Halldórsson, M. M., & Konrad, C. (2017). Improved Distributed Algorithms for Coloring Interval Graphs with Application to Multicoloring Trees. In S. Das, & S. Tixeuil (Eds.), *Structural Information and Communication Complexity: 24th International Colloquium, SIROCCO 2017, Porquerolles, France, June 19-22, 2017, Revised Selected Papers* (pp. 247-262). (Lecture Notes in Computer Science ; Vol. 10641). https://doi.org/10.1007/978-3-319-72050-0_15

Peer reviewed version

Link to published version (if available):
10.1007/978-3-319-72050-0_15

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

Magnús M. Halldórsson[1] and Christian Konrad[2]

[1] ICE-TCS, School of Computer Science, Reykjavik University, Iceland, mmh@ru.is
[2] Department of Computer Science, Centre for Discrete Mathematics and its Applications (DIMAP), University of Warwick, Coventry, UK, c.konrad@warwick.ac.uk

Abstract. We give a distributed $(1+\epsilon)$-approximation algorithm for the minimum vertex coloring problem on interval graphs, which runs in the $\mathcal{LOCAL}$ model and operates in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds. If nodes are aware of their interval representations, then the algorithm can be adapted to the $\mathcal{CONGEST}$ model using the same number of rounds. Prior to this work, only constant factor approximations using $\mathrm{O}(\log^* n)$ rounds were known [12]. Linial's ring coloring lower bound implies that the dependency on $\log^* n$ cannot be improved. We further prove that the dependency on $\frac{1}{\epsilon}$ is also optimal.
To obtain our $\mathcal{CONGEST}$ model algorithm, we develop a color rotation technique that may be of independent interest. We demonstrate that color rotations can also be applied to obtain a $(1+\epsilon)$-approximate multicoloring of directed trees in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds.

1 Introduction

Vertex coloring problems are central in distributed computing. Given a graph $G = (V, E)$, the objective is to compute an *s-coloring* $\gamma : V \to \{1, 2, \ldots, s\}$ in a distributed fashion, for an integer s, i.e., to assign each vertex one of s colors so that adjacent nodes receive different colors. A substantial amount of research has been carried out on computing $(\Delta+1)$-colorings, where Δ is the maximum degree of the input graph. This is an attractive bound, since it is easy to see that $\Delta + 1$ colors always suffice to color a graph. The quantity $\Delta + 1$ may however be an arbitrarily poor approximation of the *chromatic number* $\chi(G)$ of a graph G, which is the minimum number of colors needed in any coloring[3]. In this paper, we

⋆ M.M.H. is supported by grants 152679-05 and 174484-05 from the Icelandic Research Fund. C.K. is supported by the Centre for Discrete Mathematics and its Applications (DIMAP) at Warwick University and by EPSRC award EP/N011163/1.

[3] For example a complete bipartite graph $G = (A, B, E)$ with $|A| = |B| = n$ can be colored with 2 colors while $\Delta = n$.

are therefore interested in distributed approximation algorithms for the *minimum vertex coloring problem*, which asks for a coloring with $\chi(G)$ colors.

Computational Models. We consider the $\mathcal{LOCAL}$ and $\mathcal{CONGEST}$ models of distributed computation. The input graph $G = (V, E)$ models a communication network, where computational units are located at every node $v \in V$. Graph G also constitutes the problem input. The goal of a distributed coloring algorithm is to compute a (global) coloring, where every node reports its color upon termination of the algorithm. Every node $v \in V$ has a unique identifier $\mathrm{ID}(v)$ and, initially, besides their identifiers, nodes are aware of the identifiers of their neighbors (and hence also of its degree). Messages are exchanged in synchronous communication rounds, where a node may exchange individual messages with each of its neighbors. In the $\mathcal{LOCAL}$ model, messages of unbounded sizes may be exchanged, while in the $\mathcal{CONGEST}$ model, message sizes are limited to $\mathrm{O}(\log n)$, where n is the number of nodes in the input graph. In both models the objective is to minimize the number of communication rounds required to complete the algorithm.

Minimum Vertex Coloring Problem. On general graphs, the minimum vertex coloring problem is NP-complete [14] and even hard to approximate within a factor of $n^{1-\epsilon}$ [23]. Nevertheless, since many distributed models focus on the number of communication rounds rather than the runtime of individual network nodes, it is possible to compute a $O(n^{\epsilon})$-approximation in $\exp\left(O(\frac{1}{\epsilon})\right)$ communication rounds on general graphs [5]. Linial presented a lower bound showing that coloring unoriented d-regular trees with $o(\sqrt{d})$ colors[4] requires $\Omega(\log_d n)$ rounds [16]. This result shows that for every graph class that contains trees, computing a C-approximation requires $\Omega(\log n)$ rounds, for every constant C.

Multicoloring. There are conceptual links between graph coloring and graph *multicoloring*:

Definition 1. *Let $G = (V, E, w)$ be a graph with vertex weights $w : V \to \mathbb{N}$. For an integer $k \geq 1$, a* k-multicoloring *of G is an assignment $\phi : V \to 2^{[k]}$ such that:*

1. *For every $v \in V : |\phi(v)| = w(v)$, and*
2. *For every pair of adjacent vertices $u, v \in V : \phi(u) \cap \phi(v) = \varnothing$.*

[4] Chang, Kopelowitz and Pettie argue in [8] that using a graph construction by Bollobás [7], the same lower bound holds even for colorings with $o(d/\log d)$ colors.

The multichromatic number $\chi^m(G)$ is the largest number of colors needed in every multicoloring of G. In the *minimum vertex multicoloring problem*, the goal is to find a multicoloring that uses $\chi^m(G)$ colors. Distributed algorithms for graph multicoloring find applications in computing MAC schedules (see [15] and the references therein). Kuhn [15] studied a distributed multicoloring problem on general graphs, where a node v of degree $\deg(v)$ receives a $(1-\epsilon)\frac{1}{\deg(v)+1}$ fraction of all colors. Similar to the notion of a $(\Delta+1)$-coloring, this quality bound is given by the degrees of the vertices alone and may be arbitrarily far from an optimal multicoloring.

Results. In this paper, we primarily address the minimum vertex coloring problem in *interval graphs*, which are the intersection graphs of intervals on the line. Interval graphs do not contain trees and the previously mentioned lower bound thus does not apply. In a previous work, we gave a distributed 8-approximation algorithm that runs in $\mathrm{O}(\log^* n)$ in the $\mathcal{LOCAL}$, and an extension of the algorithm to the $\mathcal{CONGEST}$ model if the representation of the intervals are known by the network nodes [12]. We improve on [12] and give distributed $(1+\epsilon)$-approximation algorithms for coloring interval graphs for both the $\mathcal{LOCAL}$ and $\mathcal{CONGEST}$ models, which run in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds. Similar to [12], our $\mathcal{CONGEST}$ model algorithm requires that nodes are aware of their interval boundaries.

We further employ connections between coloring interval graphs and multicoloring paths and directed trees. We prove that every $(1+\epsilon)$-approximation algorithm for multicoloring the path requires $\Omega(\frac{1}{\epsilon})$ rounds. Since every $\mathcal{LOCAL}$ algorithm for coloring interval graphs can be used for multicoloring the path, this lower bound thus also holds for coloring interval graphs.

Our $\mathcal{CONGEST}$ model algorithm uses a color rotation technique that may be of independent interest. We demonstrate that this technique finds applications for other coloring problems as well: We present a $(1+\epsilon)$-approximation algorithm for multicoloring directed trees using color rotations, which runs in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds in the $\mathcal{CONGEST}$ model. This is the first distributed algorithm for multicoloring problems with non-trivial approximation guarantee.

Techniques. The $\mathcal{LOCAL}$ model algorithm of [12] simulates the sequential Greedy coloring algorithm, which traverses the vertices in an arbitrary order and assigns the smallest color possible. The approximation guarantee of their algorithm follows from the fact that a Greedy coloring of interval graphs gives an 8-approximation [21]. They construct a dominating set, which can be colored using few colors, via a somewhat technical algorithm.

Our strategy is arguably simpler. We first compute a maximal distance k-independent set I ($k = \Theta(\frac{1}{\epsilon})$) using an algorithm of Schneider and Wattenhofer [19]. Then, nodes of I color their inclusive neighborhoods optimally. Notice that every inclusive neighborhood $\Gamma[v]$ of a vertex v is a separator in interval graphs. The set of yet uncolored nodes thus form connected components of diameter $\Theta(k)$. Using a theorem about the chromatic number of circular arc graphs by Valencia-Pabon [22], we show that there exists a completion of the current partial coloring to a $(1+\epsilon)$-approximate coloring of the entire graph. Nodes of I then color the connected components of uncolored nodes using the optimal color completion.

