Improved Distributed Algorithms for Coloring Interval Graphs with Application to Multicoloring Trees

Magnús M. Halldórsson and Christian Konrad

22.06.2017

Distributed Vertex Coloring

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

The $\mathcal{LOCAL}$ and $\mathcal{CONGEST}$ Models:

- Nodes host processors and have unique IDs
- Synchronous communication along edges, individual messages
  $\mathcal{LOCAL}$: messages of unbounded size
  $\mathcal{CONGEST}$: messages of size $\mathrm{O}(\log n)$
- Local computation is free
- Running time = number of communication rounds

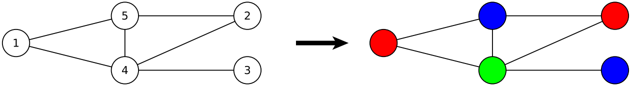
Distributed Vertex Coloring

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

The $\mathcal{LOCAL}$ and $\mathcal{CONGEST}$ Models:

- Nodes host processors and have unique IDs
- Synchronous communication along edges, individual messages
  $\mathcal{LOCAL}$: messages of unbounded size
  $\mathcal{CONGEST}$: messages of size $\mathrm{O}(\log n)$
- Local computation is free
- Running time = number of communication rounds

Minimum Vertex Coloring Problem:



Chromatic number: $\chi(G)$

Output: Upon termination of algorithm, every node knows its color

Minimum Vertex Coloring

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**

Minimum Vertex Coloring

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**
- Hard to Approximate within factor $n^{1-\epsilon}$ **[Håstad, 1999]**

Minimum Vertex Coloring

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**
- Hard to Approximate within factor $n^{1-\epsilon}$ **[Håstad, 1999]**

How to deal with such a hard problem?

Minimum Vertex Coloring

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**
- Hard to Approximate within factor $n^{1-\epsilon}$ **[Håstad, 1999]**

How to deal with such a hard problem?

- Degree-based quality bound: $\Delta + 1$-coloring
  Extensively studied in distributed algorithmics

Minimum Vertex Coloring

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**
- Hard to Approximate within factor $n^{1-\epsilon}$ **[Håstad, 1999]**

How to deal with such a hard problem?

- Degree-based quality bound: $\Delta + 1$-coloring
  Extensively studied in distributed algorithmics

Often very poor quality guarantee

Minimum Vertex Coloring

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**
- Hard to Approximate within factor $n^{1-\epsilon}$ **[Håstad, 1999]**

How to deal with such a hard problem?

- Degree-based quality bound: $\Delta + 1$-coloring
  Extensively studied in distributed algorithmics

Often very poor quality guarantee

- Specific graph classes: trees, interval graphs, unit disc graphs, etc.
  Much less studied in distributed algorithmics

Minimum Vertex Coloring

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**
- Hard to Approximate within factor $n^{1-\epsilon}$ **[Håstad, 1999]**

How to deal with such a hard problem?

- Degree-based quality bound: $\Delta + 1$-coloring
  Extensively studied in distributed algorithmics

Often very poor quality guarantee

- Specific graph classes: trees, interval graphs, unit disc graphs, etc.
  Much less studied in distributed algorithmics

- Exponential time algorithms
  n^{ϵ}-approximation in $\exp \mathrm{O}(\frac{1}{\epsilon})$ rounds **[Barenboim, Elkin, Gavoille, 2015]**

Distributed Minimum Vertex Coloring

Relevant Previous Works:

- LB: $\Omega(\log n)$ rounds for $\mathrm{O}(1)$-coloring on trees **[Linial, 1992]**

Distributed Minimum Vertex Coloring

Relevant Previous Works:

- LB: $\Omega(\log n)$ rounds for $\mathrm{O}(1)$-coloring on trees **[Linial, 1992]**
- $\mathrm{O}(a)$-approximation in $\mathrm{O}(a \log n)$ rounds on graphs with arboricity a (3-coloring on trees in $\mathrm{O}(\log n)$ rounds) **[Barenboim, Elkin, 2010]**

Distributed Minimum Vertex Coloring

Relevant Previous Works:

- LB: $\Omega(\log n)$ rounds for $\mathrm{O}(1)$-coloring on trees **[Linial, 1992]**
- $\mathrm{O}(a)$-approximation in $\mathrm{O}(a \log n)$ rounds on graphs with arboricity a (3-coloring on trees in $\mathrm{O}(\log n)$ rounds) **[Barenboim, Elkin, 2010]**
- $\mathrm{O}(1)$-approximation in $\mathrm{O}(\log^* n)$ rounds on interval graphs **[Halldórsson, Konrad, 2014]**

Distributed Minimum Vertex Coloring

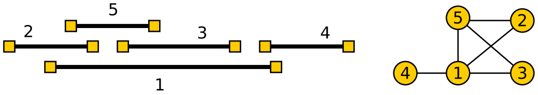
Relevant Previous Works:

