Distributed Algorithms for Coloring Interval Graphs

DISC 2014

Magnús M. Halldórsson and Christian Konrad

Reykjavik University

15.10.2014

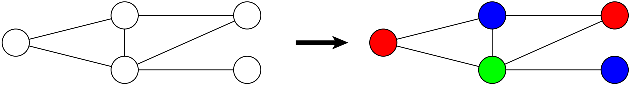
Distributed Vertex Coloring

Input: $G = (V, E)$, $n = |V|$, max. degree Δ

The $\mathcal{LOCAL}$ Model:

- Nodes host processors
- Synchronous communication along edges, individual messages of unbounded size
- Local computation is free
- Running time = maximal number of communication rounds

Vertex Coloring:



Chromatic number: $\chi(G)$

Output: When algorithm terminated, every node knows its color

Known Results

$(\Delta + 1)$**-coloring:**

- In general graphs: $O(\log n)$ rounds **[Luby, Siam J. on Comp., 1986]**
- **[Kuhn, Wattenhofer, PODC 2006], [Schneider, Wattenhofer, PODC 2008], [Kuhn, SPAA 2009], [Barenboim, Elkin, STOC 2009], [Schneider, Wattenhofer, PODC 2010], [Barenboim, Elkin, Pettie, Schneider, FOCS 2012], . . .**

Algorithm with approximation guarantee:

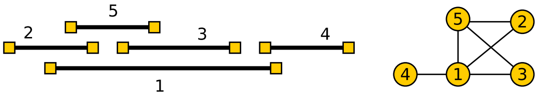
$\tilde{O}(\sqrt{n})$-approx. in $O(1)$ rounds **[Barenboim, ICALP 2012]**

Research Question:

Which graph classes admit distributed coloring algorithms with better approximation guarantees?

Our Main Result

Interval Graphs: Intersection graph of intervals of arbitrary lengths on the line



Theorem: (UB) There is a deterministic $\mathrm{O}(\log^* n)$ rounds distributed algorithm for coloring interval graphs with approximation factor $\mathrm{O}(1)$.

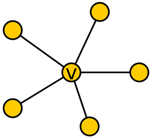
Theorem: (LB) Every distributed $\mathrm{O}(1)$-approximation algorithm for coloring interval graphs requires $\Omega(\log^* n)$ rounds.

Simple Class of Coloring Algorithms

Round-based coloring scheme:

In round i do:

1. Every not yet colored node $v \in V$ pre-selects itself with probability p_i
2. If no neighbor of v pre-selected itself:
   v colors itself with color i
3. Stop when all nodes are colored



Algorithms implementing this scheme have to determine the p_i

Properties:

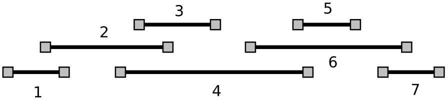
- Single bit messages, # rounds = # colors
- Scheduling in wireless networks **[Kesselheim, Vöcking, DISC 2010], [Halldórsson, Mitra, ICALP 2011], [Halldórsson, et al., SODA 2013]**
- Beep model **[Cornejo, Kuhn, DISC 2010]** (either beep or listen): One round of algorithm can be implemented in $O(\log n)$ rounds in Beep model

Simple Class of Coloring Algorithms: LB Result

UB 1: There is an algorithm following this scheme that colors any graph with $O(\Delta + \log^2 n)$ rounds (e.g. unit int. graphs: $\Delta = \Theta(\chi(G))$)
[Kesselheim,, Vöcking, DISC 2010]

UB 2: There is an algorithm following this scheme that colors interval graphs in $O(\chi(G) \log n)$ rounds **[Halldórsson et al., SODA 2013]**

1. *d-inductive-independent* graphs can be colored with the previous scheme in $O(d\chi(G) \log n)$ rounds
2. Interval graphs are 1-inductive independent (=perfect elimination ordering):



We prove:

Lower Bound: There is an interval graph so that any algorithm that follows the previous scheme requires $\Omega(\chi(G) \frac{\log n}{\log \log n})$ rounds

Outline

$O(1)$-approx. in $O(\log^* n)$ rounds in $\mathcal{LOCAL}$ model

Existing $O(\log^* n)$ Independent Set Algorithms

Reduction in $\mathcal{LOCAL}$-model:
Maximal Independent Set algorithm implies $(\Delta + 1)$-coloring

$O(\log^* n)$ **Rounds MIS algorithms:**

- Ring **[Cole, Vishkin, STOC 1986]**
- Extension to trees and constant degree graphs
- Bounded-independence Graphs **[Schneider, Wattenhofer, PODC 2008]**

Definition: A graph $G = (V, E)$ is of *bounded-independence* if there exists a bounding function $f(r)$ so that for each node $v \in V$, the size of a maximum independent set in the r-neighborhood of v is at most $f(r)$.