This approach relies heavily on the ability to send messages of unbounded sizes. When nodes of I color connected components of uncolored nodes, they first need to collect the topology of entire components, which cannot be done in the $\mathcal{CONGEST}$ model. To overcome this difficulty, we develop a *color rotation* technique: Let $u, v \in I$ be nodes of the distance-k maximal independent set such that u is located left of v and there is no other node of I between them (the distance between u and v is thus at most $2k$). Suppose that their local neighborhoods have already been colored optimally. We show that a Greedy left-to-right coloring sweep can be initiated at u that respects the colors of u's neighborhood and colors all nodes between u and v. This coloring however does not necessarily respect the colors of v's neighborhood. Similarly, a right-to-left coloring with similar properties is initiated by v. Since a Greedy coloring that processes the vertices with increasing left boundaries (or decreasing right boundaries) gives an optimal coloring in interval graphs, the two coloring sweeps produce optimal colorings. We then apply our color rotation technique: Guided by the right-to-left coloring, we perform color rotations using few additional colors that transform the left-to-right coloring into a coloring that respects the colors of the neighborhood of v, giving a $(1+\epsilon)$-approximation.

We demonstrate that the color rotation technique can be applied for multicoloring directed trees as well. Our algorithm first computes a partitioning of the input tree into subtrees, which are then colored independently. The potential color conflicts between subtrees are then resolved via color rotations. In order to obtain the partitioning of the input tree into subtrees, we develop an algorithm for computing a particular *ruling set*, which may be of independent interest.

Further Related Work. To our best knowledge, the only works that explicitly address distributed algorithms for the minimum vertex coloring problem are the already mentioned algorithms by Barenboim et al. [5] for general graphs and our previous work on coloring interval graphs [12]. Goldberg, Plotkin and Shannon [11] gave a 7-coloring algorithm of planar graphs, which runs in $\mathrm{O}(\log n)$ rounds, and a 5-coloring algorithm, which runs in $\mathrm{O}(\log n \log\log n)$ rounds and requires the planar representation of the input graph. Barenboim and Elkin [3] gave a $(\lfloor (2+\epsilon)a \rfloor + 1)$-coloring algorithm for graphs of arboricity a, which runs in $\mathrm{O}(a \log n)$ rounds and thus subsumes the previously mentioned 7-coloring algorithm (planar graphs have arboricity at most 3). Schneider, Elkin and Wattenhofer gave a $((1 - \mathrm{O}(\chi(G)))\Delta)$-coloring algorithm whose runtime depends on the chromatic number $\chi(G)$ [18].

Due to the similarity in the problem statement, we also expand on distributed $(\Delta{+}1)$-coloring algorithms: The first randomized $(\Delta{+}1)$-coloring algorithm uses a reduction to the maximal independent set problem given by Luby [17]. Since the maximal independent set problem can be solved in $\mathrm{O}(\log n)$ time via the algorithms of Luby [17] or Alon, Babai and Itai [1], a $\mathrm{O}(\log n)$ rounds algorithm is obtained. Improved randomized algorithms were given by Schneider and Wattenhofer [20], which runs in $\mathrm{O}(\log \Delta + \sqrt{\log n})$ rounds, and later by Barenboim et al. [6], which runs in $\mathrm{O}(\log \Delta) + 2^{\mathrm{O}(\sqrt{\log\log n})}$ rounds. Very recently, the first randomized algorithm, which runs in $o(\log n)$ rounds for any value of Δ, was presented. Harris, Schneider and Hsin-Hao Su showed that $\mathrm{O}(\sqrt{\log \Delta}) + 2^{\mathrm{O}(\sqrt{\log\log n})}$ suffice [13]. Deterministic $(\Delta + 1)$-coloring algorithms have been extensively studied as well. The currently fastest algorithm is by Fraigniaud, Heinrich and Kosowski [10] and uses $\mathrm{O}(\sqrt{\Delta} \log^{2.5} \Delta + \log^* n)$. This result improved on Barenboim's algorithm [2], which uses $\mathrm{O}(\Delta^{\frac{3}{4}} \log \Delta + \log^* n)$ rounds and was the first deterministic $(\Delta + 1)$-coloring algorithm which achieved a sublinear in Δ number of rounds. Faster $(\Delta + 1)$-colorings can be achieved on special graph classes. The well-known Cole-Vishkin algorithm [9] colors cycles (and directed trees) using 3 colors in $\mathrm{O}(\log^* n)$ rounds, which is best possible due to a lower bound given by Linial [16]. This algorithm has been extended to bounded-independence graphs by Schneider and Wattenhofer [19]. For further references, we refer the reader to the survey by Barenboim and Elkin [4] and the references therein.

Outline. We give notations and definitions in Section 2. Then we present our coloring algorithms for interval graphs in Section 3 and our lower bound in Section 4. Our result on multicoloring directed trees is then given in Section 5. Finally, we conclude in Section 6.

2 Preliminaries

Definitions. A *distance-k independent set* in a graph $G = (V, E)$ is a subset of vertices $I \subseteq V$ such that every pair of vertices $v_1, v_2 \in I$ is at distance at least k. A distance-k independent set I is *maximal* if $I \cup v$ is not a distance-k independent set, for all $v \in V \setminus I$. We call a distance-2 independent set simply *independent set.* For an integer k, a *distance-k-coloring* of a graph $G = (V, E)$ is an assignment $\gamma : V \to \{1, \ldots, s\}$ of s colors to the vertices such that every pair of vertices at distance at most k receives different colors. A *partial coloring* of a graph $G = (V, E)$ is an assignment $\gamma : V \to \{1, \ldots, s\} \cup \{\perp\}$, where uncolored nodes are assigned the symbol $\perp$.

For simplicity, we assume that the input graphs are connected. The *neighborhood* of a vertex v in graph G is denoted by $\Gamma_G(v)$, and we define the inclusive neighborhood of v by $\Gamma_G[v] = \Gamma_G(v) \cup \{v\}$. For a subset $V' \subseteq V$, we write $\Gamma_{V'}(v)$ to denote $\Gamma_G(v) \cap V'$. Furthermore, the *k-neighborhood* of a vertex v is the set of nodes that are within distance at most k from v (excluding v), and we denote it by $\Gamma_G^k(v)$. Then $\Gamma_G^1(v) = \Gamma_G(v)$. For a vertex $v \in V$, we denote by $\deg_G(v)$ the degree of v in G. For a subset $V' \subseteq V$, we may also write $\deg_{V'}(v)$ for the degree of v in the subgraph of G induced by the nodes V', that is, $\deg_{V'}(v) := \deg_{G|_{V'}}(v)$.

Interval Graphs. Let $V = \{v_1, \ldots, v_n\}$ be a set of intervals with $v_j = (a_j, b_j)$ for all $1 \leq j \leq n$ and real numbers a_j, b_j such that $a_j < b_j$. We assume that all a_i, b_i are distinct. Let $G = (V, E)$ be the corresponding interval graph, i.e., there is an edge between vertices (intervals) v_j, v_k if the two intervals intersect. We denote the number of edges by m.

We say that an interval v is *proper* if no other interval u satisfies $\Gamma_G[v] \subsetneq \Gamma_G[u]$. For an interval graph $G = (V, E)$, we denote by $G_P = (V_P, E|_{V_P})$ the subgraph of G induced by the proper intervals of G. It is easy to see that G_P is connected if G is connected as well.

Graph G_P is of bounded-independence, a property that restricts the sizes of maximum independent sets in local neighborhoods, formally defined as follows:

Definition 2 (Bounded-independence Graphs). *A graph $G = (V, E)$ is of* bounded-independence *if there is a bounding function $f(r)$ such that for each node $v \in V$, the size of a maximum independent set in the r-neighborhood of v is at most $f(r), \forall r \geq 0$.*

Schneider and Wattenhofer [19] gave a distributed maximal independent set algorithm for graphs of bounded-independence that runs in time

$O(\log^* n)$. We denote this algorithm by MisBI. It can be implemented in the $\mathcal{CONGEST}$ model.

3 Algorithms for Coloring Interval Graphs

We first give our algorithm for the $\mathcal{LOCAL}$ model in Section 3.1 and then show how to extend the algorithm to the $\mathcal{CONGEST}$ model in Section 3.2.

3.1 Algorithm in the $\mathcal{LOCAL}$ model

Our algorithm is depicted and explained in Algorithm 1. It is parametrized by an integer k, which determines the approximation guarantee and whose precise value is determined later.