- LB: $\Omega(\log n)$ rounds for $\mathrm{O}(1)$-coloring on trees **[Linial, 1992]**
- $\mathrm{O}(a)$-approximation in $\mathrm{O}(a \log n)$ rounds on graphs with arboricity a (3-coloring on trees in $\mathrm{O}(\log n)$ rounds) **[Barenboim, Elkin, 2010]**
- $\mathrm{O}(1)$-approximation in $\mathrm{O}(\log^* n)$ rounds on interval graphs **[Halldórsson, Konrad, 2014]**

This work: Improvements on **[Halldórsson, Konrad, 2014]**

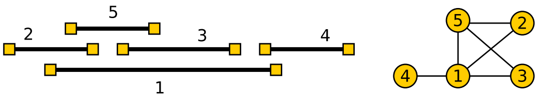
Distributed Coloring of Interval Graphs

Interval Graphs: Intersection graph of intervals on the line



Distributed Coloring of Interval Graphs

Interval Graphs: Intersection graph of intervals on the line

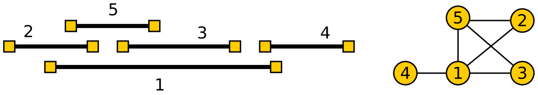


[Halldórsson, Konrad, 2014] :

- Constant factor approximation in $\mathrm{O}(\log^* n)$ rounds ($\mathcal{LOCAL}$)
- Interval boundaries known: adaptation to $\mathcal{CONGEST}$
- Every $\mathrm{O}(1)$-approximation requires $\Omega(\log^* n)$ rounds (via Linial)

Distributed Coloring of Interval Graphs

Interval Graphs: Intersection graph of intervals on the line



[Halldórsson, Konrad, 2014] :

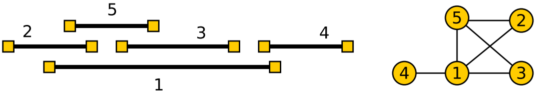
- Constant factor approximation in $\mathrm{O}(\log^* n)$ rounds ($\mathcal{LOCAL}$)
- Interval boundaries known: adaptation to $\mathcal{CONGEST}$
- Every $\mathrm{O}(1)$-approximation requires $\Omega(\log^* n)$ rounds (via Linial)

Our Results:

- $(1+\epsilon)$-approximation in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds
- Interval boundaries known: adaptation to $\mathcal{CONGEST}$
- LB: $\Omega(\frac{1}{\epsilon})$ rounds necessary
- $(1+\epsilon)$-approx. for multicoloring directed trees in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds

Distributed Coloring of Interval Graphs

Interval Graphs: Intersection graph of intervals on the line



[Halldórsson, Konrad, 2014] :

- Constant factor approximation in $\mathrm{O}(\log^* n)$ rounds ($\mathcal{LOCAL}$)
- Interval boundaries known: adaptation to $\mathcal{CONGEST}$
- Every $\mathrm{O}(1)$-approximation requires $\Omega(\log^* n)$ rounds (via Linial)

Our Results:

- $(1+\epsilon)$-approximation in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds
- Interval boundaries known: adaptation to $\mathcal{CONGEST}$
- LB: $\Omega(\frac{1}{\epsilon})$ rounds necessary
- $(1+\epsilon)$-approx. for multicoloring directed trees in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds

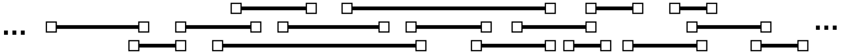
Outline

1. $\mathcal{LOCAL}$ model algorithm
2. Adaptation to $\mathcal{CONGEST}$

Outline

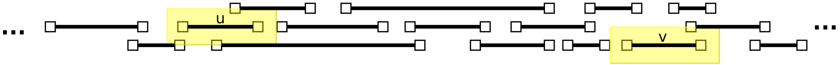
1. $\mathcal{LOCAL}$ model algorithm
2. Adaptation to $\mathcal{CONGEST}$

Algorithm in the $\mathcal{LOCAL}$ model



Algorithm:

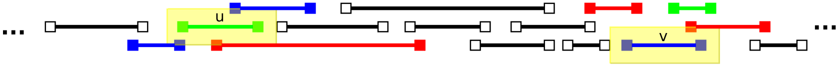
Algorithm in the $\mathcal{LOCAL}$ model



Algorithm:

1. $I \leftarrow$ distance-k maximal independent set

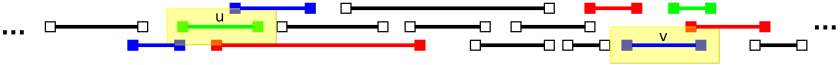
Algorithm in the $\mathcal{LOCAL}$ model



Algorithm:

1. $I \leftarrow$ distance-k maximal independent set
2. Nodes of I color inclusive neighborhoods optimally

