Distributed Algorithms for Coloring Interval Graphs[⋆]

Magnús M. Halldórsson and Christian Konrad

ICE-TCS, School of Computer Science, Reykjavik University, Iceland {mmh,christiank}@ru.is

Abstract. We explore the question how well we can color graphs in distributed models, especially in graph classes for which $\Delta+1$-colorings provide no approximation guarantees. We particularly focus on interval graphs.

In the $\mathcal{LOCAL}$ model, we give an algorithm that computes a constant factor approximation to the coloring problem on interval graphs in $\mathrm{O}(\log^* n)$ rounds, which is best possible. The result holds also for the $\mathcal{CONGEST}$ model when the representation of the nodes as intervals is given.

We then consider restricted beep models, where communication is restricted to the aggregate acknowledgment of whether a node's attempted coloring succeeds. We apply an algorithm designed for the SINR model and give a simplified proof of a $O(\log n)$-approximation. We show a nearly matching $\Omega(\log n / \log\log n)$-approximation lower bound in that model.

1 Introduction

In this paper, we study distributed algorithms for vertex coloring, especially on interval graphs. Given a set of intervals on the line $V = \{I_1, \ldots, I_n\}$ with $I_j = (a_j, b_j)$ and a_j, b_j being real numbers such that $a_j < b_j$, an interval graph G is obtained from V as follows: The vertex set of G are the intervals V, and two vertices $I_j, I_k \in V$ are adjacent if and only if I_j and I_k intersect. Interval graphs have a multitude of applications, appear naturally in scheduling problems, and can for instance be seen as one-dimensional projections of disk graphs that are often used for modeling wireless networks [1–3].

Graph Coloring. For an integer s, an s-coloring of a graph $G = (V, E)$ is an assignment $\gamma : V \rightarrow \{1, \ldots, s\}$ of colors to the vertices of a graph such that any two adjacent vertices have different colors. The chromatic number $\chi(G)$ of a graph G is the minimum number of colors that is needed to color G. It is well-known that determining $\chi(G)$ is NP-complete [4] in general graphs and it is even hard to approximate it within a $\Theta(n^{1-\epsilon})$ factor, for any $\epsilon > 0$ [5]. Sequentially, it is easy to find a coloring that uses at most $\Delta + 1$ colors where Δ is the maximal degree of a graph: Traverse the vertices of G in any order and assign the smallest possible color to the current vertex. Since there are graphs for which $\Delta = \Theta(n)$ and $\chi(G) = \Theta(1)$ (for instance a star graph), such a coloring may be as bad as a $\Theta(n)$-approximation. An optimal coloring of an interval graph can be found by traversing the intervals with increasing left interval boundary and coloring them with the smallest possible color. Even if the intervals are traversed in arbitrary order, we obtain a *canonical* coloring, where a node colored $s(v)$ is adjacent to nodes colored $1, 2, \ldots, s(v) - 1$. It is known that such colorings of interval graphs yield a C-approximation, where $5 \leq C \leq 8$ [6].

Distributed Graph Coloring. Graph coloring has been extensively studied in the distributed setting (see [7–11] to name a few). In the distributed computational model, we assume a network of computational units modeled by a graph $G = (V, E)$ which is also the input graph of the problem. The computational units constitute the vertices of G, and two computational units can exchange messages if and only if an edge connecting them is included in E. Then, the runtime of a distributed algorithm is the number of communication rounds required to complete the algorithm. We assume that each vertex has a unique ID. In the $\mathcal{LOCAL}$ model, in each round, messages of unbounded size may be exchanged. In the $\mathcal{CONGEST}$ model, all message are of size at most $\mathrm{O}(\log n)$ (n is the number of computational units). Due to the hardness of the graph coloring problem, the objective of most works on this topic in the distributed model is to find a coloring that uses $\Delta + 1$ or $\mathrm{O}(\Delta)$ colors on general graphs. A $(\Delta + 1)$-coloring can be found by a distributed randomized algorithm in $\mathrm{O}(\log n)$ communication rounds by a reduction to the maximal independent set

[⋆] Both authors are supported by Icelandic Research Fund grant-of-excellence no. 120032011.

