can forget about all edges s.t. $L(u) \cap L(v) = \emptyset$.

Definition (Sparsified graph)

The *sparsified graph* $\widetilde{G} = (V, E')$ of G is defined as $E' = \{uv \in E : L(u) \cap L(v) \neq \emptyset\}$.

Lemma

With every node sampling $\Theta(\log^d n)$ *colors out of a space of size* $\Theta(\Delta)$ *with* $d \geq 1/2$*, the maximum degree is* $O(\log^{2d} n)$ *w.h.p.*

Potential to be relevant to many settings at once: [ACK19] and later papers implied results for streaming, query complexity, and MPC with quasi-linear memory.

Starter: palette sparsification with 2Δ colors

1. For $c \log n$ rounds
   1.1 Each uncolored node ($\varphi(v) = \perp$) picks a random color $\mathrm{try}(v) \in \{1, ..., 2\Delta\}$ u.a.r
   1.2 If the color $\mathrm{try}(v)$ is s.t. $\mathrm{try}(v) \neq \mathrm{try}(u)$ and $\mathrm{try}(v) \neq \varphi(u)$ for all neighbors $u \in N(v)$, set $\varphi(v) \leftarrow \mathrm{try}(v)$.

Starter: palette sparsification with 2Δ colors

1. For $c \log n$ rounds
   1.1 Each uncolored node ($\varphi(v) = \perp$) picks a random color $\mathrm{try}(v) \in \{1, ..., 2\Delta\}$ u.a.r
   1.2 If the color $\mathrm{try}(v)$ is s.t. $\mathrm{try}(v) \neq \mathrm{try}(u)$ and $\mathrm{try}(v) \neq \varphi(u)$ for all neighbors $u \in N(v)$, set $\varphi(v) \leftarrow \mathrm{try}(v)$.

In every round, each uncolored node has a probability $\geq 1/2$ to get colored. $\Rightarrow$ everyone gets colored w.h.p. $1 - n^{-c}$.

Starter: palette sparsification with 2Δ colors

1. For $c \log n$ rounds
   1.1 **Each uncolored node ($\varphi(v) = \bot$) picks a random color** $\mathrm{try}(v) \in \{1, \ldots, 2\Delta\}$ **u.a.r**
   1.2 If the color $\mathrm{try}(v)$ is s.t. $\mathrm{try}(v) \neq \mathrm{try}(u)$ and $\mathrm{try}(v) \neq \varphi(u)$ for all neighbors $u \in N(v)$, set $\varphi(v) \leftarrow \mathrm{try}(v)$.

In every round, each uncolored node has a probability $\geq 1/2$ to get colored. $\Rightarrow$ everyone gets colored w.h.p. $1 - n^{-c}$.

Key observation: Sampling is non-adaptive.

Starter: palette sparsification with 2Δ colors

1. For $c \log n$ rounds
   1.1 **Each uncolored node ($\varphi(v) = \bot$) picks a random color** $\mathrm{try}(v) \in \{1, ..., 2\Delta\}$ **u.a.r**
   1.2 If the color $\mathrm{try}(v)$ is s.t. $\mathrm{try}(v) \neq \mathrm{try}(u)$ and $\mathrm{try}(v) \neq \varphi(u)$ for all neighbors $u \in N(v)$, set $\varphi(v) \leftarrow \mathrm{try}(v)$.

In every round, each uncolored node has a probability $\geq 1/2$ to get colored. $\Rightarrow$ everyone gets colored w.h.p. $1 - n^{-c}$.

Key observation: Sampling is non-adaptive.

In terms of palette sparsification: W.h.p., each node sampling $\Theta(\log n)$ colors out of $\{1, ..., 2\Delta\}$ results in a list-coloring problem with a solution.

Starter bonus properties: coloring not just existential

Suppose following scenario:

1. Every node v samples its list $L(v) \subseteq \{1, \ldots, 2\Delta\}$ of size $\Theta(\log n)$.
2. We forget about edges uv s.t. $L(u) \cap L(v) = \emptyset$ (getting the sparsified graph $\widetilde{G}$),
3. We attempt to color the graph communicating only on $\widetilde{G}$.

Starter bonus properties: coloring not just existential

Suppose following scenario:

1. Every node v samples its list $L(v) \subseteq \{1, \ldots, 2\Delta\}$ of size $\Theta(\log n)$.
2. We forget about edges uv s.t. $L(u) \cap L(v) = \emptyset$ (getting the sparsified graph $\widetilde{G}$),
3. We attempt to color the graph communicating only on $\widetilde{G}$.

W.h.p., this is still easy here: we can basically still run the same algorithm.

Starter bonus properties: coloring not just existential

Suppose following scenario:

1. Every node v samples its list $L(v) \subseteq \{1, \ldots, 2\Delta\}$ of size $\Theta(\log n)$.
2. We forget about edges uv s.t. $L(u) \cap L(v) = \emptyset$ (getting the sparsified graph $\widetilde{G}$),
3. We attempt to color the graph communicating only on $\widetilde{G}$.