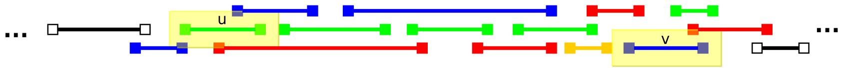
Algorithm in the $\mathcal{LOCAL}$ model



Algorithm:

1. $I \leftarrow$ distance-k maximal independent set
2. Nodes of I color inclusive neighborhoods optimally
3. Uncolored nodes form connected components of diameter at most $2k$

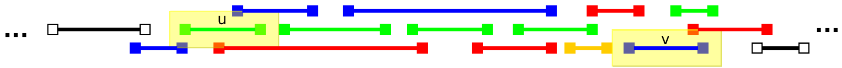
Algorithm in the $\mathcal{LOCAL}$ model



Algorithm:

1. $I \leftarrow$ distance-k maximal independent set
2. Nodes of I color inclusive neighborhoods optimally
3. Uncolored nodes form connected components of diameter at most $2k$
4. The node with smaller ID among u, v completes coloring of uncolored nodes between u, v optimally

Algorithm in the $\mathcal{LOCAL}$ model

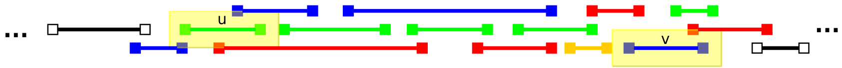


Algorithm:

1. $I \leftarrow$ distance-k maximal independent set
2. Nodes of I color inclusive neighborhoods optimally
3. Uncolored nodes form connected components of diameter at most $2k$
4. The node with smaller ID among u, v completes coloring of uncolored nodes between u, v optimally

Runtime?

Algorithm in the $\mathcal{LOCAL}$ model

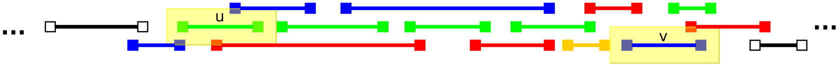


Algorithm:

1. $I \leftarrow$ distance-k maximal independent set $O(?)$
2. Nodes of I color inclusive neighborhoods optimally
3. Uncolored nodes form connected components of diameter at most $2k$
4. The node with smaller ID among u, v completes coloring of uncolored nodes between u, v optimally

Runtime?

Algorithm in the $\mathcal{LOCAL}$ model

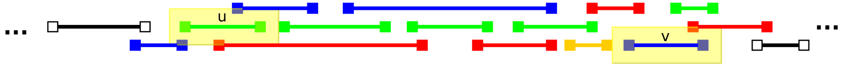


Algorithm:

1. $I \leftarrow$ distance-k maximal independent set $O(?)$
2. Nodes of I color inclusive neighborhoods optimally $O(1)$
3. Uncolored nodes form connected components of diameter at most $2k$
4. The node with smaller ID among u, v completes coloring of uncolored nodes between u, v optimally

Runtime?

Algorithm in the $\mathcal{LOCAL}$ model

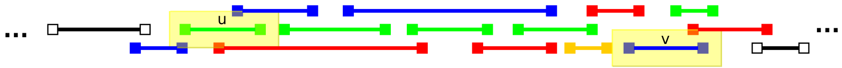


Algorithm:

1. $I \leftarrow$ distance-k maximal independent set $O(?)$
2. Nodes of I color inclusive neighborhoods optimally $O(1)$
3. Uncolored nodes form connected components of diameter at most $2k$
4. The node with smaller ID among u, v completes coloring of uncolored nodes between u, v optimally $O(k)$

Runtime?

Algorithm in the $\mathcal{LOCAL}$ model

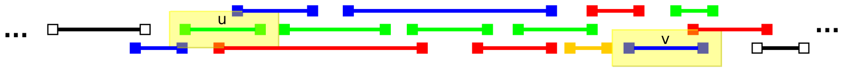


Algorithm:

1. $I \leftarrow$ distance-k maximal independent set $O(?)$
2. Nodes of I color inclusive neighborhoods optimally $O(1)$
3. Uncolored nodes form connected components of diameter at most $2k$
4. The node with smaller ID among u, v completes coloring of uncolored nodes between u, v optimally $O(k)$

Runtime? $O(? + k)$

Algorithm in the $\mathcal{LOCAL}$ model



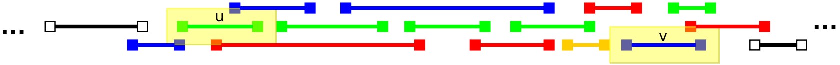
Algorithm:

1. $I \leftarrow$ distance-k maximal independent set $O(?)$
2. Nodes of I color inclusive neighborhoods optimally $O(1)$
3. Uncolored nodes form connected components of diameter at most $2k$
4. The node with smaller ID among u, v completes coloring of uncolored nodes between u, v optimally $O(k)$

Runtime? $O(? + k)$

Approximation Factor?