Algorithm 1 Algorithm for Coloring Interval Graphs in the $\mathcal{LOCAL}$ Model

---

1. **Identify the subgraph G_P of proper intervals:** Each node v checks if it has a neighbor u with $\Gamma_G[u] \supsetneq \Gamma_G[v]$. If no such neighbor exists then v is in G_P. This involves a single communication round where v sends the list of its neighbors to all its neighbors. In one additional round, each node v informs its neighbors whether $v \in G_P$.
2. **Compute a distance-k maximal independent set J of G_P:** The nodes simulate MisBI on graph G_P^k, where nodes are adjacent if they are at distance at most k in G_P, in $O(k \cdot \log^* n)$ rounds. The result is a distance-k maximal independent set J of G_P.
3. **Color inclusive neighborhoods of J:** Each dominator $v \in J$ colors its inclusive neighborhood $\Gamma_G[v]$ optimally using at most $\chi(G)$ colors.
4. **Color remaining nodes:** For any two dominators u, v with $dist(u, v) < 2k$ and $ID(u) > ID(v)$, u colors all uncolored nodes between u and v in $O(k)$ rounds as follows: u collects its $2k$-neighborhood including the color constraints given by the already colored neighborhood of v. The best coloring of the remaining nodes is computed locally and newly colored nodes are informed of their color.

The key part of our analysis is to show that Step 4 of the algorithm does not require too many colors. To this end, we employ a result by Valencia-Pabon [22] on coloring *circular-arc graphs*, which are the intersection graphs of a set of arcs on a circle. Given a circular arc graph F, the *load* $L(F)$ is the cardinality of the largest subset of arcs containing

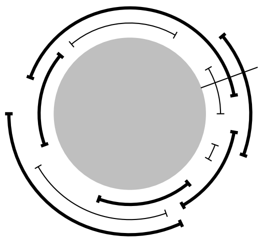

Fig. 1. A 3-colorable circular arc graph F with load $L(F) = 3$ and circular-cover $l(F) = 6$. The vertices of a circular-cover are illustrated in bold. The result of Valencia-Pabon (Theorem 1) gives an upper bound of four colors.

the same point. The *circular-cover* $l(F)$ is the cardinality of the smallest subset of arcs that cover the entire circle. See Figure 1 for an example.

Theorem 1 (Valencia-Pabon [22]). *Let F be a circular arc graph with load L and circular-cover $l \geq 4$. Then $\lfloor \left(1 + \frac{1}{l-2}\right) L \rfloor + 1$ colors suffice to color F.*

Equipped with Theorem 1, we prove now the existence of a good coloring that is required in Step 4 of our algorithm.

Lemma 1. *Let $G = (V, E)$ be an interval graph, $C_1, C_2 \subseteq V$ disjoint maximal cliques such that $dist(v_1, v_2) \geq l$ for every pair of nodes $v_1 \in C_1, v_2 \in C_2$, for an integer $l \geq 5$. Let $G' = G[C_1 \cup C_2 \cup D]$, where $D \subseteq V$ is the set of intervals located between C_1 and C_2. Let γ be a partial coloring of G' such that $\gamma(v) \in [\chi(G')]$ for every $v \in C_1 \cup C_2$, and $\gamma(v) = \perp$ for $v \in D$. Then, γ can be extended to a coloring that employs at most $\lfloor (1 + \frac{1}{l-3})\chi(G') \rfloor + 1$ colors.*

Proof. Let F be the graph obtained from G' by contracting every pair of vertices $v_1 \in C_1$, $v_2 \in C_2$ with $\gamma(v_1) = \gamma(v_2)$. We will argue that F is a circular arc graph with load $\chi(G')$ and circular-cover $l - 1$. Our result then follows from Theorem 1.

A representation of F with circular arcs can be obtained by, first, wrapping the line segment that contains all intervals of G' onto an arc $A \subsetneq C$ of a circle C, and then replacing every pair of intervals/arcs $v_1 \in C_1, v_2 \in C_2$ with $\gamma(v_1) = \gamma(v_2)$ with an arc of minimal length that includes v_1, v_2 and all points of $C \setminus A$. See Figure 2 for an illustration.

Since we replaced at most $\chi(G')$ pairs of arcs with arcs that cover $C \setminus A$, every point of the arc $C \setminus A$ is covered by at most $\chi(G')$ arcs.

Furthermore, since all points on the arc A are covered by at most $\chi(G')$ arcs, we obtain $L(F) = \chi(G')$. The circular-cover is at least the length of the shortest path in G' from C_1 to C_2 minus one, i.e., $l-1$, since all arcs of D need to be covered, and pairs of nodes of C_1 and C_2 are contracted in F. $\square$

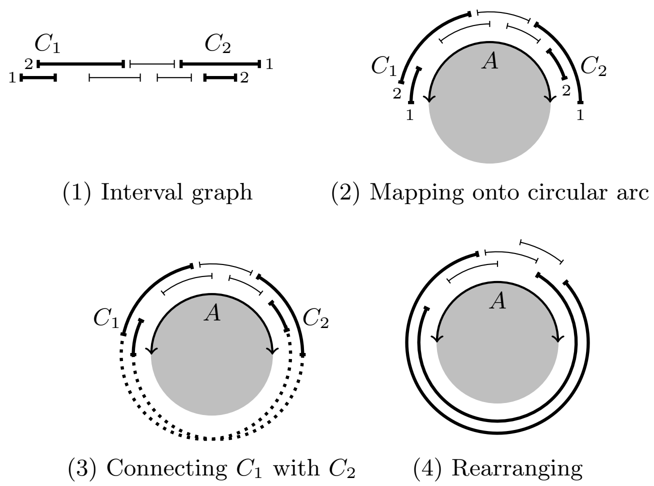

Fig. 2. Construction used in the proof of Lemma 1 . Edges of the maximal cliques C_1 and C_2 are illustrated in bold. The colors of the intervals in C_1 and C_2 are indicated by the small numbers next to the intervals. In this example, a path of length 7 is mapped onto an arc A. Then pairs of vertices with the same color of the maximal cliques C_1 and C_2 are connected, which gives a cycle of length 5. Every coloring of C_5 requires 3 colors.

Using Lemma 1, we show next that our algorithm gives a $(1+\epsilon)$-approximation guarantee:

Theorem 2. *Let $G = (V, E)$ be an interval graph and suppose that $\epsilon \geq \frac{2}{\chi(G)}$. Then, there is a deterministic $(1+\epsilon)$-approximation algorithm for coloring interval graphs in the $\mathcal{LOCAL}$ model that runs in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds.*

Proof. Let $k = \lceil \frac{2}{\epsilon} \rceil + 5$. Our algorithm computes a distance-k maximal independent set J in the subgraph of proper intervals. Observe that J is also a distance-k independent set in G, since for every shortest path in G between two nodes of V_P, there is one that only traverses edges of G_P. Let u, v be two nodes as in step 4 of the algorithm such that u is left of

v. Let C_1 be the set of intervals that intersect u's right boundary, and let C_2 be the set of intervals that intersect v's left boundary. Then, C_1 and C_2 are maximal cliques and were colored in step 3 of the algorithm. Since J is a distance-k independent set, every pair of nodes of C_1 and C_2 are at a distance of at least $k-2$. Then, by Lemma 1, the coloring can be completed using $\chi(G)(1+\frac{1}{(k-2)-3})+1 = \chi(G)(1+1/\lceil\frac{2}{\epsilon}\rceil)+1$ colors, which simplifies to:

$$\chi(G)(1+\frac{1}{\lceil\frac{2}{\epsilon}\rceil})+1 \leq \chi(G)(1+\frac{\epsilon}{2})+\frac{\epsilon\chi(G)}{2} \leq (1+\epsilon)\chi(G) \ ,$$

where we used the assumption $\epsilon > \frac{2}{\chi(G)}$ which implies $1 \leq \frac{\chi(G)\epsilon}{2}$. The runtime of the algorithm is dominated by the computation of the distance-k maximal independent set in step 2, which runs in $\mathrm{O}(k \cdot \log^* n) = \mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds. □

3.2 Adapting the Algorithm to the $\mathcal{CONGEST}$ Model

We now adapt the previous algorithm to the $\mathcal{CONGEST}$ model when each node $v_i \in V$ knows its interval representation boundaries a_i, b_i. We assume that representing the numbers a_i, b_i uses logarithmic space.

We reuse Steps 1 and 3 from our previous algorithm. Step 2, i.e., finding a distance-k maximal independent set in G_P by running MisBI on G_P^k cannot be implemented in the $\mathcal{CONGEST}$ model, since the nodes cannot collect their full distance-k neighborhoods quickly. Instead, we replace this step by a subroutine that finds a $(k, 3k/2+1)$-*ruling set*. A (p,q)-ruling set in graph $G=(V,E)$ is a subset of vertices $I \subseteq V$ such that every pair of vertices in I are at distance at least p, while every vertex outside I is at distance at most q from a vertex in I. We will argue that such a set can be computed in the $\mathcal{CONGEST}$ model in $\mathrm{O}(k\log^* n)$ rounds on graph G_P. Step 4 is replaced by a more technical coloring process.

Regarding Step 1, exchanging interval boundaries is enough in order to determine whether a node $v \in V$ is also in V_P. For a node $v \in J$ to color its neighborhood optimally in Step 3, it requires only the interval representation of its neighboring nodes in order to determine the neighborhood relations among them. This information can be exchanged in one round in the $\mathcal{CONGEST}$ model.