W.h.p., this is still easy here: we can basically still run the same algorithm.

But in general, can't expect existence to mean easily computable, especially across all models

Topic of this talk: Distributed Palette Sparsification Theorem

Theorem (Flin, Ghaffari, Halldórsson, Kuhn, and N. [FGH$^+$24])

Suppose that each node in a graph G samples $\Theta(\log^2 n)$ colors u.a.r. from $[\Delta + 1]$. There is a distributed message-passing algorithm operating on the sparsified graph, that computes a valid list-coloring in $O(\log^2 \Delta + \log^3 \log n)$ rounds with high probability, using $O(\log n)$-bit messages. In particular, each node needs to communicate with only $O(\log^4 n)$ different neighbors.

Topic of this talk: Distributed Palette Sparsification Theorem

Theorem (Flin, Ghaffari, Halldórsson, Kuhn, and N. [FGH+24])

Suppose that each node in a graph G samples $\Theta(\log^2 n)$ colors u.a.r. from $[\Delta + 1]$. There is a distributed message-passing algorithm operating on the **sparsified graph**, *that computes a valid list-coloring in $O(\log^2 \Delta + \log^3 \log n)$ rounds with high probability, using $O(\log n)$-bit messages. In particular, each node needs to communicate with only $O(\log^4 n)$ different neighbors.*

We are interested in the high Δ setting, $\geq \operatorname{poly}\log n$. (No need to sparsify if the number of colors is already small).

Topic of this talk: Distributed Palette Sparsification Theorem

Theorem (Flin, Ghaffari, Halldórsson, Kuhn, and N. [FGH+24])

Suppose that each node in a graph G samples $\Theta(\log^2 n)$ colors u.a.r. from $[\Delta + 1]$. There is a distributed message-passing algorithm operating on the **sparsified graph***, that computes a valid list-coloring in $O(\log^2 \Delta + \log^3 \log n)$ rounds with high probability, using $O(\log n)$-bit messages. In particular, each node needs to communicate with only $O(\log^4 n)$ different neighbors.*

We are interested in the high Δ setting, $\geq \text{poly} \log n$. (No need to sparsify if the number of colors is already small).

Goal: Designing a coloring algorithm s.t. w.h.p. at the end 1) everyone is colored; 2) each node sampled $O(\log^2 n)$ colors, and 3) the samples were non-adaptive.

Intuition for the $\log \Delta$ dependency

Suppose our graph $G = (V, E)$ is just one of more $\Delta + 1$-cliques.

- Diameter of the sparsified graph when sampling $\operatorname{poly}\log n$ colors?
- How many times can each color be used?

Intuition for the $\log \Delta$ dependency

Suppose our graph $G = (V, E)$ is just one of more $\Delta + 1$-cliques.

- Diameter of the sparsified graph when sampling $\operatorname{poly}\log n$ colors?
  $\rightarrow D(\widetilde{G}) \in \Omega(\log \Delta / \log\log n)$
- How many times can each color be used?

Intuition for the $\log \Delta$ dependency

Suppose our graph $G = (V, E)$ is just one of more $\Delta + 1$-cliques.

- Diameter of the sparsified graph when sampling $\operatorname{poly}\log n$ colors?
  $\rightarrow D(\widetilde{G}) \in \Omega(\log \Delta / \log \log n)$
- How many times can each color be used? Exactly once per clique.

Intuition for the $\log \Delta$ dependency

Suppose our graph $G = (V, E)$ is just one of more $\Delta + 1$-cliques.

- Diameter of the sparsified graph when sampling $\operatorname{poly}\log n$ colors?
  $\rightarrow D(\widetilde{G}) \in \Omega(\log \Delta / \log\log n)$
- How many times can each color be used? Exactly once per clique.

  Coloring such a graph corresponds to finding a perfect matching in the bipartite graph (V, C, E) with an edge between $vc \in V \times C$ iff color c is sampled by v.

Intuition for the $\log \Delta$ dependency

Suppose our graph $G = (V, E)$ is just one of more $\Delta + 1$-cliques.

- Diameter of the sparsified graph when sampling $\operatorname{poly}\log n$ colors?
  $\rightarrow D(\widetilde{G}) \in \Omega(\log \Delta / \log\log n)$
- How many times can each color be used? Exactly once per clique.

  Coloring such a graph corresponds to finding a perfect matching in the bipartite graph (V, C, E) with an edge between $vc \in V \times C$ iff color c is sampled by v.

Theorem (Flin, Ghaffari, Halldórsson, Kuhn, and N. [FGH+24])

Any LOCAL *algorithm that operates on the sparsified graph and computes a $(\Delta + 1)$-coloring with at least a constant probability of success needs $\Omega\left(\frac{\log \Delta}{\log\log n}\right)$ rounds. This holds even if the original graph is a $(\Delta + 1)$-clique, even if the distributed algorithm running on the sparsified graph uses unbounded messages, and even if each node samples a large* $\operatorname{poly}\log n$ *number of colors in the sparsification.*

Before we dive into the algorithm

Any questions at this point?