Algorithm in the $\mathcal{LOCAL}$ model



Algorithm:

1. $I \leftarrow$ distance-k maximal independent set $O(?)$
2. Nodes of I color inclusive neighborhoods optimally $O(1)$
3. Uncolored nodes form connected components of diameter at most $2k$
4. The node with smaller ID among u, v completes coloring of uncolored nodes between u, v optimally $O(k)$

Runtime? $O(? + k)$

Approximation Factor? $\#$ colors used in coloring completion step

Distance-k MIS

MIS algorithm $\rightarrow$ distance-k MIS algorithm:

- Simulate MIS on G^k (nodes adjacent if distance at most k)
- MIS in r rounds gives distance-k MIS in $\mathrm{O}(kr)$ rounds

Distance-k MIS

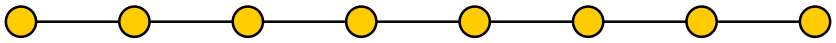
MIS algorithm $\rightarrow$ distance-k MIS algorithm:

- Simulate MIS on G^k (nodes adjacent if distance at most k)
- MIS in r rounds gives distance-k MIS in $\mathrm{O}(kr)$ rounds

MIS on bounded-independence graphs: **[Schneider, Wattenhofer, 2008]**
$\mathrm{O}(\log^* n)$ rounds in $\mathcal{CONGEST}$ model on BI graphs

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

Path/Ring:



$f(r) = r + 1$

Distance-k MIS

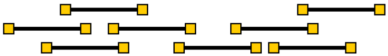
MIS algorithm $\rightarrow$ distance-k MIS algorithm:

- Simulate MIS on G^k (nodes adjacent if distance at most k)
- MIS in r rounds gives distance-k MIS in $\mathrm{O}(kr)$ rounds

MIS on bounded-independence graphs: **[Schneider, Wattenhofer, 2008]**
$\mathrm{O}(\log^* n)$ rounds in $\mathcal{CONGEST}$ model on BI graphs

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

Unit Interval Graphs:



$f(r) = r + 1$

Distance-k MIS

MIS algorithm $\rightarrow$ distance-k MIS algorithm:

- Simulate MIS on G^k (nodes adjacent if distance at most k)
- MIS in r rounds gives distance-k MIS in $O(kr)$ rounds

MIS on bounded-independence graphs: **[Schneider, Wattenhofer, 2008]**
$O(\log^* n)$ rounds in $\mathcal{CONGEST}$ model on BI graphs

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

~~Interval Graphs~~:

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

Distance-k MIS

MIS algorithm $\rightarrow$ distance-k MIS algorithm:

- Simulate MIS on G^k (nodes adjacent if distance at most k)
- MIS in r rounds gives distance-k MIS in $\mathrm{O}(kr)$ rounds

MIS on bounded-independence graphs: **[Schneider, Wattenhofer, 2008]**
$\mathrm{O}(\log^* n)$ rounds in $\mathcal{CONGEST}$ model on BI graphs

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

~~Interval Graphs~~:

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

Distance-k MIS

MIS algorithm $\rightarrow$ distance-k MIS algorithm:

- Simulate MIS on G^k (nodes adjacent if distance at most k)
- MIS in r rounds gives distance-k MIS in $\mathrm{O}(kr)$ rounds

MIS on bounded-independence graphs: **[Schneider, Wattenhofer, 2008]**
$\mathrm{O}(\log^* n)$ rounds in $\mathcal{CONGEST}$ model on BI graphs

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

~~Interval Graphs~~:

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

- Extract subgraph of *proper intervals* (= unit interval graph)
- Distance-k MIS in $\mathrm{O}(k \log^* n)$ rounds

Distance-k MIS

MIS algorithm $\rightarrow$ distance-k MIS algorithm:

- Simulate MIS on G^k (nodes adjacent if distance at most k)
- MIS in r rounds gives distance-k MIS in $\mathrm{O}(kr)$ rounds

MIS on bounded-independence graphs: **[Schneider, Wattenhofer, 2008]**
$\mathrm{O}(\log^* n)$ rounds in $\mathcal{CONGEST}$ model on BI graphs

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

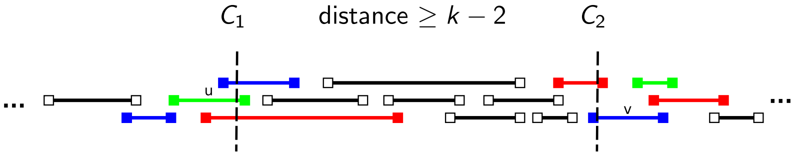
~~Interval Graphs~~:

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

- Extract subgraph of *proper intervals* (= unit interval graph)
- Distance-k MIS in $\mathrm{O}(k \log^* n)$ rounds $O(k \log^* n + k)$