Computing a $(k, 3k/2+1)$-ruling set in G_P. We next argue how to compute a $(k, 3k/2+1)$-ruling set in G_P, the subgraph of proper intervals. First, we compute a maximal independent set I_1 in G_P using MisBI. We then proceed inductively. Let $v_1, v_2, \ldots$ denote the intervals of I_j ordered from left to right. We build an auxiliary graph G_j on vertex set I_j, where node v_i is adjacent to nodes v_{i-1} and v_{i+1}. Notice that G_j is an interval graph (it is in fact a path). We then compute a maximal independent set I_{j+1} in graph G_j using MisBI (or for example the Cole-Vishkin algorithm [9]).

We now argue that $I_{\lceil \log k \rceil}$ is a $(k, 3k/2+1)$-ruling set in G_P, and the computation of I_k can be implemented in the $\mathcal{CONGEST}$ model in $\mathrm{O}(k \log^* n)$ rounds. Let $v_1, v_2, \ldots$ denote the intervals of I_j ordered from left to right. Let l_j and u_j denote the minimum and maximum distance between vertices in I_j, respectively. Since I_1 is a maximal independent set, we have $l_1 \geq 2$ and $u_1 \leq 3$. Then, it is easy to see that $l_{j+1} \geq 2l_j \geq 2^{j+1}$ and $u_{j+1} \leq 2u_j \leq 3 \cdot 2^j$. Thus, $l_{\lceil \log k \rceil} \geq k$ and $u_{\lceil \log k \rceil} \leq 3k$. Furthermore, every vertex outside $I_{\lceil \log k \rceil}$ is at distance at most $3k/2+1$ from a vertex of $I_{\lceil \log k \rceil}$.

Concerning the runtime, simulating the run of MisBI on G_j increases the runtime of MisBI by a factor of u_j. Hence, computing G_j requires $\mathrm{O}(3 \cdot 2^{j-1} \log^* n) = \mathrm{O}(2^j \log^* n)$ rounds. Overall the runtime for computing $I_{\lceil \log k \rceil}$ is $\sum_{j \leq \lceil \log k \rceil} \mathrm{O}(2^j \log^* n) = \mathrm{O}(k \log^* n)$ rounds.

Coloring. After Step 3 of the algorithm, we have computed a $(k, 3k/2+1)$-ruling set J and a partial coloring γ such that nodes $\cup_{w \in J} \Gamma_G[w]$ are colored while all other nodes are uncolored, i.e., $\gamma(z) \in [\chi(G)]$, if $z \in \cup_{w \in J} \Gamma_G[w]$, and $\gamma(z) = \perp$, otherwise. Every pair of adjacent nodes $u, v \in J$ with u located left of v executes the coloring procedure presented in the following in order to color the uncolored nodes located between them.

Fix two nodes $u, v \in J$ as described above. In the description of our algorithm, we use the following notations. Denote by C_1 (C_2) the maximal clique consisting of intervals that intersect u's right boundary (resp. v's left boundary). Let D be the set of intervals outside $C_1 \cup C_2$ located between u and v. Let $N_i \subseteq C_1 \cup C_2 \cup D$ be the set of nodes at distance i from u, and let $N_{\geq i} = \cup_{j \geq i} N_j$. We also ensure that every node $a \in D$ learns its distance from u (and thus the index i such that $a \in N_i$). This can be established by flooding the network with a token initially broadcasted by u, and the number of rounds it takes until the token reaches node $a \in D$ equals $dist_G(a, u)$. Denote by n_i the interval of N_i that reaches out furthest to the right. Nodes of N_i can identify this interval easily by communicating their interval boundaries to their neighbors.

Our coloring procedure requires an implementation of the Greedy coloring algorithm as a subroutine in the $\mathcal{CONGEST}$ model.

Greedy Coloring Subroutine. W.l.o.g. we present a left-to-right coloring initiated by node u; a right-to-left coloring initiated by v can be obtained similarly. First, node n_1 colors the uncolored nodes of its neighborhood: It traverses its uncolored neighbors with increasing left interval boundary and assigns the smallest possible color. Then, n_2 continues with the same process. The coloring carries on until all n_i have colored their neighborhoods. The runtime of this procedure is $\mathrm{O}(dist_G(u,v)) = \mathrm{O}(k)$.

Coloring Process. Node u initiates a left-to-right Greedy coloring γ_1 that respects the colors given by γ of C_1 (and not necessarily the colors of C_2), and simultaneously, v initiates a right-to-left Greedy coloring γ_2 that respects the colors given by γ of C_2 (and not necessarily the colors of C_1). We then transform the coloring γ_1 into one that respects the colors γ_2 on C_2.

Our algorithm operates in p phases, each consisting of three recoloring steps. In phase i, we alter the coloring γ_1 of nodes $N_{\geq 3i}$ such that γ_1 is non-conflicting with colors $T_i = \{(i-1)B+1, \ldots, iB\}$ of γ on C_2, where $B = \lceil \frac{\chi(G)}{p} \rceil$. To this end, nodes of $N_{\geq 3i}$ with a color of T_i recolor themselves to new colors $[\chi(G)+1, \chi(G)+B]$ by adding $\chi(G)-(i-1)B$ to their own color. Then, nodes of $N_{\geq 3i+1}$ with a target color (the color given by γ_2) in T_i recolor themselves to their target color. Last, we initiate a Greedy recoloring sweep at node n_{3i+2} that recolors all nodes of $N_{\geq 3i+2}$ with a current color $> iB$ to colors in $\{iB+1, \ldots, \chi(G)\}$.

We prove correctness of this algorithm in the following lemma.

Lemma 2. *After phase i of the previous coloring process, the following holds:*

1. $\forall w \in N_{\geq 3i+2} : \gamma_1(w) \in [\chi(G)]$,
2. $\forall w \in N_{\geq 3i+1} : \gamma_2(w) \leq iB \Rightarrow \gamma_1(w) = \gamma_2(w)$,
3. γ_1 *is legal.*

Proof. Before iteration one (i.e. $i = 0$), all three items are trivially true. The first recoloring step of phase i assigns nodes of $N_{\geq 3i}$ with current color in T_i a color larger than $\chi(G)$. Note that this leads to a legal coloring, since by Item 1, none of the nodes of N_{3i-1} are colored with a color larger than $\chi(G)$. In the second recoloring step, nodes of $N_{\geq 3i+1}$ with target color in T_i receive their target color (which gives Item 2). Again, γ_1 remains legal since after the first recoloring step, none of the nodes of N_{3i} are colored with a color in T_i. In the third recoloring step, the Greedy coloring algorithm is executed on the subgraph induced by nodes $V_i = \{v \in N_{\geq 3i+2} : \gamma_1(v) \geq iB+1\}$. We claim that the algorithm recolors

V_i with colors in $[\chi(G)]$. Indeed, first note that $\chi(G[V_i]) \leq \chi(G) - iB$, since γ_2 restricted to V_i gives such a coloring. Next, since the Greedy coloring algorithm processes the intervals with increasing left boundary, all color constraints when coloring an interval x are imposed by intervals that intersect x's left boundary (note that for two intervals $x \in N_j$, $y \in N_{j+1}$ we always have $l(x) < l(y)$). Since there are at most $\omega(G[V_i]) - 1 = \chi(G[V_i]) - 1$ such intervals, there is always an available color for x in $[\chi(G)]$, which proves Item 1. Since legality of γ_1 is preserved throughout the three recoloring steps, Item 1 follows. $\square$

This gives the following theorem:

Theorem 3. *Let $G = (V, E)$ be an interval graph and suppose that $\epsilon \geq \frac{2}{\chi(G)}$. Then, there is a deterministic $(1+\epsilon)$-approximation algorithm for coloring interval graphs in the $\mathcal{CONGEST}$ model that runs in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds.*

Proof. Correctness of the algorithm was established in Lemma 2. By construction, the algorithm uses at most $\chi(G)(1+\frac{1}{p})+1$ colors. Thus, we set $p = \lceil \frac{2}{\epsilon} \rceil$ which implies that the number of colors is bounded by $(1+\epsilon)\chi(G)$, which proves the approximation guarantee. In order to execute p phases of the color rotation algorithm, it is necessary that adjacent nodes of J are far enough apart. To this end, we set parameter k to $k = 3p + 2 = \mathrm{O}(\frac{1}{\epsilon})$.