First ingredient: sparse-dense decomposition

Multiple versions of this kind of ideas have appeared now [Ree98, HSS18, ACK19, CLP20, AA20, AW22, AKM22, HKNT22, FHM23, FHN24].
For this result, one of the simpler versions.

First ingredient: sparse-dense decomposition

Multiple versions of this kind of ideas have appeared now [Ree98, HSS18, ACK19, CLP20, AA20, AW22, AKM22, HKNT22, FHM23, FHN24].
For this result, one of the simpler versions.

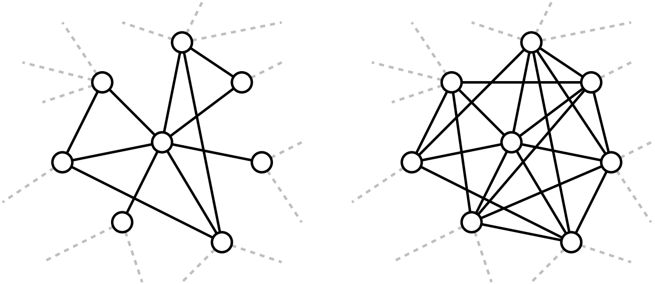
Definition (Sparsity)

The *sparsity* of a node v is the value $\zeta_v = \frac{1}{\Delta}\left(\binom{\Delta}{2} - |E(N(v))|\right)$. We say a node is ζ*-sparse* if $\zeta_v \geq \zeta$, otherwise it is ζ*-dense*.

First ingredient: sparse-dense decomposition

Definition (Sparsity)

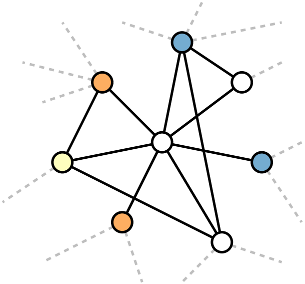
The *sparsity* of a node v is the value $\zeta_v = \frac{1}{\Delta}\left(\binom{\Delta}{2} - |E(N(v))|\right)$. We say a node is ζ*-sparse* if $\zeta_v \geq \zeta$, otherwise it is ζ*-dense*.

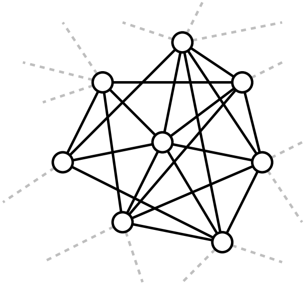


First ingredient: sparse-dense decomposition

Definition (Sparsity)

The *sparsity* of a node v is the value $\zeta_v = \frac{1}{\Delta}\left(\binom{\Delta}{2} - |E(N(v))|\right)$. We say a node is ζ*-sparse* if $\zeta_v \geq \zeta$, otherwise it is ζ*-dense*.

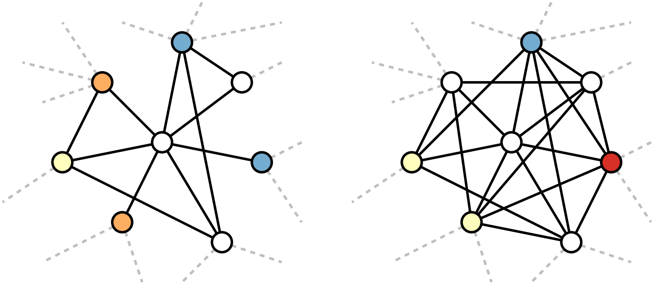




First ingredient: sparse-dense decomposition

Definition (Sparsity)

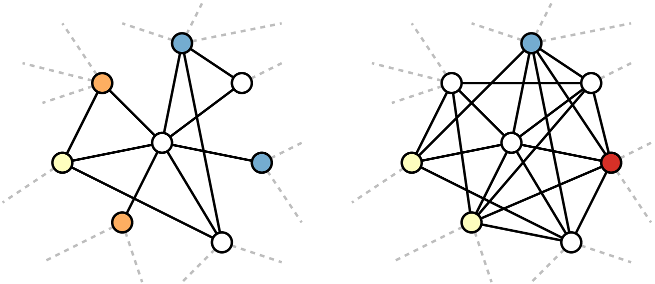
The *sparsity* of a node v is the value $\zeta_v = \frac{1}{\Delta}\left(\binom{\Delta}{2} - |E(N(v))|\right)$. We say a node is ζ*-sparse* if $\zeta_v \geq \zeta$, otherwise it is ζ*-dense*.



First ingredient: sparse-dense decomposition

Definition (Sparsity)

The *sparsity* of a node v is the value $\zeta_v = \frac{1}{\Delta}\left(\binom{\Delta}{2} - |E(N(v))|\right)$. We say a node is ζ-*sparse* if $\zeta_v \geq \zeta$, otherwise it is ζ-*dense*.