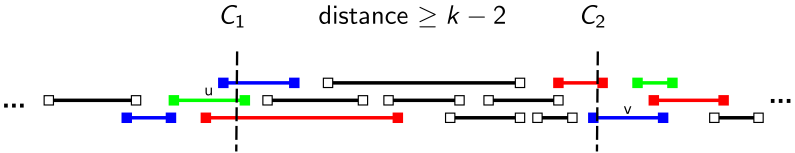
Approximation Factor

Goal: Prove that color completion with few colors exists

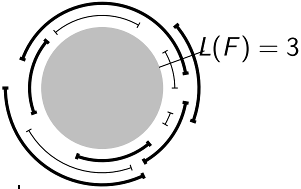


Approximation Factor

Goal: Prove that color completion with few colors exists



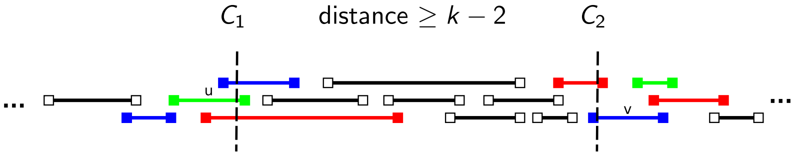
Circular Arc Graphs:



- Load $L(G)$: Largest subset containing the same point
- Circular cover length $l(G)$: cardinality of smallest subset of arcs covering the circle

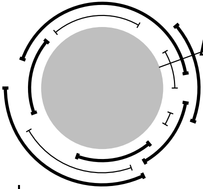
Approximation Factor

Goal: Prove that color completion with few colors exists



Circular Arc Graphs:

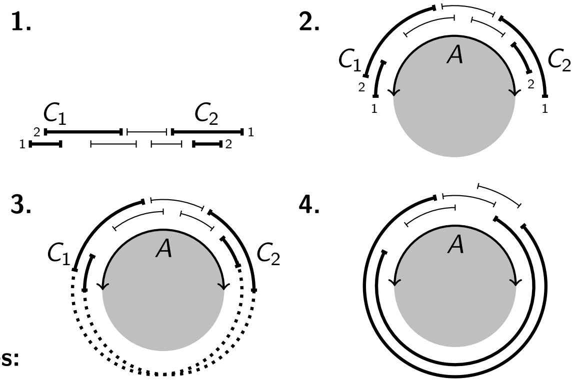
- Load $L(G)$: Largest subset containing the same point
- Circular cover length $l(G)$: cardinality of smallest subset of arcs covering the circle



[Valencia-Pabon, 2003] :

$\lfloor \left(1 + \frac{1}{l(G)-2}\right) L(G) \rfloor + 1$ colors suffice to color circular arc graph G

Pre-colored Interval Graph G to Circular Arc Graph F:



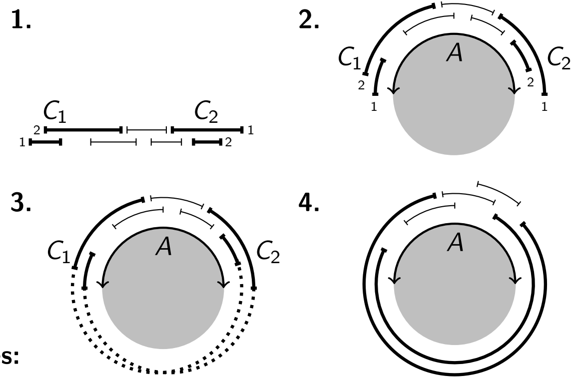
Properties:

1. Load: $L(F) \leq \chi(G)$
2. Circular cover: $l(F) \geq k - 2$

Valencia-Pabon: $\left\lfloor \left(1 + \frac{1}{k-4}\right) \chi(G) \right\rfloor + 1$ colors suffice

Approximation Factor (2)

Pre-colored Interval Graph G to Circular Arc Graph F:



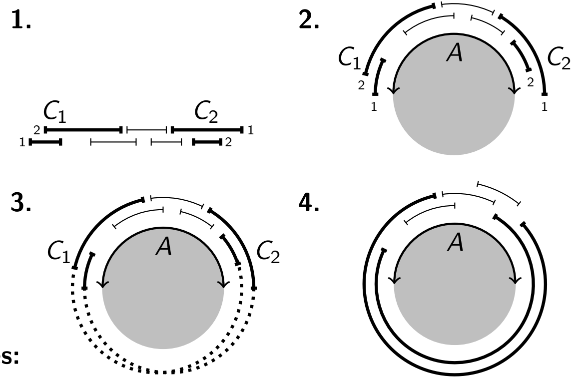
Properties:

1. Load: $L(F) \leq \chi(G)$
2. Circular cover: $l(F) \geq k - 2$

$k \sim \frac{1}{\epsilon}$, $\epsilon \geq \frac{2}{\chi(G)}$

Valencia-Pabon: $\lfloor \left(1 + \frac{1}{k-4}\right) \chi(G) \rfloor + 1$ colors suffice

Pre-colored Interval Graph G to Circular Arc Graph F:



Properties:

1. Load: $L(F) \leq \chi(G)$
2. Circular cover: $l(F) \geq k - 2$

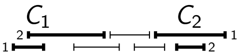
$k \sim \frac{1}{\epsilon}$, $\epsilon \geq \frac{2}{\chi(G)}$

Valencia-Pabon: $\lfloor \left(1 + \frac{\epsilon}{2}\right) \chi(G) \rfloor + \frac{\epsilon}{2}\chi(G)$ colors suffice

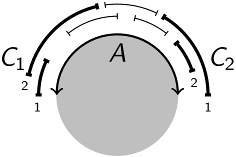
Approximation Factor (2)

Pre-colored Interval Graph G to Circular Arc Graph F:

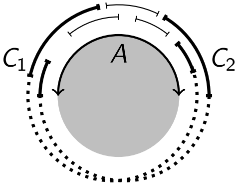
1.