problem that was first mentioned in [7] (a maximal independent set can be found in $\mathrm{O}(\log n)$ time by Luby's algorithm [12]). An $\mathrm{O}(\Delta)$-coloring can be computed by a distributed randomized algorithm in $2^{\mathrm{O}(\sqrt{\log \log n})}$ rounds [13]. The best deterministic distributed coloring algorithm that finds an $\mathrm{O}(\Delta)$-coloring performs $\mathrm{O}(\Delta^{\epsilon} \log n)$ rounds, for any $\epsilon > 0$ [10]. Only few works consider more specialized graph classes for which better colorings can be obtained, and we reuse some of those works in this paper. In $\mathrm{O}(\log^* n)$ rounds, Cole and Vishkin showed that a 3-coloring of a ring can be computed [14]. In [15], this technique has been extended to coloring *bounded-independence graphs* with $\Delta + 1$ colors (see Definition 1). Linial showed in [7] that coloring a ring with 3 colors requires $\Omega(\log^* n)$ rounds which renders the previous algorithms optimal.

Distributed Algorithms for Coloring Interval Graph. Our interest in coloring interval graphs in a distributed fashion stems from the following observation. As previously mentioned, most distributed coloring algorithms compute $(\Delta + 1)$-colorings which may be as bad as $\Theta(n)$ approximations. We are therefore interested in graph classes for which better approximation ratios can be obtained. Surprisingly, for interval graphs, we identify that in the $\mathcal{LOCAL}$ model, a constant factor approximation with runtime $\mathrm{O}(\log^* n)$ can be obtained (**Theorem 5**). To this end, we identify that the subgraph $G_P \subseteq G$ of proper intervals (roughly those intervals that are not properly contained in other intervals) has a maximal degree of $\mathrm{O}(\chi(G))$. Furthermore, G_P is of *bounded-independence* (**Theorem 1**) which is defined as follows:

Definition 1. *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 $f(r), \forall r \geq 0$. The r-neighborhood of a node v is the set of nodes at distance at most r from v (excluding v).*

Schneider and Wattenhofer present in [15] a distributed maximal independent set algorithm for bounded-independence graphs that runs in time $\mathrm{O}(\log^* n)$. Using this algorithm, we compute an independent set in the subgraph G_P in $\mathrm{O}(\log^* n)$ time, and we show how to extend it to a dominating set that dominates the whole graph G. Then, we use this dominating set to coordinate the coloring of all vertices. By construction, this coloring is a canonical one, and since every canonical coloring in an interval graph is at least an 8-approximation, the result follows.

Furthermore, we show that computing an $\mathrm{O}(\log^* n)$-approximation to the coloring problem in interval graphs requires $\Omega(\log^* n)$ time by a reduction to a result of Linial [7]. Linial showed that obtaining a 3-coloring on a ring requires $\Omega(\log^* n)$ rounds. We show that any algorithm that colors interval graphs with fewer rounds would imply a faster 3-coloring algorithm of the ring contradicting the previous lower bound. This renders our algorithm tight. Moreover, we observe that if nodes are aware of their interval boundaries then the previous algorithm can even be implemented in the $\mathcal{CONGEST}$ where all messages are of size at most $\mathrm{O}(\log n)$.

A Simple Coloring Scheme. We also consider a particular class of simple distributed coloring algorithms that have been successfully applied in the past to solve the coloring problem in the SINR (Signal to Interference plus Noise Ratio) model for wireless communication [16–18]. From a graph theoretical point of view, in the SINR model, a complete directed edge-weighted graph with vertex set L is given, where each vertex $l \in L$ represents a transmission link consisting of a sender and a receiver. The weights of the edges between transmission links determine the amount of relative interference that a transmitting sender has on the receiver of another link. The notion of independent sets and colorings are adapted as follows: A subset of nodes $L' \subseteq L$ is an independent set if the in-degree of every node $l \in L'$ from other nodes of L' is at most 1. An independent set corresponds here to a set of links that can successfully transmit simultaneously. Then an s-coloring is a decomposition of the vertex set into s independent sets. An s-coloring corresponds here to a schedule that permits the successful transmission of all links in s rounds. The algorithms for coloring SINR-instances of [16–18] are round-based, and they follow the scheme of Algorithm 1 (Algorithm 1 is stated for unweighted graphs $G = (V, E)$ which is the form we need in this paper).