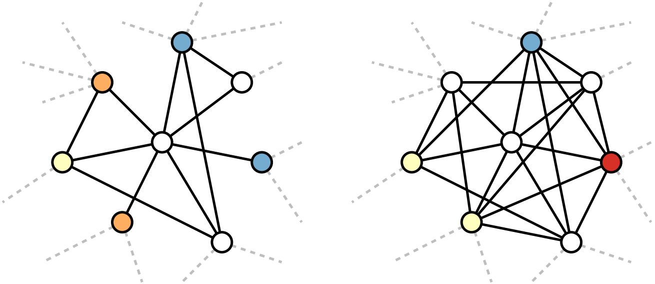


- Sparsity allows to *generate slack* as repeated colors around a node. [EPS15]

First ingredient: sparse-dense decomposition

Definition (Sparsity)

The *sparsity* of a node v is the value $\zeta_v = \frac{1}{\Delta}\left(\binom{\Delta}{2} - |E(N(v))|\right)$. We say a node is ζ*-sparse* if $\zeta_v \geq \zeta$, otherwise it is ζ*-dense*.



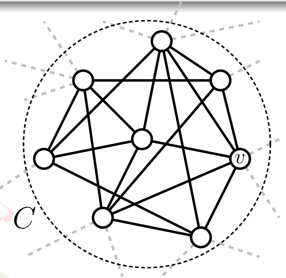
- Sparsity allows to *generate slack* as repeated colors around a node. [EPS15]
- A dense node cannot really be dense on its own, it is part of an *almost-clique*.

First ingredient: sparse-dense decomposition (continued)

Definition (Almost-Clique Decomposition)

For $\epsilon \in (0, 1/3)$, a ϵ-almost-clique decomposition is a partitioning of the vertices into sets $V_{\text{sparse}}, C_1, \ldots, C_k$ for some k such that:

1. All $v \in V_{\text{sparse}}$ are $\Omega(\epsilon^2 \Delta)$-sparse.
2. For any $i \in [k]$, almost-clique C_i has the following properties:
   2.1 $|C_i| \leq (1 + \epsilon)\Delta$;
   2.2 $|N(v) \cap C_i| \geq (1 - \epsilon)\Delta$ for all nodes $v \in C_i$.

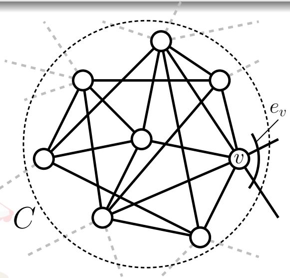


First ingredient: sparse-dense decomposition (continued)

Definition (Almost-Clique Decomposition)

For $\epsilon \in (0, 1/3)$, a ϵ-almost-clique decomposition is a partitioning of the vertices into sets $V_{\text{sparse}}, C_1, \ldots, C_k$ for some k such that:

1. All $v \in V_{\text{sparse}}$ are $\Omega(\epsilon^2\Delta)$-sparse.
2. For any $i \in [k]$, almost-clique C_i has the following properties:
   2.1 $|C_i| \leq (1+\epsilon)\Delta$;
   2.2 $|N(v) \cap C_i| \geq (1-\epsilon)\Delta$ for all nodes $v \in C_i$.



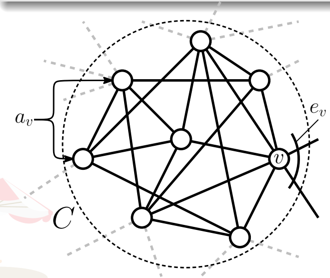
- External degree of $v \in C$: $e_v = |N(v) \setminus C|$

First ingredient: sparse-dense decomposition (continued)

Definition (Almost-Clique Decomposition)

For $\epsilon \in (0, 1/3)$, a ϵ-almost-clique decomposition is a partitioning of the vertices into sets $V_{\text{sparse}}, C_1, \dots, C_k$ for some k such that:

1. All $v \in V_{\text{sparse}}$ are $\Omega(\epsilon^2 \Delta)$-sparse.
2. For any $i \in [k]$, almost-clique C_i has the following properties:
   2.1 $|C_i| \leq (1+\epsilon)\Delta$;
   2.2 $|N(v) \cap C_i| \geq (1-\epsilon)\Delta$ for all nodes $v \in C_i$.



- External degree of $v \in C$: $e_v = |N(v) \setminus C|$
- Anti-degree of $v \in C$: $a_v = |C \setminus N(v)|$
- Average anti-degree in an almost-clique C: $\bar{a}_C$

But almost-clique decompositions in the sparsified graph?

Goal is to compute an almost-clique decomposition of the *original graph* G while having only access to the *sparsified graph* $\widetilde{G}$. Is that possible?

But almost-clique decompositions in the sparsified graph?

Goal is to compute an almost-clique decomposition of the *original graph* G while having only access to the *sparsified graph* $\widetilde{G}$. Is that possible?