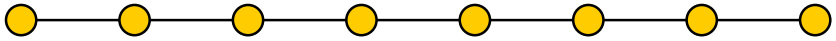
r-**Neighborhood:** $\Gamma^r(v) = \{u \in V : d(u, v) \leq r\}$.

Important: f is independent of n

Bounded-independence Graphs

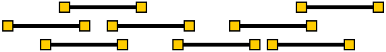
Examples:

- Path/Ring:



$$f(r) = r + 1$$

- Unit Interval Graphs:



$$f(r) = r + 1$$

- ~~Interval Graphs~~:

$(n-1)$-claw, not of bounded-independence

We will use: $\mathrm{O}(\log^* n)$ rounds algorithms to compute:

- $(\Delta + 1)$-coloring in constant-degree graph
- Maximal Independent Set in unit interval graph

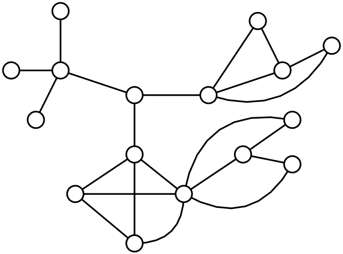
Algorithm in the $\mathcal{LOCAL}$ model

Goal: Find a dominating set $D \subseteq V$ so that:

$$\forall v \in D : |\Gamma^3(v) \cap D| = \mathrm{O}(1).$$

D implies $\mathrm{O}(1)$-approximation to coloring in $\mathrm{O}(\log^* n)$ rounds:

1. Find a distance-3 coloring of D, obtain color classes $D_1, \ldots, D_{\mathrm{O}(1)}$
2. Go through color classes D_i, each node $v \in D_i$ colors its neighbors



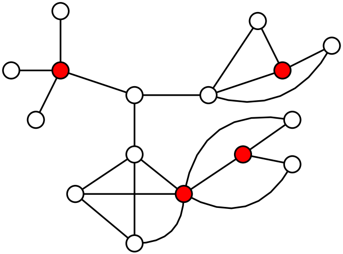
Algorithm in the $\mathcal{LOCAL}$ model

Goal: Find a dominating set $D \subseteq V$ so that:

$$\forall v \in D : |\Gamma^3(v) \cap D| = \mathrm{O}(1).$$

D implies $\mathrm{O}(1)$-approximation to coloring in $\mathrm{O}(\log^* n)$ rounds:

1. Find a distance-3 coloring of D, obtain color classes $D_1, \dots, D_{\mathrm{O}(1)}$
2. Go through color classes D_i, each node $v \in D_i$ colors its neighbors



Dominating set D

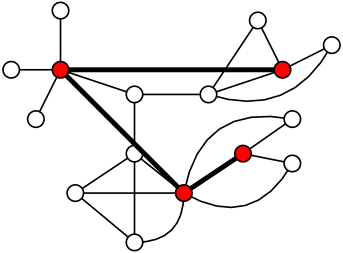
Algorithm in the $\mathcal{LOCAL}$ model

Goal: Find a dominating set $D \subseteq V$ so that:

$$\forall v \in D : |\Gamma^3(v) \cap D| = \mathrm{O}(1).$$

D implies $\mathrm{O}(1)$-approximation to coloring in $\mathrm{O}(\log^* n)$ rounds:

1. Find a distance-3 coloring of D, obtain color classes $D_1, \dots, D_{\mathrm{O}(1)}$
2. Go through color classes D_i, each node $v \in D_i$ colors its neighbors



Dominating set D, distance-3-adjacency of D

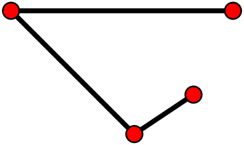
Algorithm in the $\mathcal{LOCAL}$ model

Goal: Find a dominating set $D \subseteq V$ so that:

$$\forall v \in D : |\Gamma^3(v) \cap D| = \mathrm{O}(1).$$

D implies $\mathrm{O}(1)$-approximation to coloring in $\mathrm{O}(\log^* n)$ rounds:

1. Find a distance-3 coloring of D, obtain color classes $D_1, \dots, D_{\mathrm{O}(1)}$
2. Go through color classes D_i, each node $v \in D_i$ colors its neighbors



Dominating set D, distance-3-adjacency of D, max. degree $= \mathrm{O}(1)$

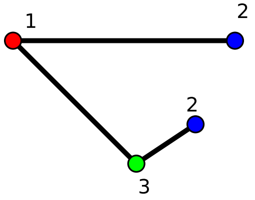
Algorithm in the $\mathcal{LOCAL}$ model

Goal: Find a dominating set $D \subseteq V$ so that:

$$\forall v \in D : |\Gamma^3(v) \cap D| = \mathrm{O}(1).$$

D implies $\mathrm{O}(1)$-approximation to coloring in $\mathrm{O}(\log^* n)$ rounds:

1. Find a distance-3 coloring of D, obtain color classes $D_1, \dots, D_{\mathrm{O}(1)}$
2. Go through color classes D_i, each node $v \in D_i$ colors its neighbors



Distance-3 coloring of D, e.g. using $(\Delta + 1)$-coloring algorithm for bounded-degree graphs (Cole & Vishkin) in $\mathrm{O}(\log^* n)$ rounds

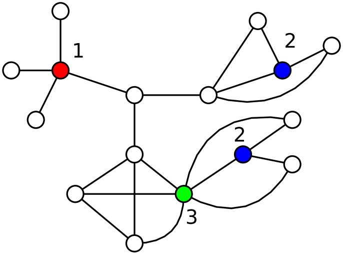
Algorithm in the $\mathcal{LOCAL}$ model

Goal: Find a dominating set $D \subseteq V$ so that:

$$\forall v \in D : |\Gamma^3(v) \cap D| = \mathrm{O}(1).$$

D implies $\mathrm{O}(1)$-approximation to coloring in $\mathrm{O}(\log^* n)$ rounds:

1. Find a distance-3 coloring of D, obtain color classes $D_1, \ldots, D_{\mathrm{O}(1)}$
2. Go through color classes D_i, each node $v \in D_i$ colors its neighbors



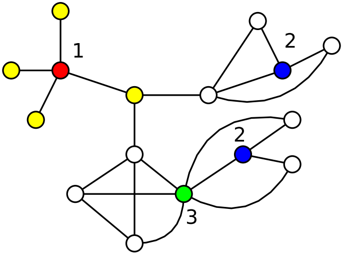
Algorithm in the $\mathcal{LOCAL}$ model

Goal: Find a dominating set $D \subseteq V$ so that:

$$\forall v \in D : |\Gamma^3(v) \cap D| = \mathrm{O}(1).$$

D implies $\mathrm{O}(1)$-approximation to coloring in $\mathrm{O}(\log^* n)$ rounds:

1. Find a distance-3 coloring of D, obtain color classes $D_1, \dots, D_{\mathrm{O}(1)}$
2. Go through color classes D_i, each node $v \in D_i$ colors its neighbors



Color class 1 colors its yet uncolored neighbors

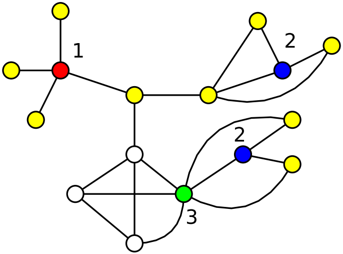
Algorithm in the $\mathcal{LOCAL}$ model

Goal: Find a dominating set $D \subseteq V$ so that:

$$\forall v \in D : |\Gamma^3(v) \cap D| = \mathrm{O}(1).$$

D implies $\mathrm{O}(1)$-approximation to coloring in $\mathrm{O}(\log^* n)$ rounds:

1. Find a distance-3 coloring of D, obtain color classes $D_1, \ldots, D_{\mathrm{O}(1)}$
2. Go through color classes D_i, each node $v \in D_i$ colors its neighbors



Color class 2 colors its yet uncolored neighbors
(no conflicts due to distance-3 coloring)

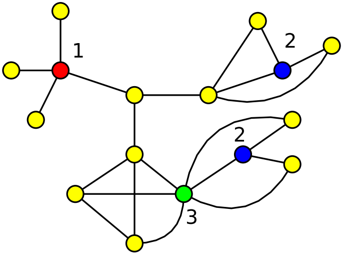
Algorithm in the $\mathcal{LOCAL}$ model