Concerning the runtime of the algorithm, besides an $\mathrm{O}(k \log^* n)$ term for the computation of the $(k,\ 3k/2+1)$-ruling set, an additive $\mathrm{O}(k^2)$ term is incurred by the Greedy coloring algorithm: In each of the $\mathrm{O}(k)$ phases, we execute the Greedy coloring algorithm which requires $\mathrm{O}(k)$ steps. We argue now that the k^2 term can be reduced to k by pipelining the Greedy recoloring sweeps. Iteration i can be started as soon as nodes N_{3i+2} have been recolored by the recoloring sweep initiated in iteration $i-1$. After the initiation of the recoloring sweep of iteration $i-1$, it takes only a constant number of iterations until this sweep reaches nodes N_{3i+2}. Thus, iteration i can be started after a constant number of iterations after the start of iteration $i-1$. Thus, by induction, iteration k can be started $\mathrm{O}(k)$ iterations after iteration 1 has been started. The overall runtime is thus $\mathrm{O}(k \log^* n + k) = \mathrm{O}(\frac{1}{\epsilon}\log^* n)$. $\square$

Remark: In the recoloring step, we assume that nodes know $\chi(G)$. This can be circumvented as follows: The initial left-to-right coloring γ_1 is an optimal coloring of nodes $C_1 \cup C_2 \cup D$. Nodes in $C_1 \cup C_2 \cup D$ compute the largest color employed by γ_1. This value replaces $\chi(G)$ in the algorithm.

4 Lower Bound for Coloring Interval Graphs in the $\mathcal{LOCAL}$ Model

Linial's lower bound shows that every distributed algorithm for coloring the n-cycle with three colors requires time $\Omega(\log^* n)$. Since it is possible to decrement the number of colors of a c-coloring, for $c \geq \Delta + 2$, in a single communication rounds using a standard method, Linial's lower bound even holds for coloring the ring with $\mathrm{O}(\log^* n)$ colors. Furthermore, this lower bound can easily be adapted to hold for a path of length n (which is also an interval graph and can be colored with two colors). it follows that computing a $\mathrm{O}(\log^* n)$-approximate interval coloring requires $\Omega(\log^* n)$ rounds.

We present now a different lower bound argument that holds for interval graphs with arbitrary chromatic number. Specifically, we show that every distributed $(1+\epsilon)$-approximation algorithm for interval coloring requires $\Omega(\frac{1}{\epsilon})$ rounds.

To this end, we give a lower bound for multicoloring a path and provide a reduction between coloring intervals and path multicoloring.

Let $G(V, E, w)$ be a weighted path on n vertices with $w(v) = k$, for every $v \in V$. Then the multichromatic number of G is $\chi^m(G) = 2k$: alternate between the first k and the second k colors while traversing the path from left to right. We prove now that if ϕ is a $(1+\epsilon)$-approximate multicoloring of G, then the color sets of nodes at even distances have a large intersection and the color sets of nodes at odd distances have small intersection.

Lemma 3. *Let $\phi : V \to 2^{\mathbb{N}}$ be a $(1+\epsilon)$-approximate multicoloring of a path $G = (V, E, w)$ with $w(v) = k$, for every $v \in V$. Then, for $u, v \in V$ and an integer $r \geq 1$:*

1. *If $dist(u, v) = 2r$ then $|\phi(u) \cap \phi(v)| \geq k - 2kr\epsilon$,*
2. *If $dist(u, v) = 2r + 1$ then $|\phi(u) \cap \phi(v)| \leq 2kr\epsilon$.*

Proof. Since ϕ is a $(1+\epsilon)$-approximate multicoloring of G, we have $|\phi(V)| \leq 2(1+\epsilon)k$. Let $v_0, v_1, \dots$ denote the vertices of the path so that v_i and v_{i+1} are adjacent. Then, by Item 1 of Definition 1, it holds that $|\phi(v_i) \cap \phi(v_{i+1})| = 0$. We further have $|\phi(v_i) \cap \phi(v_{i+2})| \geq k - 2\epsilon k$, since the total number of colors employed is bounded by

$$\begin{aligned} 2(1+\epsilon)k &\geq |\phi(v_{i+1})| + |\phi(v_i)| + |\phi(v_{i+2})| - |\phi(v_i) \cap \phi(v_{i+2})| \\ &= 3k - |\phi(v_i) \cap \phi(v_{i+2})|, \end{aligned}$$

which implies the claimed bound. Next, we use the relationship

$$\begin{aligned}|\phi(v_0) \cap \phi(v_{2r})| &\geq |\phi(v_0) \cap \phi(v_{2r-2})| - |\phi(v_{2r}) \setminus \phi(v_{2r-2})| \\ &= |\phi(v_0) \cap \phi(v_{2r-2})| - (k - |\phi(v_{2r}) \cap \phi(v_{2r-2})|) \\ &\geq |\phi(v_0) \cap \phi(v_{2r-2})| - 2\epsilon k,\end{aligned}$$

which implies $|\phi(v_0) \cap \phi(v_{2r})| \geq k(1 - 2r\epsilon)$ and proves Item 1. Last, since $|\phi(v_0) \cap \phi(v_1)| = 0$ and $|\phi(v_1) \cap \phi(v_{2r+1})| \geq k(1 - 2r\epsilon)$, we obtain

$$|\phi(u) \cap \phi(v_{2r+1})| \leq |\phi(v_{2r+1}) \setminus \phi(v_1)| \leq k - |\phi(v_1) \cap \phi(v_{2r+1})| = 2kr\epsilon,$$

which proves Item 2. □

Equipped with Lemma 3 we are ready to prove our lower bound on computing a multicoloring on the path. Let $G = (V, E, w)$ denote the path of length n with $w(v) = k$, for every $v \in V$, and suppose that every vertex v receives a unique label $\mathcal{L}(v)$, where $\mathcal{L}$ is chosen uniformly at random from the set of bijections between V and $\{1, \ldots, n\}$.

Theorem 4. *Every possibly randomized distributed algorithm with error probability at most $1/3$ that computes a $(1+\epsilon)$-approximate multicoloring on a path $G = (V, E, w)$ with vertex weights $w(v) = k$, for every $v \in V$, requires at least $\frac{1}{4\epsilon} - \frac{1}{2}$ rounds.*

Proof. Let $v_1, \ldots, v_n$ denote the vertices of G such that v_i and v_{i+1} are adjacent, and let ϕ denote the output multicoloring of the algorithm. Suppose that the algorithm runs in r rounds. Consider an index j such that $r + 1 \leq j \leq n - 3r - 2$. Then, since the error probability of the algorithm is at most $1/3$ and by applying Lemma 3, we obtain

$$\mathbb{P}\left[|\phi(v_j) \cap \phi(v_{j+2r+2})| \geq k - 2k(r+1)\epsilon\right] \geq 2/3 \ , \quad (1)$$

$$\mathbb{P}\left[|\phi(v_j) \cap \phi(v_{j+2r+1})| \leq 2kr\epsilon\right] \geq 2/3 \ , \quad (2)$$

where the probabilities are taken over the coin flips of the nodes and the labeling function $\mathcal{L}$. Since the outputs of two nodes at distance at least $2r + 1$ are independent (the output of a node is a function of the labels and random coin flips in its r-neighborhood), for every integer c we obtain $\mathbb{P}\left[|\phi(v_j) \cap \phi(v_{j+2r+2})| = c\right] = \mathbb{P}\left[|\phi(v_j) \cap \phi(v_{j+2r+1})| = c\right]$. Thus, Inequality 1 implies

$$\mathbb{P}\left[|\phi(v_j) \cap \phi(v_{j+2r+1})| \geq k - 2k(r+1)\epsilon\right] \geq 2/3 \ . \quad (3)$$

Suppose now that $r < \frac{1}{4\epsilon} - \frac{1}{2}$. Then, Inequality 3 gives

$$\mathbb{P}\left[|\phi(v_j) \cap \phi(v_{j+2r+1})| > k(\frac{1}{2} + \epsilon)\right] \geq 2/3\,,$$

while Inequality 2 gives $\mathbb{P}\left[|\phi(v_j) \cap \phi(v_{j+2r+1})| < k(\frac{1}{2} + \epsilon)\right] \geq 2/3$, a contradiction. Thus, $r \geq \frac{1}{4\epsilon} - \frac{1}{2}$ holds which completes the proof. $\square$

Finally, we provide a reduction from multicoloring the path to interval coloring.

Theorem 5. *Every possibly randomized distributed* $(1+\epsilon)$*-approximation algorithm with error probability* $1/3$ *for coloring interval graphs requires* $\Omega(\frac{1}{\epsilon} + \log^* n)$ *rounds.*

Proof. The $\Omega(\log^* n)$ part of the lower bound follows from Linial's ring coloring lower bound [16] (see also [12]).