The scheme of Algorithm 1 computes a coloring $\gamma : V \to \mathbb{N}$. In each round i, a probability p_i is determined in Line 4. Different Algorithms that follow this scheme such as the algorithms of [16–18] compute different sequences $(p_i)_i$. The sequence of probabilities p_i determine the efficiency of the scheme, and different graphs classes may require different sequences. Then, all not-yet colored nodes v pre-select themselves as candidates to be colored with probability p_i in Line 6. We assume that we have a function coin: $[0, 1] \to \{\mathit{true}, \mathit{false}\}$ to

Algorithm 1 Simple coloring scheme

```
Require: $G = (V, E)$ {Input graph}
 1: $\gamma(v) \leftarrow \bot$ for all $v \in V$ {The coloring to be computed}
 2: $i \leftarrow 1$ {Current color}
 3: while $\exists v \in V$ with $\gamma(v) = \bot$ do
 4:     Determine $p_i$ {Algorithms following this scheme have to implement this line}
 5:     for all $v \in V$ with $\gamma(v) = \bot$ do
 6:         $T_v \leftarrow$coin($p_i$) {Pre-selection step: If coin($p_i$) = *true* then $v$ is a candidate to be colored}
 7:     end for
 8:     for all $v \in V$ with $\gamma(v) = \bot$ and $T_v = \mathit{true}$ do
 9:         if $\bigvee_{u \in \Gamma_V(v) \text{ with } \gamma(u)=\bot} T_u = \mathit{false}$ then {Check whether a neighbor of $v$ has been pre-selected}
10:             $\gamma(v) \leftarrow i$ {Color node $v$}
11:         end if
12:     end for
13:     $i \leftarrow i + 1$
14: end while
15: return $\gamma$
```

our disposal such that coin(p) returns *true* with probability p, otherwise *false*. Next, in Line 10, pre-selected nodes color themselves with color i if none of its not yet colored neighbors pre-selected themselves.

Algorithms that follow the scheme of Algorithm 1 are simple and easy to implement. They do not require a complicated mechanism for breaking ties as a pre-selected node is only colored if none of its neighbors is pre-selected, or, in other words, a node only has to learn the logical OR of the bits of its neighbors indicating whether a neighbor is pre-selected. As we will discuss in Section 5, exchanging this type of information does not put high demands on the distributed model in which this algorithm is implemented. This makes the algorithm a good candidate for being implemented in various models. We will show that an implementation of this scheme is possible in the very restrictive discrete beeping model [19] in which, among other things, nodes cannot distinguish between different neighboring nodes, and the number of neighbors of a node is unknown to the node itself. Algorithms of type Algorithm 1 are essentially the only type of coloring algorithms that can be implemented in this model.

This scheme of algorithms is referred to as *acknowledgement-only* (ack-only) algorithms [16–18] in the SINR community. As previously mentioned, in the SINR model, a set of communication links each consisting of a sender and a receiver is considered. Links are not aware of their neighborhood. In each round, a sender may either attempt to transmit (pre-select itself) and hope for a successful transmission, or it may remain silent and wait. Ack-only algorithms assume that senders receive an acknowledgment of whether their transmission was successful or whether it failed (check whether there are neighbors that pre-selected themselves). Successful links then remain silent until all links successfully transmitted (once a node is colored it does not attempt to color itself again). Since there is no information exchange between communication links, in each round, senders essentially can only flip a coin and transmit with a certain probability. Note that this situation is modeled by the scheme of Algorithm 1. While in the scheme of Algorithm 1 the identification of whether a communication attempt was success is checked in Line 9, this is achieved in the SINR model with explicit acknowledgments in a separate round that succeed with constant probability.

It is known that:

Theorem 1 ([20]). *There is an algorithm that follows the scheme of Algorithm 1 and colors a graph with* $\mathrm{O}(d\chi(G)\log n)$ *colors w.h.p. where* d *is the inductive independence number of a graph.*

Inductive independence [21] is defined as follows:

Definition 2. *A graph* $G = (V, E)$ *is* inductive d-independent *if there exists an ordering* π *of the vertices* V *such that for every independent set* $I \subseteq V$ *and every vertex* $v \in V$*:*

$$|\{u \in \Gamma_G(v) \text{ with } \pi(u) > \pi(v)\} \cap I| \leq d.$$

The inductive independence number of G is the smallest d such that G is inductive d-independent.

Many interesting graph classes have bounded inductive independence, e.g., disc graphs are inductive 5-independent, planar graphs are inductive 3-independent, claw-free graphs are inductive 2-independent, and most importantly, chordal graphs (a superclass of interval graphs) are inductive 1-independent. It is well-known that chordal graphs are exactly those graphs that admit a *perfect elimination ordering*: A perfect elimination ordering in a graph $G = (V, E)$ is an ordering π of the vertices V such that, for each $v \in V$, $v \cup \{u \in \mathit{\Gamma}_G(v) \text{ with } \pi(u) > \pi(v)\}$ forms a clique. Note that this is equivalent to the definition of inductive 1-independence.