Goal: Find a dominating set $D \subseteq V$ so that:

$$\forall v \in D : |\Gamma^3(v) \cap D| = \mathrm{O}(1).$$

D implies $\mathrm{O}(1)$-approximation to coloring in $\mathrm{O}(\log^* n)$ rounds:

1. Find a distance-3 coloring of D, obtain color classes $D_1, \ldots, D_{\mathrm{O}(1)}$
2. Go through color classes D_i, each node $v \in D_i$ colors its neighbors



Color class 3 colors its yet uncolored neighbors

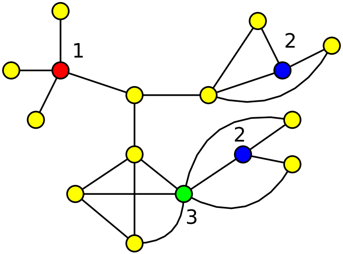
Algorithm in the $\mathcal{LOCAL}$ model

Goal: Find a dominating set $D \subseteq V$ so that:

$$\forall v \in D : |\Gamma^3(v) \cap D| = \mathrm{O}(1).$$

D implies $\mathrm{O}(1)$-approximation to coloring in $\mathrm{O}(\log^* n)$ rounds:

1. Find a distance-3 coloring of D, obtain color classes $D_1, \ldots, D_{\mathrm{O}(1)}$
2. Go through color classes D_i, each node $v \in D_i$ colors its neighbors



Summary: Given D, in $\mathrm{O}(\log^* n)$ rounds, we obtain an $\mathrm{O}(1)$-approximation to the coloring problem

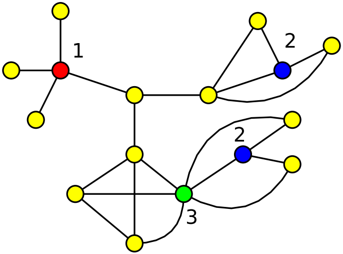
Algorithm in the $\mathcal{LOCAL}$ model

Goal: Find a dominating set $D \subseteq V$ so that:

$$\forall v \in D : |\Gamma^3(v) \cap D| = \mathrm{O}(1).$$

D implies $\mathrm{O}(1)$-approximation to coloring in $\mathrm{O}(\log^* n)$ rounds:

1. Find a distance-3 coloring of D, obtain color classes $D_1, \dots, D_{\mathrm{O}(1)}$
2. Go through color classes D_i, each node $v \in D_i$ colors its neighbors

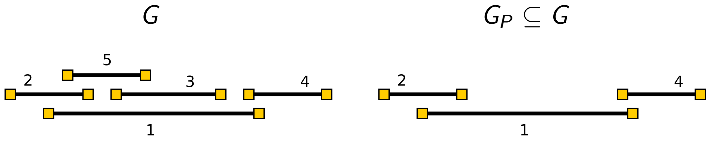


Remark: Greedy coloring in interval graphs = constant factor approximation

Finding D in interval graph

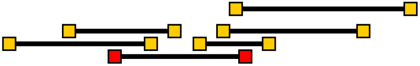
Strategy:

- Find subgraph $G_P \subseteq G$ of *proper* intervals by exploring 2-neighborhood:



G_P is also a unit interval graph

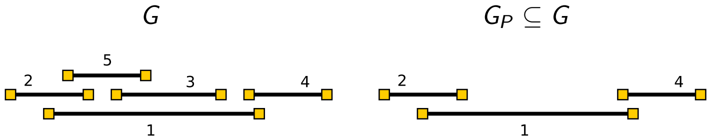
- Compute maximal independent set I in G_P in $\mathrm{O}(\log^* n)$ rounds
- Every node $v_i \in I$ selects the two nodes $v_i^1, v_i^2 \in \Gamma_{G_P}(v_i)$ that maximizes $|\Gamma(v_i) \cup \Gamma(v_i^1) \cup \Gamma(v_i^2)|$ (Intervals stretching out to the left and right as far as possible)
- $D = \bigcup_i \{v_i, v_i^1, v_i^2\}$



Finding D in interval graph

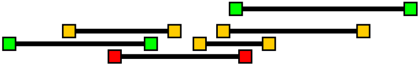
Strategy:

- Find subgraph $G_P \subseteq G$ of *proper* intervals by exploring 2-neighborhood:



  G_P is also a unit interval graph

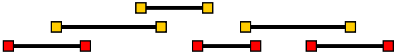
- Compute maximal independent set I in G_P in $\mathrm{O}(\log^* n)$ rounds
- Every node $v_i \in I$ selects the two nodes $v_i^1, v_i^2 \in \Gamma_{G_P}(v_i)$ that maximizes $|\Gamma(v_i) \cup \Gamma(v_i^1) \cup \Gamma(v_i^2)|$ (Intervals stretching out to the left and right as far as possible)
- $D = \bigcup_i \{v_i, v_i^1, v_i^2\}$



D has the desired properties

D is a dominating set in G:

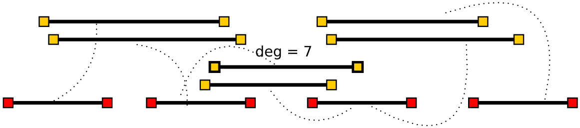
- Two adjacent nodes $u, v \in I$ are at distance at most 3



- Intervals reaching out furthest to the left/right bridge this gap

Property: $\forall v \in D : |\Gamma^3(v) \cap D| = \mathrm{O}(1)$.

- Maximal degree in $G|_D$ is 7



- At most 7^3 nodes in $\Gamma^3(v)$ for any $v \in D$.

Summary

Algorithm is optimal: Reduction to Linial's ring coloring lower bound
[Linial, SIAM Journal on Computing, 1992]

Main idea:

Use dominating set D that has a distance-3-coloring using $\mathrm{O}(1)$ colors

- Algorithm relies heavily on properties of interval graphs
- This allows the application of existing $\mathrm{O}(\log^* n)$ algorithms
- However, difficult to generalise

Open Questions:

- Can a similar result be obtained for disc graphs? (we can do $\mathrm{O}(1)$-approximation in $\mathrm{O}(\log n)$ rounds)

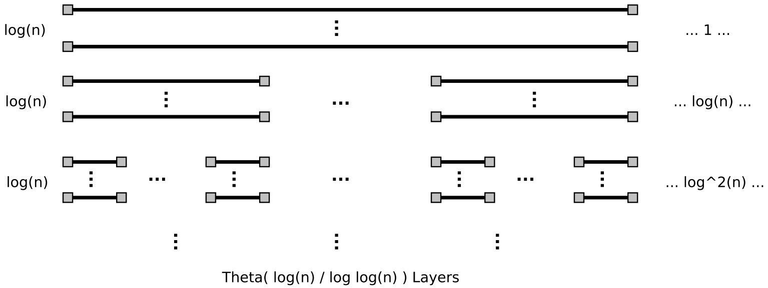
Lower Bound for Round-based Coloring Scheme

Round-based Coloring Scheme

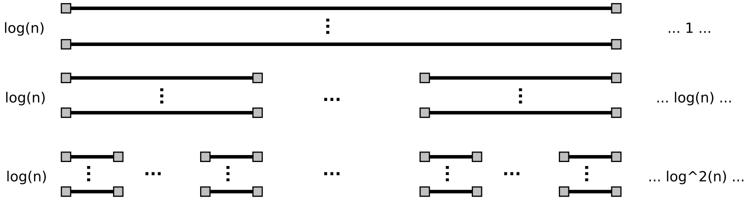
Round-based coloring scheme: (round i)

1. Every not yet colored node $v \in V$ pre-selects itself with probability p_i
2. If no neighbor of v pre-selected itself:
   v colors itself with color i
3. Stop when all nodes are colored

Hard Instance with chrom. number: $\chi(G) = \Theta(\log^2(n)/\log\log(n))$



Hard Instance



Choice of Probabilities:

- If $p_i \gg \chi(G)^{-1}$ (e.g. $p_i = \log(n)$): no progress as too many nodes pre-selected and they exclude each other
- If $p_i \ll \chi(G)^{-1}$ (e.g. $p_i = \log^3(n)$): not enough progress as too few nodes pre-selected

$\rightarrow$ Best progress if $p_i = \Theta(\chi(G)^{-1})$

We prove:

- $\Omega(\chi(G))$ iterations necessary to "eliminate" one layer
- Then, elimination of all layers: $\Omega(\chi(G)\frac{\log n}{\log\log n})$.

Summary

Summary: Round-based Coloring Scheme

- Lower Bound tight up to a $\log\log(n)$ factor
- Seems like $\log(n)$ factor has to be paid for non-trivial graph classes:
  - **UB 1:** There is an algorithm following this scheme that colors any graph with $\mathrm{O}(\Delta + \log^2 n)$ rounds
  - **UB 2:** There is an algorithm following this scheme that colors interval graphs in $\mathrm{O}(\chi(G) \log n)$ rounds

Thanks