To obtain the $\Omega(\frac{1}{\epsilon})$ part of the lower bound, consider an algorithm $\mathbf{A}$ as described in the statement of the theorem. Then, $\mathbf{A}$ can be used to compute a multicoloring of the path $G = (V, E, w)$ with $w(v) = k$, for any integer k, as follows: The nodes $v \in V$ simulate a *fat path* $G' = (V', E')$ where every vertex $v \in V$ is replaced by a clique $C(v)$ of size k and two cliques $C(v)$ and $C(u)$ are adjacent if and only if u and v are adjacent in G. Such a fat path constitutes an interval graph and thus algorithm $\mathbf{A}$ can be simulated to compute a $(1 + \epsilon)$-approximate coloring γ. We then set $\phi(v) = \cup_{v' \in C(v)} \gamma(v')$ which gives a $(1 + \epsilon)$-approximation to the multicoloring problem. The simulation can be implemented with a constant factor blow-up on the number of communication rounds. The lower bound of Theorem 4 thus translates within a constant factor. $\square$

5 Multicoloring Directed Trees

In this section, we work with a directed rooted tree $T = (V, E, w)$ with vertex weights w, where edges are directed from root to leaves. We still assume that nodes that are connected via a directed edge can communicate in both directions. The directions of the edges provides the nodes with additional information about the structure of the underlying tree that help during the multicoloring process, i.e., knowledge of their parent.

In the following, for a vertex $v \in V$ in tree T, we write $\text{parent}_T(v)$ to denote the parent of v in T (the parent of the root node is undefined). We first partition T into clusters, which is described in Subsection 5.1. The multicoloring step is then described in Subsection 5.2.

5.1 Clustering Step

In the first step of our algorithm we compute a partitioning of T into subtrees or clusters, where the set of root nodes of the subtrees forms a (p, q)*-ancestor ruling set*, defined as follows:

Definition 3 ((p, q)-ancestor ruling set). *A (p, q)-ancestor ruling set of a rooted tree $T = (V, E)$ is a subset of vertices $I \subseteq V$ such that:*

1. *If $u, v \in I$ and v is a descendant of u, then the distance between u and v is at least p.*
2. *For every $u \in V \setminus I$, there is an ancestor of u at distance at most q which is also in I.*

Notice that the root is in every ancestor ruling set. We show now how to compute a $(2^t, 2^{t+2} - 1)$-ancestor ruling set in $O(2^t \log^* n)$ rounds. Our algorithm runs in t iterations. In each iteration, the maximal independent set algorithm depicted in Algorithm 2 is used.

Algorithm 2 Subroutine employed for computing a (p, q)-ancestor ruling set

Require: $T(V, E)$ directed tree
  Coloring: Compute 3-coloring on T using the Cole-Vishkin algorithm in $O(\log^* n)$ rounds
  for $i \leftarrow 2, 3$ **do**
    Shift-down: Each node colors itself using the color of its parent. If $i = 2$ then the root selects a color which is different from its old color and different from the first color. If $i = 3$ then the root selects the first color.
  end for
  Independent Set Computation:
  $I \leftarrow$ first color class of coloring
  for $i \leftarrow 2, 3$ **do**
    Every node v of color class i joins I if $I \cup \{v\}$ is an independent set
  end for
  return I

Given a directed tree, we first employ the Cole-Vishkin algorithm and compute a 3-coloring of T. Next, we use two iterations of the *shift-down* technique (see for example [4]), which is a subroutine employed for the task of reducing the number of colors from six to three in the Cole-Vishkin algorithm. The effect of one iteration is that the children of any node have the same color. Executing two iterations of shift-down ensures that for every node, not only its children have the same color but also the children of its children. Finally, we augment the first color class I, which forms an independent set, to a maximal independent set by adding additional vertices of the second and third color classes to I. Notice that I is maximal by construction. Furthermore, set I always contains the root node of the tree, since we assigned the first color to the root in the second shift-down step. The runtime of the algorithm is dominated by the execution of the Cole-Vishkin algorithm, which runs in $\mathrm{O}(\log^* n)$ rounds. Set I has the following useful property:

Lemma 4. *Let $T(V, E)$ be the directed input tree of Algorithm 2, and let I denote the output independent set. Then, for every $v \in V \setminus I$, either the parent or grandparent of v is in I.*

Proof. Notice that the root is in I. The statement is thus true for every node at distance at most 2 from the root. Suppose now that v is at distance at least 3 from the root. Let $p = \mathrm{parent}(v)$ and $q = \mathrm{parent}(p)$. For the sake of a contradiction, suppose that neither p nor q is in I. Then p, q and v were not in the first color class of the computed coloring. Further, v and q received the same color and p the remaining color. Suppose that p received the second color (the case when p received the third color is similar and omitted). Since p is not in I, a vertex incident to p is necessarily in the first color class. Since neither q nor v are in the first color class, one of v's siblings must be in the first color class. This is a contradiction, since due to the shift-down steps employed in the algorithm, the siblings of v have the same color as v. □

Our ruling set algorithm employs t iterations of Algorithm 2. Let $I_1 = V$. For $1 \leq i \leq t$, construct directed tree $T_i(I_i, E_i)$, where two nodes $u, v \in I_i$ are adjacent if among the nodes of I_i, u is the closest ancestor of v in T. Let F_i be the forest obtained from T_i by removing all edges (u, v) with $\mathrm{dist}_T(u, v) \geq 2^i$. We run Algorithm 2 on every tree of F_i simultaneously. Let I_{i+1} be the union of the computed independent sets of all trees of F_i. The algorithm returns I_{t+1}.

We prove next that this algorithm computes a $(2^t, 2^{t+2} - 1)$-ancestor ruling set.

Lemma 5. *For every integer t, a $(2^t, 2^{t+2} - 1)$-ancestor ruling set in a directed tree $T = (V, E)$ can be computed in $\mathrm{O}(2^t \log^* n)$ rounds.*

Proof. We employ t iterations of Algorithm 2. Let $I_1 = V$. For $1 \leq i \leq t$, construct directed tree $T_i(I_i, E_i)$, where two nodes $u, v \in I_i$ are adjacent if among the nodes of I_i, u is the closest ancestor of v in T. Let F_i be the forest obtained from T_i by removing all edges (u, v) with $\text{dist}_T(u, v) \geq 2^i$. We run Algorithm 2 on every tree of F_i simultaneously. Let I_{i+1} be the union of the computed independent sets of all trees of F_i. The algorithm returns I_{t+1}. Notice that $V = I_1 \supseteq I_2 \supseteq \cdots \supseteq I_{t+1}$.

We will next prove the following two claims by induction on i:

1. For every pair of adjacent vertices (u, v) in T_i, it holds that $2^{i-1} \leq \text{dist}_T(u, v) < 2^{i+1}$.
2. For every leaf u in T, there exists a vertex $v \in I_i$ with $\text{dist}_T(u, v) \leq 2^{i+1} - 1$.

For $i = 1$, we consider $T_1 = T$ and both claims are trivially true. Suppose now that both claims are true for index $i \geq 1$. We first prove that Claim 1 also holds for $i + 1$:

Claim 1. Let R_i be the set of root nodes of the trees of F_i. Since the root node of a tree is always included in the output of Algorithm 2, we have $R_i \subseteq I_{i+1}$.

First, consider $v \in R_i$ and let $u = \text{parent}_{T_i}(v)$. Then since v is a root node in F_i, we have $2^i \leq \text{dist}_T(v, u) < 2^{i+1}$. If $u \in I_{i+1}$ then the claim is true. Hence, suppose that $u \notin I_{i+1}$ and let $u' = \text{parent}_{T_i}(u)$. Then $\text{dist}_T(v, u') = \text{dist}_T(v, u) + \text{dist}_T(u, u')$ and hence $2^i + 2^{i-1} \leq \text{dist}_T(v, u') < 2^{i+1} + 2^i$. Thus, if $u' \in I_{i+1}$ then the claim is true. Hence, suppose that u' is not in I_{i+1} and let $u'' = \text{parent}_{T_i}(u')$. By Lemma 4 (applied to node u), u'' is included in I_{i+1}. Then, $\text{dist}_T(v, u'') = \text{dist}_T(v, u') + \text{dist}_T(u', u'')$ and hence $2^i + 2^{i-1} + 2^{i-1} \leq \text{dist}_T(v, u') < 2^{i+1} + 2^i + 2^i$, which implies the claim.

Consider now an arbitrary node $v \in I_{i+1} \setminus R_i$. Since I_{i+1} is an independent set in F_i, the distance between v and any other node $u \in I_{i+1}$ in T_i is at least two, which in turn implies $\text{dist}_T(u, v) \geq 2 \cdot 2^{i-1}$ and proves the lower bound of the claim. To see the upper bound, consider the node $u = \text{parent}_{T_i}(v)$. Then by Lemma 4, either u's parent u' or grandparent u'' is included in I_{i+1}. Since all vertices v, u, u' and u'' are included in the same tree, we have $\text{dist}_T(v, u'') < 3 \cdot 2^i$ and the upper bound thus also holds.