No problem!

- If two nodes shared many neighbors pre-sparsification, they still do post-sparsification, and reciprocally.
- If a node shared many neighbors with most of its neighbors pre-sparsification, it is still true post-sparsification.

But almost-clique decompositions in the sparsified graph?

Goal is to compute an almost-clique decomposition of the *original graph* G while having only access to the *sparsified graph* $\widetilde{G}$. Is that possible?

No problem!

- If two nodes shared many neighbors pre-sparsification, they still do post-sparsification, and reciprocally.
- If a node shared many neighbors with most of its neighbors pre-sparsification, it is still true post-sparsification.

Biggest difference is that computing the almost-clique decomposition becomes a $O(\log \Delta)$-round algorithm instead of $O(1)$.

Main course: the algorithm

(All happening in a sparsified graph)

1. Compute the almost-clique decomposition.
2. Generate slack for sparse nodes and dense nodes with high external degree.
3. Color the dense nodes with high external degree $\geq \Delta / \log n$ and almost-cliques containing many such nodes.
   Only leaves uncolored the densest almost-cliques, of low external degree.
4. Compute a colorful matching in each almost-clique.
5. Reduce the number of uncolored nodes to $O(\Delta / \log n)$ in each almost-clique through $O(\log n \log \log n)$ (non-adaptive) random color trials.
6. Finish the coloring in each almost-clique using augmenting paths.

Preprocessing steps

1. Compute the almost-clique decomposition.
2. Generate slack for sparse nodes and dense nodes with high external degree.
3. Color the dense nodes with high external degree $\geq \Delta / \log n$ and almost-cliques containing many such nodes.
   Only leaves uncolored the densest almost-cliques, of low external degree.

Regarding slack: In just one round of each node trying a random color in $\{1, \ldots, \Delta + 1\}$ with constant probability, nodes of sparsity ζ_v get $\Omega(\zeta_v)$ repeated colors in their neighborhood w.p. $\geq 1 - \exp(-\Omega(\zeta_v))$. [EPS15]

Preprocessing steps

1. Compute the almost-clique decomposition.
2. Generate slack for sparse nodes and dense nodes with high external degree.
3. Color the dense nodes with high external degree $\geq \Delta / \log n$ and almost-cliques containing many such nodes.
   Only leaves uncolored the densest almost-cliques, of low external degree.

Regarding slack: In just one round of each node trying a random color in $\{1, \dots, \Delta + 1\}$ with constant probability, nodes of sparsity ζ_v get $\Omega(\zeta_v)$ repeated colors in their neighborhood w.p. $\geq 1 - \exp(-\Omega(\zeta_v))$. [EPS15]

Regarding high external degree nodes: with $\Delta / \log n$ slack, each color tried has a probability $\geq 1 / \log n$ to be free.

Trying $\log n$ colors implies constant probability of success. Doing it for $\log \log n$ rounds reduces degree to $\Delta / \log n$, reducing competition, allowing to finish in $\log \Delta$ extra attempts.

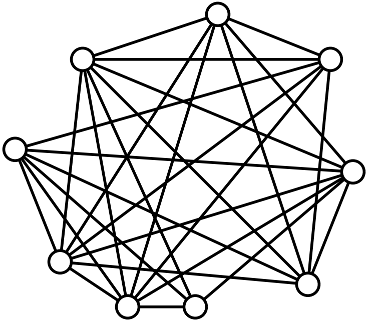
Back to the main course: the algorithm

(All happening in a sparsified graph)

1. Compute the almost-clique decomposition.
2. Generate slack for sparse nodes and dense nodes with high external degree.
3. Color the dense nodes with high external degree $\geq \Delta / \log n$ and almost-cliques containing many such nodes.
   Only leaves uncolored the densest almost-cliques, of low external degree.
4. **Compute a colorful matching in each almost-clique.**
5. Reduce the number of uncolored nodes to $O(\Delta / \log n)$ in each almost-clique through $O(\log n \log \log n)$ (non-adaptive) random color trials.
6. **Finish the coloring in each almost-clique using augmenting paths.**

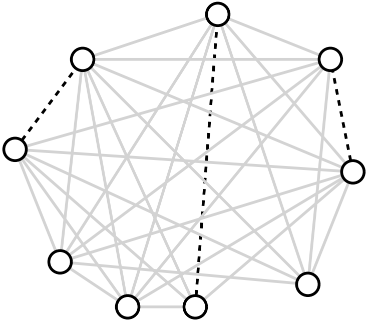
Colorful matching

Exploiting the missing edges in an almost-clique to the fullest. Obtained by sampling.



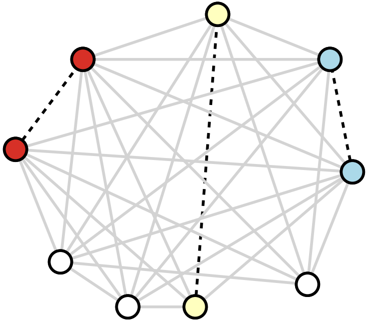
Colorful matching

Exploiting the missing edges in an almost-clique to the fullest. Obtained by sampling.