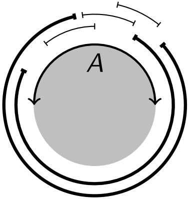
2.



3.



4.



Properties:

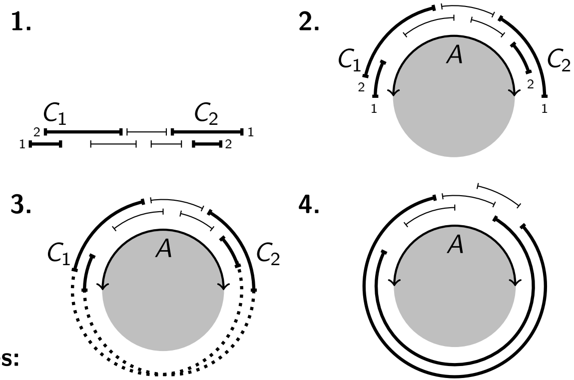
1. Load: $L(F) \leq \chi(G)$
2. Circular cover: $l(F) \geq k - 2$

$k \sim \frac{1}{\epsilon}$, $\epsilon \geq \frac{2}{\chi(G)}$

Valencia-Pabon: $(1 + \epsilon)\,\chi(G)$ colors suffice

Approximation Factor (2)

Pre-colored Interval Graph G to Circular Arc Graph F:



$k \sim \frac{1}{\epsilon}$, $\epsilon \geq \frac{2}{\chi(G)}$

Properties:

1. Load: $L(F) \leq \chi(G)$
2. Circular cover: $l(F) \geq k - 2$

Valencia-Pabon: $(1+\epsilon)\,\chi(G)$ colors suffice

Runtime: $\mathrm{O}(k \log^* n) = \mathrm{O}(\frac{1}{\epsilon} \log^* n)$

Outline

1. $\mathcal{LOCAL}$ model algorithm
2. Adaptation to $\mathcal{CONGEST}$

Algorithm in the $\mathcal{CONGEST}$ model

Adapting the $\mathcal{LOCAL}$ algorithm:

1. $I \leftarrow$ distance-k maximal independent set (identify proper intervals)
2. Nodes of I color inclusive neighborhoods optimally
3. Uncolored nodes form connected components of diameter at most $2k$
4. The node with smaller ID among u, v completes coloring of uncolored nodes between u, v optimally

Assumption:

Interval representation is known

Remaining Difficulty:

Color completion requires knowledge of distance-$\Theta(k)$ neighborhood

Algorithm in the $\mathcal{CONGEST}$ model

Adapting the $\mathcal{LOCAL}$ algorithm:

1. $I \leftarrow$ distance-k maximal independent set (identify proper intervals)
2. Nodes of I color inclusive neighborhoods optimally
3. Uncolored nodes form connected components of diameter at most $2k$
4. The node with smaller ID among u, v completes coloring of uncolored nodes between u, v optimally

Assumption:

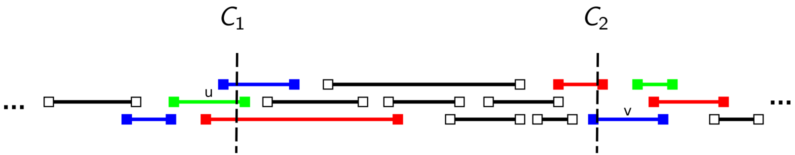
Interval representation is known

Remaining Difficulty:

Color completion requires knowledge of distance-$\Theta(k)$ neighborhood

Implementation in $\mathcal{CONGEST}$ via *Color Rotations*

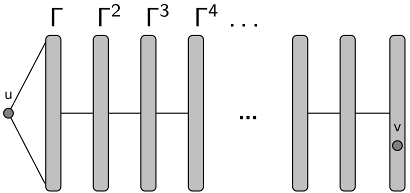
Greedy Colorings

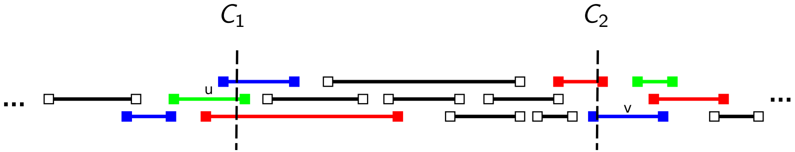


Greedy Coloring Sweep:

Traverse intervals with increasing left boundaries, assign smallest possible color → Optimal coloring

$\mathcal{CONGEST}$ **model version:**

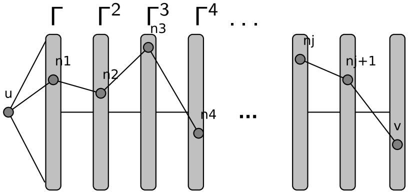




Greedy Coloring Sweep:

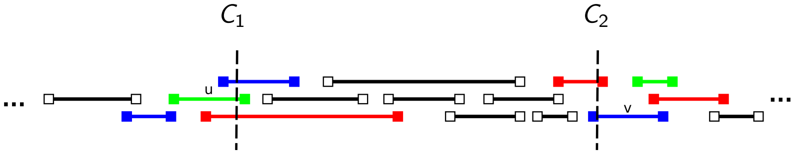
Traverse intervals with increasing left boundaries, assign smallest possible color → Optimal coloring

$\mathcal{CONGEST}$ **model version:**



n_i reaches out furthest to the right, $u, n_1, n_2, \ldots, n_{i+1}, v$ forms path P

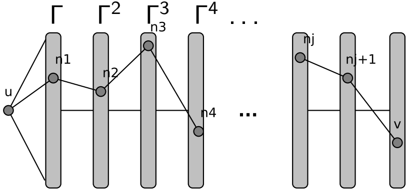
Greedy Colorings



Greedy Coloring Sweep:

Traverse intervals with increasing left boundaries, assign smallest possible color $\rightarrow$ Optimal coloring

$\mathcal{CONGEST}$ **model version:**

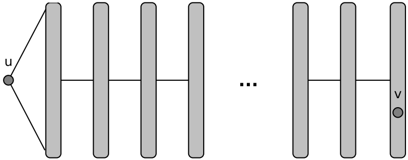


Simulate Greedy coordinated by vertices in P in $\mathrm{O}(k)$ rounds

Color Rotation

Algorithm:

1. u initiates left-to-right Greedy coloring γ_1, respecting colors of $\Gamma[u]$, not respecting colors of $\Gamma[v]$ (initial colors)
2. v initiates right-to-left Greedy coloring γ_2, respecting colors of $\Gamma[v]$, not respecting colors of $\Gamma[u]$ (target colors)
3. Transform γ_1 into a coloring that respects colors of $\Gamma[v]$

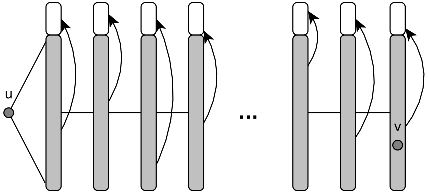


Gray: initial colors γ_1

Color Rotation

Algorithm:

1. u initiates left-to-right Greedy coloring γ_1, respecting colors of $\Gamma[u]$, not respecting colors of $\Gamma[v]$ (initial colors)
2. v initiates right-to-left Greedy coloring γ_2, respecting colors of $\Gamma[v]$, not respecting colors of $\Gamma[u]$ (target colors)
3. Transform γ_1 into a coloring that respects colors of $\Gamma[v]$

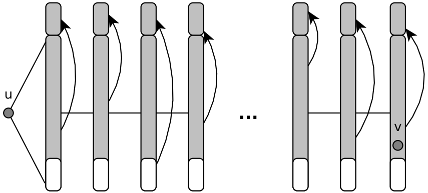


Add $\epsilon\chi(G)$ new colors
Recolor vertices with initial colors $1, \ldots, \epsilon\chi(G)$ to new colors

Color Rotation

Algorithm:

1. u initiates left-to-right Greedy coloring γ_1, respecting colors of $\Gamma[u]$, not respecting colors of $\Gamma[v]$ (initial colors)
2. v initiates right-to-left Greedy coloring γ_2, respecting colors of $\Gamma[v]$, not respecting colors of $\Gamma[u]$ (target colors)
3. Transform γ_1 into a coloring that respects colors of $\Gamma[v]$

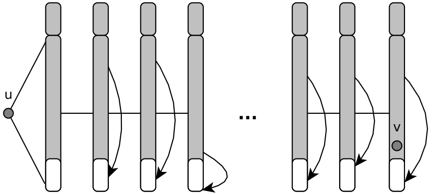


Colors $1, \dots, \epsilon\chi(G)$ are unused

Color Rotation

Algorithm:

1. u initiates left-to-right Greedy coloring γ_1, respecting colors of $\Gamma[u]$, not respecting colors of $\Gamma[v]$ (initial colors)
2. v initiates right-to-left Greedy coloring γ_2, respecting colors of $\Gamma[v]$, not respecting colors of $\Gamma[u]$ (target colors)
3. Transform γ_1 into a coloring that respects colors of $\Gamma[v]$

