

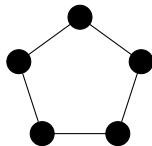
Distributed coloring in sparse graphs with fewer colors

Marthe Bonamy

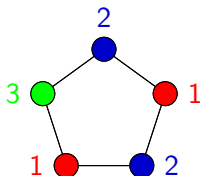
with Pierre Aboulker, Nicolas Bousquet, Louis Esperet



Coloring with fewer colors

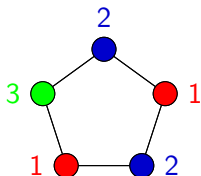


Coloring with fewer colors



$$\textcircled{c} \text{---} \textcircled{d} \Rightarrow c \neq d$$

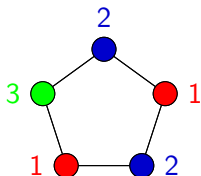
Coloring with fewer colors



χ : Minimum number of colors to guarantee:

$$\textcircled{c} \text{---} \textcircled{d} \Rightarrow c \neq d$$

Coloring with fewer colors



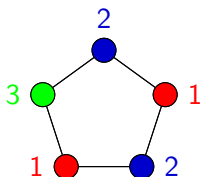
χ : Minimum number of colors to guarantee:

$$\textcircled{c} - \textcircled{d} \Rightarrow c \neq d$$

Δ : Maximum number of neighbors

$$\chi \leq \Delta + 1$$

Coloring with fewer colors



χ : Minimum number of colors to guarantee:

$$\textcircled{c} \text{---} \textcircled{d} \Rightarrow c \neq d$$

Δ : Maximum number of neighbors

$$\chi \leq \Delta + 1$$

Theorem (Brooks '61)

If G is neither a *clique* nor an *odd cycle*, then $\chi(G) \leq \Delta(G)$.

Changing the rules: The LOCAL Model

- Every vertex is its own agent (but has ∞ computational power).

Changing the rules: The LOCAL Model

- Every vertex is its own agent (but has ∞ computational power).
- Initially, vertices know nothing but their name (unique identifier).

Changing the rules: The LOCAL Model

- Every vertex is its own agent (but has ∞ computational power).
- Initially, vertices know nothing but their name (unique identifier).
- At each round, every vertex can exchange ∞ information with its neighbors.

Changing the rules: The LOCAL Model

- Every vertex is its own agent (but has ∞ computational power).
- Initially, vertices know nothing but their name (unique identifier).
- At each round, every vertex can exchange ∞ information with its neighbors.

Objective: **Minimize** the number of rounds before a solution can be computed.

Changing the rules: The LOCAL Model

- Every vertex is its own agent (but has ∞ computational power).
- Initially, vertices know nothing but their name (unique identifier).
- At each round, every vertex can exchange ∞ information with its neighbors.

Objective: **Minimize** the number of rounds before a solution can be computed.

Example of a long path (blackboard).

Changing the rules: The LOCAL Model

- Every vertex is its own agent (but has ∞ computational power).
- Initially, vertices know nothing but their name (unique identifier).
- At each round, every vertex can exchange ∞ information with its neighbors.

Objective: **Minimize** the number of rounds before a solution can be computed.

Example of a long path (blackboard).

Information theory

Changing the rules: The LOCAL Model

- Every vertex is its own agent (but has ∞ computational power).
- Initially, vertices know nothing but their name (unique identifier).
- At each round, every vertex can exchange ∞ information with its neighbors.

Objective: **Minimize** the number of rounds before a solution can be computed.

Example of a long path (blackboard).

Information theory

Randomized/Deterministic.

n -coloring: Easy.

Palette reduction

n -coloring: Easy.

n -coloring to $\Delta + 1$ -coloring?

n -coloring: Easy.

n -coloring to $\Delta + 1$ -coloring?

Theorem

We can compute a $(\Delta + 1)$ -coloring in:

- $2^{O(\sqrt{\log n})}$ rounds (Panconesi, Srinivasan '92)
- $O(\sqrt{\Delta} \text{polylog} \Delta) + \log^* n$ rounds (Fraigniaud, Heinrich, Kosowski '15)

Theorem (Panconesi, Srinivasan '95)

We can compute a Δ -coloring in $O\left(\frac{\Delta}{\log \Delta} \log^3 n\right)$ rounds.
(Assuming $\Delta \geq 3$ and the graph is not a clique.)

Theorem (Panconesi, Srinivasan '95)

We can compute a Δ -coloring in $O\left(\frac{\Delta}{\log \Delta} \log^3 n\right)$ rounds.
(Assuming $\Delta \geq 3$ and the graph is not a clique.)

What about list coloring?

Theorem (Panconesi, Srinivasan '95)

We can compute a Δ -coloring in $O\left(\frac{\Delta}{\log \Delta} \log^3 n\right)$ rounds.
(Assuming $\Delta \geq 3$ and the graph is not a clique.)

What about list coloring? G is **degree-choosable** iff it is colorable for any list assignment L s.t. $|L(v)| \geq d(v)$ for any vertex v .

Theorem (Panconesi, Srinivasan '95)

We can compute a Δ -coloring in $O\left(\frac{\Delta}{\log \Delta} \log^3 n\right)$ rounds.
(Assuming $\Delta \geq 3$ and the graph is not a clique.)

What about list coloring? G is **degree-choosable** iff it is colorable for any list assignment L s.t. $|L(v)| \geq d(v)$ for any vertex v .