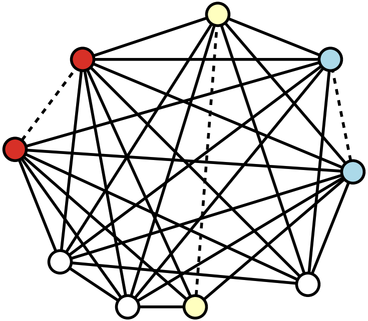
Colorful matching

Exploiting the missing edges in an almost-clique to the fullest. Obtained by sampling.



Colorful matching

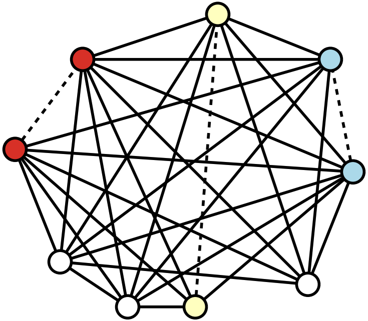
Exploiting the missing edges in an almost-clique to the fullest. Obtained by sampling.



Can find such a matching of size $\Omega(\bar{a}_C/\varepsilon)$.

Colorful matching

Exploiting the missing edges in an almost-clique to the fullest. Obtained by sampling.

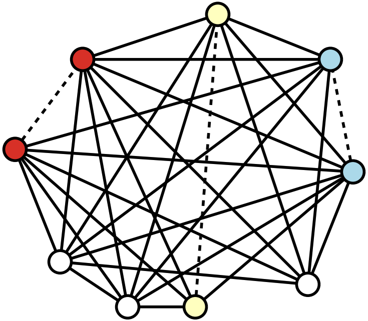


Can find such a matching of size $\Omega(\bar{a}_C/\varepsilon)$.

Guarantees that outside of this matching, we can afford to use each color only once in the almost-clique. An almost-clique of size $> \Delta + 1 + x$ gets a colorful matching of size $> x$.

Colorful matching

Exploiting the missing edges in an almost-clique to the fullest. Obtained by sampling.



Can find such a matching of size $\Omega(\bar{a}_C/\varepsilon)$.

Guarantees that outside of this matching, we can afford to use each color only once in the almost-clique. An almost-clique of size $> \Delta + 1 + x$ gets a colorful matching of size $> x$.

Important property for computing it: two nodes in an almost-clique whose lists of colors intersect are at distance 2 w.h.p. in the sparsified graph.

Limit of trying random colors

While a large part of the almost-clique is uncolored, having nodes sample a random color in $\{1, \ldots, \Delta + 1\}$ and keep it if free is efficient. But it soon runs into a corner.

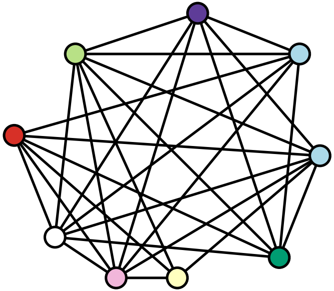
Limit of trying random colors

While a large part of the almost-clique is uncolored, having nodes sample a random color in $\{1, \ldots, \Delta + 1\}$ and keep it if free is efficient. But it soon runs into a corner.

Most extreme example: how does the last uncolored node in the almost-clique color itself? Each random color it samples is only free with probability $O(1/\Delta)$.

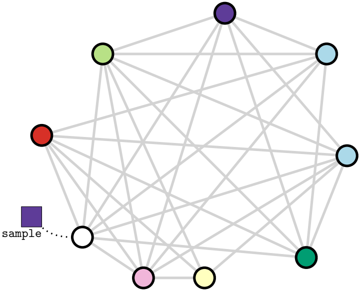
Augmenting paths

We can't have a node successfully color itself on its own. So we make it a group effort.



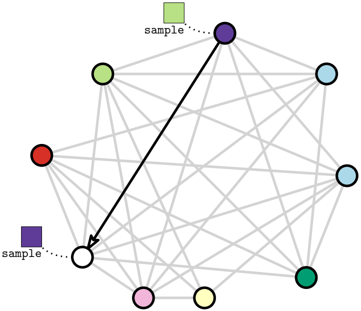
Augmenting paths

We can't have a node successfully color itself on its own. So we make it a group effort.



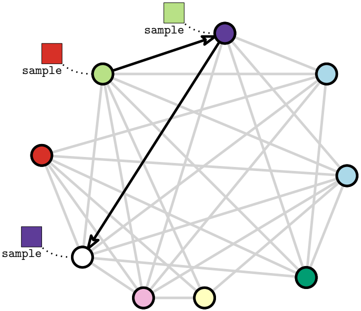
Augmenting paths

We can't have a node successfully color itself on its own. So we make it a group effort.