Claim 2. Let v be an any leaf in T. Then, by the induction hypothesis, there exists a vertex $u \in I_i$ with $\text{dist}_T(v, u) \leq 2^{i+1} - 1$. If u is also included in I_{i+1}, then the claim is trivially true. Suppose now that $u \notin I_{i+1}$. Then, by Lemma 4, either u's parent u' or u's grandparent u'' in T_i is included in I_{i+1}. We have $\text{dist}_T(u, u'') = \text{dist}_T(u, u') + \text{dist}_T(u', u'') < 2^i + 2^i = 2^{i+1}$. Hence, $\text{dist}_T(v, u'') = \text{dist}_T(v, u) + \text{dist}_T(u, u'') \leq 2^{i+1} - 1 + 2^{i+1} = 2^{i+2} - 1$, which proves the claim.

We argue now that I_{t+1} is a $(2^t, 2^{t+2}-1)$-ancestor ruling set. Indeed, Claim 1 shows that Item 1 of Definition 3 is fulfilled. To see Item 2 of Definition 3, consider an arbitrary node $v \in V \setminus I_{t+1}$. Notice that the root of T is included in I_{t+1}, since Algorithm 2 always includes the root node in the output independent set Suppose that the subtree below v contains a node of I_{i+1} and let $u \in I_{i+1}$ be the one that is closest to v. Let u' be the closest ancestor of v in I_{t+1}. Since by Claim 1 we have $\mathrm{dist}_T(u, u') < 2^{t+2}$, we obtain $\mathrm{dist}_T(v, u') \leq 2^{t+2} - 1$. Suppose next that the subtree below v does not contain a node of I_{t+1}. By Claim 2, for every leaf in T there is an ancestor which belongs to I_{t+1} at distance at most $2^{t+2} - 1$. Hence, v is either a leaf or located between a leaf and a node of I_{t+1}, and hence, the distance between v and a node of I_{t+1} is at most $2^{t+2} - 1$.

Last, concerning the runtime of the algorithm, the execution of Algorithm 2 on T_i can be simulated with $\mathrm{O}(2^{i+1} \log^* n)$ rounds on T. Since we run t iterations, the runtime of the algorithm is $\mathrm{O}(\sum_{i=1}^{t} 2^{i+1} \log^* n) = \mathrm{O}(2^t \log^* n)$. $\square$

5.2 Multicoloring Step

Let I be a $(2^t, 2^{t+2} - 1)$-ancestor ruling set as output by the algorithm of Lemma 5, for a parameter t whose value we determine later. From I we obtain a clustering of T into subtrees T_v with $v \in I$ as follows: Every vertex $u \in V$ joins the subtree rooted at vertex $v \in I$, where v is the closest ancestor of u in I or $u = v$. The length of every root-to-leaf path in a cluster T_v is then bounded by $2^{t+2} - 1$ and the roots of adjacent clusters are at distances at least 2^t.

Coloring. Our multicoloring process consists of two steps. First, each cluster T_v multicolors itself using at most $\chi^m(G)$ colors. The multicolorings of two adjacent clusters T_u, T_v are independent from each other and may thus be conflicting. Then, in a second step we correct the local multicolorings to obtain a coherent global multicoloring. In this correction step, the root node of a cluster may change all of its colors. Then a color rotation is applied, which gradually changes the initial multicoloring from the root towards the leaves.

Each cluster T_v executes the following two steps:

1. **Initial Multicoloring.** Each cluster T_v first computes a quantity $w^v_{\max} \leq \chi^m(G)$, which is sufficient to color T_v, as follows: Let T'_v be the subtree that contains T_v, the parent of v, and the children of the leaves of T_v. Let $w^v_{\max} \leftarrow \max_{(u,v) \in E(T'_v)} w(u) + w(v)$. Then, nodes of T_v color themselves as follows:

- Nodes u at even distances to v (including v) use colors $\{1, \ldots, w(u)\}$.
- Nodes u at odd distances to v use colors $\{w^v_{\max} - w(u) + 1, \ldots, w^v_{\max}\}$.

2. **Color Rotations.** If there is no coloring conflict between v and parent(v) (which is a node of a different cluster), then stop and return the current multicoloring. Otherwise, let $\delta = \frac{1}{2^{t-1}-2} + \frac{2}{w^v_{\max}}$. Nodes $u \in T_v$ recolor themselves as follows:
   - If $\mathrm{dist}(u, v) = 2l$ (the case $l = 0$ captures vertex v) then use colors starting at color 1 up to color at most $l \cdot \lfloor \delta w^v_{\max} \rfloor$ first. If more colors are needed, i.e., $w(u) > l \cdot \lfloor \epsilon w^v_{\max} \rfloor$, then use additional colors starting at the highest color $w^v_{\max} + \lfloor \delta w^v_{\max} \rfloor$ downwards.
   - If $\mathrm{dist}(u, v) = 2l+1$ we distinguish two cases: If $l \cdot \lfloor \delta w^v_{\max} \rfloor + w(u) \leq w^v_{\max} + \lfloor \delta w^v_{\max} \rfloor$ then use colors from $l \cdot \lfloor \delta w^v_{\max} \rfloor + 1$ upwards. Otherwise, use colors starting at color $w^v_{\max} + \lfloor \delta w^v_{\max} \rfloor$ downwards.

The key observation that proves correctness of the color rotation scheme is that a node u that is far enough away from the root keeps its colors if its initial colors were the first $w(u)$ colors, and shifts its color palette from $\{w^v_{\max} - w(u) + 1, \ldots, w^v_{\max}\}$ upwards by $\lfloor \delta w^v_{\max} \rfloor$ colors otherwise. This property is used to show that there are no conflicts between adjacent trees.

Lemma 6. *The algorithm produces a legal multicoloring.*

Proof. The proof consists of two steps. First, we prove that two adjacent vertices in the same cluster receive different colors. Then, we show that two adjacent vertices of different clusters also receive different colors.

Let $u, u' = \mathrm{parent}(u) \in T_v$, for some $v \in I$. First notice that $w(u) + w(u') \leq w^v_{\max}$, by definition of $w^v_{\max}$. Thus, after the initial coloring step, the color sets of u and u' are disjoint. Suppose that the color rotation takes place since otherwise this case is trivial. We only prove the case $\mathrm{dist}(u, v) = 2l + 1$, for some integer l, the case when $\mathrm{dist}(u, v)$ is even is similar. Suppose first that $l \cdot \lfloor \delta w^v_{\max} \rfloor + w(u) \leq w^v_{\max} + \lfloor \delta w^v_{\max} \rfloor$. Then u uses colors starting at color $l \cdot \lfloor \delta w^v_{\max} \rfloor + 1$ upwards. Node u' uses colors 1 up to $l \cdot \lfloor \delta w^v_{\max} \rfloor$, which are non-conflicting with the colors of u, and potentially additional colors starting with the highest color $w^v_{\max} + \lfloor \delta w^v_{\max} \rfloor$ downwards. Thus, if there was an overlap of colors, then u and u' together would use more than $w^v_{\max}$ colors, a contradiction, since $w(u) + w(u') \leq w^v_{\max}$. In the case $l \cdot \lfloor \delta w^v_{\max} \rfloor + w(u) > w^v_{\max} + \lfloor \delta w^v_{\max} \rfloor$, u uses colors starting at the largest color downwards. Then the inequality $w(u') \leq l \cdot \lfloor \delta w^v_{\max} \rfloor$ necessarily holds, since otherwise $w(u) + w(u') > w^v_{\max}$.

Hence, u' uses colors starting at color 1 upwards and there are no coloring conflicts.

It remains to prove that the colors of a pair of vertices $u, u' = \text{parent}(u)$ with $u \in T_u$ and $u' \in T_v$ with $v \neq u$ are disjoint. Notice that u is the root of T_u. Since I is a $(2^t, 2^{t+2}-1)$-ancestor ruling set, we have $\text{dist}(v, u) \geq 2^t$ and hence $\text{dist}(v, u') \geq 2^t - 1$. Suppose first that there was no coloring conflict between u and u' after the initial multicoloring. At this point, the color set of u is $\{1, \ldots, w(u)\}$ (this color set is also u's final set of colors) and the color set of u' is $\{w^v_{\max} - w(u') + 1, \ldots, w^v_{\max}\}$ (notice that this also implies that $\text{dist}(u', v) = 2l+1$, for some l). If no color rotation takes place in T_v then there is clearly no coloring conflict. Thus, suppose that the color rotation in T_v takes place. Since $\text{dist}(u', v) = 2l + 1$, for some l, and $\text{dist}(u', v) \geq 2^t - 1$, we have $l \geq 2^{t-1} - 1$. Then,

$$\begin{aligned} l \cdot \lfloor \delta w^v_{\max} \rfloor &\geq (2^{t-1} - 1)(\delta w^v_{\max} - 1) \\ &= (2^{t-1} - 1)\left((\frac{1}{2^{t-1} - 2} + \frac{2}{w^v_{\max}}) w^v_{\max} - 1 \right) \\ &> w^v_{\max} + \delta w^v_{\max}, \end{aligned} \tag{4}$$

for $t \geq 3$, and hence after the color rotation step, u' uses colors starting at the largest color $w^v_{\max} + \lfloor \delta w^v_{\max} \rfloor$ downwards. There is hence no coloring conflict between u and u'.

