

On the Complexity of Distributed Coloring

Cyril Gavoille¹ Ralf Klasing¹ Adrian Kosowski²
Alfredo Navarra³

¹Univeristy of Bordeaux, France

²Gdańsk University of Tecnology, Poland

³University of Perugia, Italy

DISC 2007 - Limassol, Cyprus

Coloring Problems

A **k -coloring** is a color assignment $V(G) \rightarrow \{1, \dots, k\}$ such that adjacent vertices of a graph G get different colors.

A **Maximal Independent Set** is set of pairwise non-adjacent vertices and which is maximal (for inclusion).

Both problems arise in many partitioning & clustering problems in many areas.

Sequential Algorithms

Every graph has a $(\Delta + 1)$ -coloring where Δ is the maximum degree, which can be found by a trivial sequential algorithm.

[Similar for maximal independent set]

Sequential Algorithms

Every graph has a $(\Delta + 1)$ -coloring where Δ is the maximum degree, which can be found by a trivial sequential algorithm.

[Similar for maximal independent set]

Procedure *Sequential_Coloring*(G):

- Select an uncolored vertex v
- Color v with a color distinct from its colored neighbors
- Repeat until all vertices are colored

Sequential Algorithms

Every graph has a $(\Delta + 1)$ -coloring where Δ is the maximum degree, which can be found by a trivial sequential algorithm.

[Similar for maximal independent set]

Procedure *Sequential_Coloring*(G):

- Select an uncolored vertex v
- Color v with a color distinct from its colored neighbors
- Repeat until all vertices are colored

Well-known variants to minimize the largest color:

Sequential Algorithms

Every graph has a $(\Delta + 1)$ -coloring where Δ is the maximum degree, which can be found by a trivial sequential algorithm.

[Similar for maximal independent set]

Procedure *Sequential_Coloring*(G):

- Select an uncolored vertex v
- Color v with a color distinct from its colored neighbors
- Repeat until all vertices are colored

Well-known variants to minimize the largest color:

- 1 **Greedy coloring**: choose the smallest available color

Sequential Algorithms

Every graph has a $(\Delta + 1)$ -coloring where Δ is the maximum degree, which can be found by a trivial sequential algorithm.

[Similar for maximal independent set]

Procedure *Sequential_Coloring*(G):

- Select an uncolored vertex v
- Color v with a color distinct from its colored neighbors
- Repeat until all vertices are colored

Well-known variants to minimize the largest color:

- 1 **Greedy coloring**: choose the smallest available color
- 2 **Largest-first coloring**: color first high degree vertices with the smallest possible available color

Distributed Setting

The challenge is to give fast deterministic **distributed** algorithms for these problems having sequential solutions: understand the *locality* of these problems.

Distributed Setting

The challenge is to give fast deterministic **distributed** algorithms for these problems having sequential solutions: understand the *locality* of these problems.

Secret goal: Prove that these problems are inherently *non-local*

What is the hope?

(in finding fast distributed algorithms)

All these problems have a local definition!

What is the hope?

(in finding fast distributed algorithms)

All these problems have a local definition!

$$(\Delta + 1)\text{-coloring: } c(v) \in [1, \Delta + 1] \setminus c(N_v)$$

Δ = maximum degree of the graph

N_v = set of neighbors of v

What is the hope?

(in finding fast distributed algorithms)

All these problems have a local definition!

$(\Delta + 1)$ -coloring: $c(v) \in [1, \Delta + 1] \setminus c(N_v)$

Maximal Independent Set: $i(v) = \text{YES}$ iff $i(N_v) = \{\text{NO}\}$

Δ = maximum degree of the graph

N_v = set of neighbors of v

What is the hope?

(in finding fast distributed algorithms)

All these problems have a local definition!

$(\Delta + 1)$ -coloring: $c(v) \in [1, \Delta + 1] \setminus c(N_v)$

Maximal Independent Set: $i(v) = \text{YES}$ iff $i(N_v) = \{\text{NO}\}$

Greedy coloring: $c(v) \in \min[1, \Delta + 1] \setminus c(N_v)$

$\Delta =$ maximum degree of the graph

$N_v =$ set of neighbors of v

What is the hope?

(in finding fast distributed algorithms)

All these problems have a local definition!