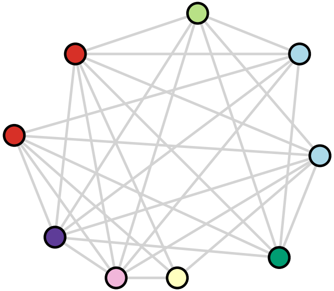
Augmenting paths

We can't have a node successfully color itself on its own. So we make it a group effort.



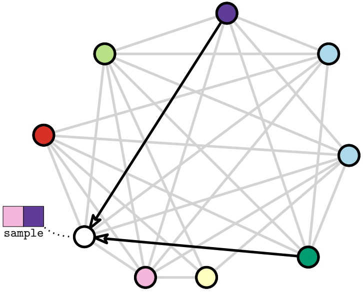
Augmenting paths

We can't have a node successfully color itself on its own. So we make it a group effort.



Augmenting paths

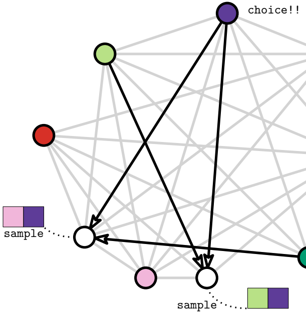
We can't have a node successfully color itself on its own. So we make it a group effort.



Works, but potentially long: make it faster with more samples!

Augmenting paths

We can't have a node successfully color itself on its own. So we make it a group effort.

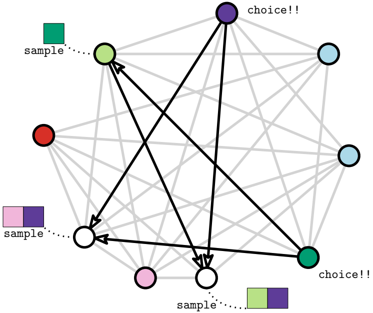


Works, but potentially long: make it faster with more samples!

And careful when doing it with multiple nodes at once.

Augmenting paths

We can't have a node successfully color itself on its own. So we make it a group effort.



Works, but potentially long: make it faster with more samples!

And careful when doing it with multiple nodes at once.

Harvesting trees

Let k be the number of uncolored nodes in a clique C.

Suppose some node v has a tree $T(v) \subseteq C$ s.t.:

- v is the root of the tree,
- For each node u in the tree, its parent could recolor itself with the color $\phi(u)$ of its child.
- For each node u in the tree, no neighbor of u outside the clique is attempting to recolor itself with one of the same color.
- The number of leaves of the tree is $\Theta(\Delta/k)$.

Harvesting trees

Let k be the number of uncolored nodes in a clique C.

Suppose some node v has a tree $T(v) \subseteq C$ s.t.:

- v is the root of the tree,
- For each node u in the tree, its parent could recolor itself with the color $\phi(u)$ of its child.
- For each node u in the tree, no neighbor of u outside the clique is attempting to recolor itself with one of the same color.
- The number of leaves of the tree is $\Theta(\Delta/k)$.

Most leaves have a probability of at least $\Omega(k/\Delta)$ to sample a free color when performing a random sample (k uncolored nodes essentially means k free colors).

Harvesting trees

Let k be the number of uncolored nodes in a clique C.

Suppose some node v has a tree $T(v) \subseteq C$ s.t.:

- v is the root of the tree,
- For each node u in the tree, its parent could recolor itself with the color $\phi(u)$ of its child.
- For each node u in the tree, no neighbor of u outside the clique is attempting to recolor itself with one of the same color.
- The number of leaves of the tree is $\Theta(\Delta/k)$.

Most leaves have a probability of at least $\Omega(k/\Delta)$ to sample a free color when performing a random sample (k uncolored nodes essentially means k free colors).

The $\Theta(\Delta/k)$ leaves give a constant success probability to the root to have an augmenting path in the tree.

How can this work?

We increase the degree of trees as k goes down, always s.t. the trees have a total of $\Theta(\Delta/k)$ leaves in total.

Not all leaves are useful to reach, but many are, and leaves are randomly distributed.

How can this work?

We increase the degree of trees as k goes down, always s.t. the trees have a total of $\Theta(\Delta/k)$ leaves in total.

Not all leaves are useful to reach, but many are, and leaves are randomly distributed.

k **large ($\geq \Omega(\log n)$):** Leaves try one color.

Get concentration from the large number of trees being grown.

How can this work?

We increase the degree of trees as k goes down, always s.t. the trees have a total of $\Theta(\Delta/k)$ leaves in total.

Not all leaves are useful to reach, but many are, and leaves are randomly distributed.

k **large** ($\geq \Omega(\log n)$)**:** Leaves try one color.

Get concentration from the large number of trees being grown.

k **small** ($\leq O(\log n)$)**:** Leaves try $\Theta(\log n)$ colors.

We gather in the almost-clique the knowledge of augmenting paths, and compute a matching.

Get concentration in each tree: each node should have $\Omega(k)$ augmenting paths to choose from.