Theorem (Borodin '77 / Erdős, Rubin, Taylor '79)

A graph is **degree-choosable** unless every 2-connected component is a clique or an odd cycle.

Theorem (Panconesi, Srinivasan '95)

We can compute a Δ -coloring in $O\left(\frac{\Delta}{\log \Delta} \log^3 n\right)$ rounds.
(Assuming $\Delta \geq 3$ and the graph is not a clique.)

What about list coloring? G is **degree-choosable** iff it is colorable for any list assignment L s.t. $|L(v)| \geq d(v)$ for any vertex v .

Theorem (Borodin '77 / Erdős, Rubin, Taylor '79)

A graph is **degree-choosable** unless every 2-connected component is a clique or an odd cycle.

What about sparse graphs?

4-colorable!

4-colorable!

Efficient 7-coloring. (Goldberg, Plotkin, Shannon '86)

4-colorable!

Efficient 7-coloring. (Goldberg, Plotkin, Shannon '86)

Theorem (Aboulker, B., Bousquet, Esperet '18)

We can compute a 6-list-coloring in $O(\log^3 n)$ rounds.

4-colorable!

Efficient 7-coloring. (Goldberg, Plotkin, Shannon '86)

Theorem (Aboulker, B., Bousquet, Esperet '18)

We can compute a 6-list-coloring in $O(\log^3 n)$ rounds.

Theorem (Aboulker, B., Bousquet, Esperet '18)

No distributed algorithm can 4-color every n -vertex planar graph in $o(n)$ rounds.

4-colorable!

Efficient 7-coloring. (Goldberg, Plotkin, Shannon '86)

Theorem (Aboulker, B., Bousquet, Esperet '18)

We can compute a 6-list-coloring in $O(\log^3 n)$ rounds.

Theorem (Aboulker, B., Bousquet, Esperet '18)

No distributed algorithm can 4-color every n -vertex planar graph in $o(n)$ rounds.

For triangle-free planar graphs: we can compute a 4-list-coloring in $O(\log^3 n)$ rounds, and no distributed algorithm can 3-color every n -vertex triangle-free planar graph in $o(n)$ rounds.

Goal: **shave off** a linear fraction of the vertices.

Goal: **shave off** a linear fraction of the vertices.

In the case of 7 colors?

Goal: **shave off** a linear fraction of the vertices.

In the case of 7 colors? **All** vertices of degree at most 6 can be shaven off.

Goal: **shave off** a linear fraction of the vertices.

In the case of 7 colors? **All** vertices of degree at most 6 can be shaven off.

In the case of 6 colors?

Goal: **shave off** a linear fraction of the vertices.

In the case of 7 colors? **All** vertices of degree at most 6 can be shaven off.

In the case of 6 colors? **Almost all!**

Actual generalization: sparse graphs

Arboricity $a(G)$ of a graph G : minimum number of edge-disjoint forests to cover the edges of G .

Actual generalization: sparse graphs

Arboricity $a(G)$ of a graph G : minimum number of edge-disjoint forests to cover the edges of G .

Theorem (Barembaum, Elkin '10)

We can compute a $(2a(G) + 1)$ -coloring in $O(a(G)^2 \log n)$ rounds.

Actual generalization: sparse graphs

Arboricity $a(G)$ of a graph G : minimum number of edge-disjoint forests to cover the edges of G .

Theorem (Barembaum, Elkin '10)

We can compute a $(2a(G) + 1)$ -coloring in $O(a(G)^2 \log n)$ rounds.

$$\text{mad}(G) \leq 2a(G)$$

Every graph is $\lceil \text{mad}(G) \rceil$ -colorable except in a few cases.

Actual generalization: sparse graphs

Arboricity $a(G)$ of a graph G : minimum number of edge-disjoint forests to cover the edges of G .

Theorem (Barembaum, Elkin '10)

We can compute a $(2a(G) + 1)$ -coloring in $O(a(G)^2 \log n)$ rounds.

$$\text{mad}(G) \leq 2a(G)$$

Every graph is $\lceil \text{mad}(G) \rceil$ -colorable except in a few cases.

Theorem (Aboulker, B., Bousquet, Esperet '18)

For $d \geq \{3, \text{mad}(G)\}$, we can compute a d -list-coloring in $O(d^4 \log^3 n)$ rounds. (Unless there is a clique on $d + 1$ vertices.)

Theorem (Borodin '77 / Erdős, Rubin, Taylor '79)

For any *nice* list assignment L , there is an L -coloring unless every 2-connected component is a clique or an odd cycle.

Nice list assignment: $\forall v, |L(v)| \geq d(v)$.

Actual generalization: local version

Theorem (Borodin '77 / Erdős, Rubin, Taylor '79)

For any *nice* list assignment L , there is an L -coloring unless every 2-connected component is a clique or an odd cycle.

Nice list assignment: $\forall v, |L(v)| \geq d(v)$.

Theorem (Aboulker, B., Bousquet, Esperet '18)

For any *very nice* list assignment L , we can compute an L -coloring in $O(\Delta^2 \log^3 n)$ rounds.

(Very nice list assignment: $\forall v, |L(v)| \geq d(v)$, with $|L(v)| \geq d(v) + 1$ if the neighborhood of v is a clique or $d(v) = 2$.)

Conclusion

5-coloring planar graphs?

Randomized setting?

5-coloring planar graphs?

Randomized setting?

Thanks!