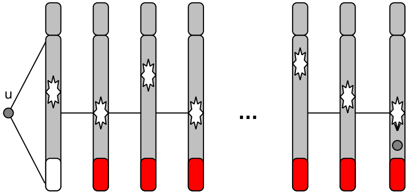


Starting from Γ^2, recolor nodes with target color $1, \ldots, \epsilon\chi(G)$ to their target color

Color Rotation

Algorithm:

1. u initiates left-to-right Greedy coloring γ_1, respecting colors of $\Gamma[u]$, not respecting colors of $\Gamma[v]$ (initial colors)
2. v initiates right-to-left Greedy coloring γ_2, respecting colors of $\Gamma[v]$, not respecting colors of $\Gamma[u]$ (target colors)
3. Transform γ_1 into a coloring that respects colors of $\Gamma[v]$

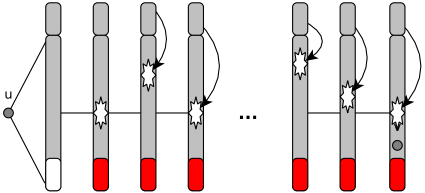


This leaves unused colors behind

Color Rotation

Algorithm:

1. u initiates left-to-right Greedy coloring γ_1, respecting colors of $\Gamma[u]$, not respecting colors of $\Gamma[v]$ (initial colors)
2. v initiates right-to-left Greedy coloring γ_2, respecting colors of $\Gamma[v]$, not respecting colors of $\Gamma[u]$ (target colors)
3. Transform γ_1 into a coloring that respects colors of $\Gamma[v]$

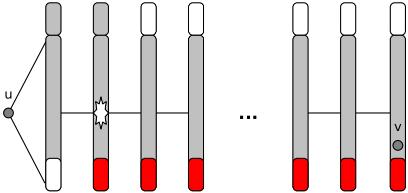


Run a Greedy left-to-right coloring, recoloring colors $\{\epsilon\chi(G) + 1, \ldots, \chi(G)(1+\epsilon)\}$ to $\{\epsilon\chi(G) + 1, \ldots, n\}$

Color Rotation

Algorithm:

1. u initiates left-to-right Greedy coloring γ_1, respecting colors of $\Gamma[u]$, not respecting colors of $\Gamma[v]$ (initial colors)
2. v initiates right-to-left Greedy coloring γ_2, respecting colors of $\Gamma[v]$, not respecting colors of $\Gamma[u]$ (target colors)
3. Transform γ_1 into a coloring that respects colors of $\Gamma[v]$

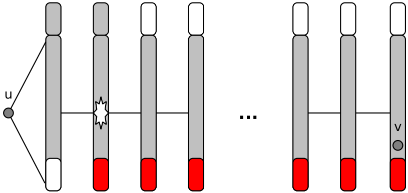


Repeat from Γ^4 onwards

Color Rotation

Algorithm:

1. u initiates left-to-right Greedy coloring γ_1, respecting colors of $\Gamma[u]$, not respecting colors of $\Gamma[v]$ (initial colors)
2. v initiates right-to-left Greedy coloring γ_2, respecting colors of $\Gamma[v]$, not respecting colors of $\Gamma[u]$ (target colors)
3. Transform γ_1 into a coloring that respects colors of $\Gamma[v]$



- Coloring completion step can be implemented in $O(k)$ rounds
- Overall runtime: $O(\frac{1}{\epsilon} \log^* n)$

Conclusion

We presented:

- $(1+\epsilon)$-approximation in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds
- Interval boundaries known: adaptation to $\mathcal{CONGEST}$
- LB: $\Omega(\frac{1}{\epsilon})$ rounds necessary
- $(1+\epsilon)$-approx. for multicoloring directed trees in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds

Open Problems

- Reduce round complexity to $\mathrm{O}(\frac{1}{\epsilon}+\log^* n)$ or prove LB of $\Omega(\frac{1}{\epsilon}\log^* n)$
- $(1+\epsilon)$-approximation on chordal graphs in $\mathrm{O}(\frac{1}{\epsilon}\log n)$ rounds?

Conclusion

We presented:

- $(1+\epsilon)$-approximation in $\mathrm{O}(\frac{1}{\epsilon} \log^* n)$ rounds
- Interval boundaries known: adaptation to $\mathcal{CONGEST}$
- LB: $\Omega(\frac{1}{\epsilon})$ rounds necessary
- $(1+\epsilon)$-approx. for multicoloring directed trees in $\mathrm{O}(\frac{1}{\epsilon} \log^* n)$ rounds

Open Problems

- Reduce round complexity to $\mathrm{O}(\frac{1}{\epsilon} + \log^* n)$ or prove LB of $\Omega(\frac{1}{\epsilon} \log^* n)$
- $(1+\epsilon)$-approximation on chordal graphs in $\mathrm{O}(\frac{1}{\epsilon} \log n)$ rounds?

Thank you