All is well that ends well

Irrespective of the number of nodes that remain uncolored, we manage to color a constant fraction of them by growing trees and selecting a set of augmenting paths in them.

All is well that ends well

Irrespective of the number of nodes that remain uncolored, we manage to color a constant fraction of them by growing trees and selecting a set of augmenting paths in them.

$\Theta(\log \Delta)$ iterations, each consisting of growing augmenting paths of length $O(\log \Delta)$, colors everything.

Consequences

Algorithms for the following models:

- LocalStream model: nodes first receive their incident edges as a stream together with randomness from the other endpoint, can only store a few of them, then standard $\textsc{Congest}$ on the remembered graph.
- Coloring cluster graphs in $\operatorname{poly}\log n$ rounds (now superseded by [FHN24])
- Node capacitated clique: congested clique where nodes can only send and receive $\operatorname{poly}\log n$ messages in a round. We get the first $\operatorname{poly}\log n$ algorithm for this model.

Ending remarks

What have we learned?

1. Standard coloring can be a remarkably flexible problem, with this possibility of subsampling a more constrained smaller instance.

Ending remarks

What have we learned?

1. Standard coloring can be a remarkably flexible problem, with this possibility of subsampling a more constrained smaller instance.

What should be done next?

1. What other problems have results of this flavor? (palette sparsification / random partial commitment)
   Defective coloring ($\approx$ splitting)? Some LLLs?

Ending remarks

What have we learned?

1. Standard coloring can be a remarkably flexible problem, with this possibility of subsampling a more constrained smaller instance.

What should be done next?

1. What other problems have results of this flavor? (palette sparsification / random partial commitment)
   Defective coloring ($\approx$ splitting)? Some LLLs?
2. Can we leverage this for high-congestion problems?
   Typically, coloring power graphs, e.g. G^3.
   Issue: How to find the 3-hop surviving edges?

Ending remarks

What have we learned?

1. Standard coloring can be a remarkably flexible problem, with this possibility of subsampling a more constrained smaller instance.

What should be done next?

1. What other problems have results of this flavor? (palette sparsification / random partial commitment)
   Defective coloring ($\approx$ splitting)? Some LLLs?
2. Can we leverage this for high-congestion problems?
   Typically, coloring power graphs, e.g. G^3.
   Issue: How to find the 3-hop surviving edges?

Enjoy the rest of your holiday!

Noga Alon and Sepehr Assadi.
Palette sparsification beyond $(\Delta + 1)$ vertex coloring.
In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, volume 176 of *LIPIcs*, pages 6:1–6:22. LZI, 2020.

Sepehr Assadi, Yu Chen, and Sanjeev Khanna.
Sublinear algorithms for $(\Delta + 1)$ vertex coloring.
In *the Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 767–786, 2019.
Full version at arXiv:1807.08886.

Sepehr Assadi, Pankaj Kumar, and Parth Mittal.
Brooks' theorem in graph streams: a single-pass semi-streaming algorithm for Δ-coloring.
In *STOC*, pages 234–247. ACM, 2022.

Sepehr Assadi and Chen Wang.

Sublinear time and space algorithms for correlation clustering via sparse-dense decompositions.
In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 10:1–10:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

Yi-Jun Chang, Wenzheng Li, and Seth Pettie.
Distributed $(\Delta + 1)$-coloring via ultrafast graph shattering.
SIAM Journal of Computing, 49(3):497–539, 2020.

Abhishek Dhawan.
Palette sparsification for graphs with sparse neighborhoods.
CoRR, abs/2408.08256, 2024.

Michael Elkin, Seth Pettie, and Hsin-Hao Su.
$(2\Delta - 1)$-edge-coloring is much easier than maximal matching in the distributed setting.

In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 355–370, 2015.

Maxime Flin, Mohsen Ghaffari, Magnús M. Halldórsson, Fabian Kuhn, and Alexandre Nolin.
A distributed palette sparsification theorem.
In *the Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2024.

Manuela Fischer, Magnús M. Halldórsson, and Yannic Maus.
Fast distributed Brooks' theorem.
In *the Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2567–2588. SIAM, 2023.

Maxime Flin, Magnús M. Halldórsson, and Alexandre Nolin.
Decentralized Distributed Graph Coloring II: Degree+1-Coloring Virtual Graphs.

In *the Proceedings of the International Symposium on Distributed Computing (DISC)*, volume 319, pages 24:1–24:22, 2024.

Magnús M. Halldórsson, Fabian Kuhn, Alexandre Nolin, and Tigran Tonoyan.
Near-optimal distributed degree+1 coloring.
In *STOC*, pages 450–463. ACM, 2022.

David G. Harris, Johannes Schneider, and Hsin-Hao Su.
Distributed $(\Delta + 1)$-coloring in sublogarithmic rounds.
Journal of the ACM, 65:19:1–19:21, 2018.

Bruce A. Reed.
ω, Δ, and χ.
J. Graph Theory, 27(4):177–212, 1998.