$(\Delta + 1)$ -coloring: $c(v) \in [1, \Delta + 1] \setminus c(N_v)$

Maximal Independent Set: $i(v) = \text{YES}$ iff $i(N_v) = \{\text{NO}\}$

Greedy coloring: $c(v) \in \min[1, \Delta + 1] \setminus c(N_v)$

Largest-first coloring: $c(v) \in \min[1, \Delta + 1] \setminus c(N_v^{\geq})$

Δ = maximum degree of the graph

N_v = set of neighbors of v

N_v^{\geq} = set of neighbors of v with degree at least $\deg(v)$

Model of Computation

Linial's free model, a.k.a. *LOCAL* model

- underlying topology reliable and unknown
- unique ID node, spontaneous wakeup, totally synchronous
- 1 round = exchange messages with one or more neighbors
- unlimited message size

Model of Computation

Linial's free model, a.k.a. *LOCAL* model

- underlying topology reliable and unknown
- unique ID node, spontaneous wakeup, totally synchronous
- 1 round = exchange messages with one or more neighbors
- unlimited message size

⇒ model to understand the LOCALITY of a problem.

In time t a vertex knows at most its distance- t neighborhood

Results

	lower	upper	ref.
$(\Delta + 1)$ -coloring	$\Omega(\log^* n)$	$2^{O(\sqrt{\log n})}$	[L92/PS96]

$$\text{polylog}(n) \ll 2^{\sqrt{\log n}} \ll n^\epsilon$$

$\log^* n = \text{the smallest } i \text{ such that } \log^{(i)} n \leq 2$

Results

	lower	upper	ref.
$(\Delta + 1)$ -coloring	$\Omega(\log^* n)$	$2^{O(\sqrt{\log n})}$	[L92/PS96]
Max. Indep. Set	$\Omega\left(\sqrt{\frac{\log n}{\log \log n}}\right)$	$2^{O(\sqrt{\log n})}$	[KMW04/PS93]

$\text{polylog}(n) \ll 2^{\sqrt{\log n}} \ll n^\epsilon$
 $\log^* n = \text{the smallest } i \text{ such that } \log^{(i)} n \leq 2$

Results

	lower	upper	ref.
$(\Delta + 1)$ -coloring	$\Omega(\log^* n)$	$2^{O(\sqrt{\log n})}$	[L92/PS96]
Max. Indep. Set	$\Omega\left(\sqrt{\frac{\log n}{\log \log n}}\right)$	$2^{O(\sqrt{\log n})}$	[KMW04/PS93]
greedy coloring	$\Omega\left(\frac{\log n}{\log \log n}\right)$	$2^{O(\sqrt{\log n})}$	[here / here]

$\text{polylog}(n) \ll 2^{\sqrt{\log n}} \ll n^\epsilon$
 $\log^* n = \text{the smallest } i \text{ such that } \log^{(i)} n \leq 2$

Results

	lower	upper	ref.
$(\Delta + 1)$ -coloring	$\Omega(\log^* n)$	$2^{O(\sqrt{\log n})}$	[L92/PS96]
Max. Indep. Set	$\Omega\left(\sqrt{\frac{\log n}{\log \log n}}\right)$	$2^{O(\sqrt{\log n})}$	[KMW04/PS93]
greedy coloring	$\Omega\left(\frac{\log n}{\log \log n}\right)$	$2^{O(\sqrt{\log n})}$	[here/here]
largest-first coloring	$\Omega(\sqrt{n})$	$O(\sqrt{n})$	[KK06/here]

$\text{polylog}(n) \ll 2^{\sqrt{\log n}} \ll n^\epsilon$
 $\log^* n = \text{the smallest } i \text{ such that } \log^{(i)} n \leq 2$

The $\Omega(\log n / \log \log n)$ Lower Bound

Let **A** be *any* distributed algorithm computing a greedy coloring for all n -vertex graphs.

The $\Omega(\log n / \log \log n)$ Lower Bound

Let **A** be *any* distributed algorithm computing a greedy coloring for all n -vertex graphs.

Step 1 The maximum color assigned by **A** to a node v of a depth- d tree T is $\leq d + 1$.

The $\Omega(\log n / \log \log n)$ Lower Bound

Let **A** be *any* distributed algorithm computing a greedy coloring for all n -vertex graphs.

Step 1 The maximum color assigned by **A** to a node v of a depth- d tree T is $\leq d + 1$.

Step 2 If $\text{girth}(G) \geq \ell$ and $\chi(G) > \ell/2$, then **A** requires at least $\ell/2 - 1$ rounds for G .

The $\Omega(\log n / \log \log n)$ Lower Bound

Let \mathbf{A} be *any* distributed algorithm computing a greedy coloring for all n -vertex graphs.

Step 1 The maximum color assigned by \mathbf{A} to a node v of a depth- d tree T is $\leq d + 1$.

Step 2 If $\text{girth}(G) \geq \ell$ and $\chi(G) > \ell/2$, then \mathbf{A} requires at least $\ell/2 - 1$ rounds for G .

Step 3 There are n -vertex graphs with $\text{girth} \geq \ell$ and $\chi > \ell/2$ for $\ell = \frac{1}{4} \log n / \log \log n$.



Thank You
for your attention