Next, suppose that there is a coloring conflict between u and u' after the initial coloring. Then, after the initial coloring, the color set of u is $\{1, \ldots, w(u)\}$ and the color set of u' is $\{1, \ldots, w(u')\}$ (and hence $\text{dist}(u', v) = 2l$, for some l). Since $\text{dist}(u', v) = 2l \geq 2^t - 1$ is even, we obtain $l \geq 2^t - \frac{1}{2}$. Then, similar to Inequality 4, we obtain $l \cdot \lfloor \delta w^v_{\max} \rfloor > w^v_{\max} + \delta w^v_{\max}$. Hence, after the recoloring step of T_w the colors of u' have not changed. Then, since u recolored itself in the color rotation step with colors starting at the largest color $w^u_{\max} + \lfloor \delta w^u_{\max} \rfloor$ downwards, there is no coloring conflict. □

Theorem 6. *There is a distributed deterministic $(1 + \epsilon)$-approximation multicoloring algorithm for directed trees $G(V, E, w)$ using $O(\frac{1}{\epsilon} \log^* n)$ rounds, for every $\epsilon \geq \frac{4}{\chi^m(G)}$.*

Proof. Correctness of the algorithm is established in Lemma 6. It remains to chose an appropriate value for t employed in our algorithm. By construction, every cluster T_v uses

$$\begin{aligned} w^v_{\max} + \lfloor \delta w^v_{\max} \rfloor &\le w^v_{\max}(1 + \frac{1}{2^{t-1} - 2}) + 2 \\ &\le \chi^m(G)(1 + \frac{1}{2^{t-1} - 2}) + \frac{1}{2}\epsilon\chi^m(G) \end{aligned}$$

colors, where we used the definition of $\delta = \frac{1}{2^{t-1}-2} + \frac{2}{w^v_{\max}}$ and the assumption $2 \ge \frac{1}{2}\epsilon\chi^m(G)$ of the theorem. Hence, setting $t = \ln(\frac{1}{\epsilon} + 2) + 2$, the algorithm is a $(1 + \epsilon)$-approximation.

Notice that the computation of $w^v_{\max}$ can be done in $\mathrm{O}(2^t)$ rounds. Furthermore, every node can learn its distance from the root of its cluster also in $\mathrm{O}(2^t)$ rounds. The runtime is dominated by the computation of the $(2^t, 2^{t+2} - 1)$-ancestor ruling set, which requires $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds. □

6 Conclusion

In this paper, we have presented a distributed $(1 + \epsilon)$-approximation algorithm for coloring interval graphs, which runs in $\mathrm{O}(\frac{1}{\epsilon}\log^* n)$ rounds. It runs in the $\mathcal{LOCAL}$ model and can also be implemented in the $\mathcal{CONGEST}$ model if nodes are aware of their interval representations. We also gave a lower bound of $\Omega(\frac{1}{\epsilon})$. We further demonstrated that the color rotation technique employed in our $\mathcal{CONGEST}$ model algorithm can be useful for other coloring problems as well.

How can we extend our results to more general graph classes such as chordal graphs, which are a superclass of interval graphs? Since chordal graphs contain trees, Linial's lower bound on coloring trees [16] shows that every constant factor approximation algorithm for coloring chordal graphs requires $\Omega(\log n)$ rounds. Can we obtain a $(1 + \epsilon)$-approximation on chordal graphs using $\mathrm{O}(\mathrm{poly}(\frac{1}{\epsilon}) \cdot \mathrm{polylog}\, n)$ rounds in the $\mathcal{LOCAL}$ model? If nodes are aware of their index in a perfect elimination ordering of the chordal graph, can we obtain for example a $\mathrm{O}(\mathrm{poly}(\frac{1}{\epsilon}) \cdot \log^* n)$ rounds algorithm?

References

1. Alon, N., Babai, L., Itai, A.: A fast and simple randomized parallel algorithm for the maximal independent set problem. Journal of Algorithms 7(4), 567 – 583 (1986)
2. Barenboim, L.: Deterministic $(\Delta + 1)$-coloring in sublinear (in Δ) time in static, dynamic, and faulty networks. J. ACM 63(5), 47:1–47:22 (2016), http://dl.acm.org/citation.cfm?id=2979675

3. Barenboim, L., Elkin, M.: Sublogarithmic distributed mis algorithm for sparse graphs using nash-williams decomposition. Distributed Computing 22(5), 363–379 (2010)
4. Barenboim, L., Elkin, M.: Distributed Graph Coloring: Fundamentals and Recent Developments. Synthesis Lectures on Distributed Computing Theory, Morgan & Claypool Publishers (2013), http://dx.doi.org/10.2200/S00520ED1V01Y201307DCT011
5. Barenboim, L., Elkin, M., Gavoille, C.: A fast network-decomposition algorithm and its applications to constant-time distributed computation. In: Post-Proceedings of the 22Nd International Colloquium on Structural Information and Communication Complexity - Volume 9439. pp. 209–223. SIROCCO 2015, Springer-Verlag New York, Inc., New York, NY, USA (2015), http://dx.doi.org/10.1007/978-3-319-25258-2_15
6. Barenboim, L., Elkin, M., Pettie, S., Schneider, J.: The locality of distributed symmetry breaking. J. ACM 63(3), 20:1–20:45 (Jun 2016), http://doi.acm.org/10.1145/2903137
7. Bollobás, B.: Chromatic number, girth and maximal degree. Discrete Mathematics 24(3), 311 – 314 (1978)
8. Chang, Y.J., Kopelowitz, T., Pettie, S.: An exponential separation between randomized and deterministic complexity in the local model. In: Proceedings 57th IEEE Symposium on Foundations of Computer Science (FOCS). pp. 615–624 (2016)
9. Cole, R., Vishkin, U.: Deterministic coin tossing with applications to optimal parallel list ranking. Inf. Control 70(1), 32–53 (Jul 1986), http://dx.doi.org/10.1016/S0019-9958(86)80023-7
10. Fraigniaud, P., Heinrich, M., Kosowski, A.: Local conflict coloring. In: IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA. pp. 625–634 (2016), http://dx.doi.org/10.1109/FOCS.2016.73
11. Goldberg, A.V., Plotkin, S.A., Shannon, G.E.: Parallel symmetry-breaking in sparse graphs. SIAM J. Discrete Math. 1(4), 434–446 (1988), http://dx.doi.org/10.1137/0401044
12. Halldórsson, M.M., Konrad, C.: Distributed Algorithms for Coloring Interval Graphs, pp. 454–468. Springer (2014), http://dx.doi.org/10.1007/978-3-662-45174-8_31
13. Harris, D.G., Schneider, J., Su, H.H.: Distributed (∆+1)-coloring in sublogarithmic rounds. In: Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing. pp. 465–478. STOC '16, ACM, New York, NY, USA (2016), http://doi.acm.org/10.1145/2897518.2897533
14. Karp, R.M.: Reducibility Among Combinatorial Problems. In: Miller, R.E., Thatcher, J.W. (eds.) Complexity of Computer Computations, pp. 85–103. Plenum Press (1972)
15. Kuhn, F.: Local multicoloring algorithms: Computing a nearly-optimal TDMA schedule in constant time. In: 26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings. pp. 613–624 (2009), http://dx.doi.org/10.4230/LIPIcs.STACS.2009.1852
16. Linial, N.: Locality in distributed graph algorithms. SIAM J. Comput. 21(1), 193–201 (Feb 1992), http://dx.doi.org/10.1137/0221015

17. Luby, M.: A simple parallel algorithm for the maximal independent set problem. In: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing. pp. 1–10. STOC '85, ACM, New York, NY, USA (1985), http://doi.acm.org/10.1145/22145.22146
18. Schneider, J., Elkin, M., Wattenhofer, R.: Symmetry breaking depending on the chromatic number or the neighborhood growth. Theor. Comput. Sci. 509, 40–50 (Oct 2013), http://dx.doi.org/10.1016/j.tcs.2012.09.004
19. Schneider, J., Wattenhofer, R.: An Optimal Maximal Independent Set Algorithm for Bounded-Independence Graphs. Distributed Computing 22 (March 2010)
20. Schneider, J., Wattenhofer, R.: A new technique for distributed symmetry breaking. In: Proceedings of the 29th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing. pp. 257–266. PODC '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1835698.1835760
21. Smith, D.A.: The First-fit Algorithm Uses Many Colors on Some Interval Graphs. Ph.D. thesis, Arizona State University, Tempe, AZ, USA (2010), aAI3428197
22. Valencia-Pabon, M.: Revisiting Tucker's algorithm to color circular arc graphs. SIAM J. Comput. 32(4), 1067–1072 (2003)
23. Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. Theory of Computing 3(1), 103–128 (2007)