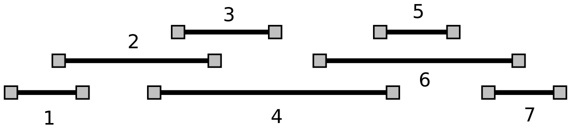

Fig. 1. Example of a perfect elimination ordering of an interval graph. For each interval with index i, the size of an independent set among its neighbors with larger index is at most 1.

In the context of SINR coloring, it is shown in [20] that many important SINR instances are inductive O(1)-independent, and by Lemma 1, an $\mathrm{O}(\log n)$-approximation algorithm to the coloring problem in the SINR model is obtained. It is an open question whether there is an algorithm that follows the scheme of Algorithm 1 and computes an O(1)-approximation (in fact, for many instances no algorithm at all is known that computes an O(1)-approximation). In [18], an instance is provided that can be colored with 2 colors, while any such algorithm requires $\mathit{\Omega}(\log n)$ rounds. However, no hard instances are known with larger chromatic number.

In this paper, we settle this question for interval graphs up to a $\log \log n$ factor. As interval graphs are inductive 1-independent, we immediately obtain an $\mathrm{O}(\log n)$-approximation by Theorem 1. We will show that every algorithm that follows the scheme of Algorithm 1 requires $\mathit{\Omega}\left(\frac{\log n}{\log \log n}\chi(G)\right)$ colors (**Theorem 7**), matching the upper bound up to a $\log \log n$ factor.

Furthermore, we provide an alternative proof of Theorem 1. We essentially identify that there is an algorithm following the scheme of Algorithm 1 that colors graphs G that have the property that any induced subgraph on α vertices has at most αk edges using $\mathrm{O}(k \log n)$ colors (**Theorem 6**). We observe that inductive d-independent graphs have at most $d\chi(G)n$ edges, which allows us to conclude the statement of Theorem 1. Alternatively, our theorem can also be applied to *k-degenerate* graphs. A graph is k-degenerate if every induced subgraph has a node of degree at most k. Clearly, such a graph has at most kn edges.

While the lower bound does not carry over to the geometric SINR model, it shows that in the abstract SINR model, improved results for scheduling in terms of inductive independence are not possible by these types of algorithms.

Outline. In Section 2, we present necessary definitions and notations, and we prove a property about interval graphs that is required in the subsequent section. In Section 3, we present our upper and lower bound for a constant factor approximation in the $\mathcal{LOCAL}$ model and its adaption to the $\mathcal{CONGEST}$ model. Then, in the following sections, we consider the previously mentioned class of simple coloring algorithms. We revisit the result that an inductive d-independent graph can be colored by an algorithm of the previous scheme with $\mathrm{O}(d\chi(G) \log n)$ colors in Section 4. Then, In Section 5, we underline the simplicity of algorithms that follows the scheme of Algorithm 1, and we show that they can be implemented in the very restrictive discrete beeping model. Finally, we prove in Section 6 that any algorithm of the previous scheme requires $\mathit{\Omega}(\frac{\log n}{\log \log n}\chi(G))$ colors on interval graphs.

Furthermore, we note that due to space restrictions, some proofs are omitted but can be found in the full version of this article.

2 Preliminaries

Definitions. An *independent set* in a graph $G = (V, E)$ is a subset $I \subseteq V$ of vertices such that for every pair of vertices $v_1, v_2 \in I : (v_1, v_2) \notin E$. An independent set I is *maximal* if $I \cup \{v\}$ is not an independent set for all $v \in V \setminus I$. A *dominating set* in a graph $G = (V, E)$ is a subset $D \subseteq V$ such that for any vertex $v \in V \setminus D$, v is adjacent to at least one vertex $u \in D$. Any maximal independent set is a dominating set, however, the converse is not true. For an integer k, a *distance-k-coloring* of a graph $G = (V, E)$ is an assignment $\gamma : V \to \{1, \ldots, s\}$ of colors to the vertices of a graph such that any two vertices at distance at most k have different colors.

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 $a_j < b_j$. 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 overlap. Let $m = |E|$. We assume that all a_i, b_i are distinct. For simplicity, we will assume that the input interval graphs are connected. The *neighborhood* of a vertex v in graph G is denoted by $\Gamma_G(v)$, and we define $\Gamma_G\,[v] = \Gamma_G(v) \cup \{v\}$. For a subset $V' \subseteq V$, we may 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, and we denote it by $\Gamma^k_G(v)$. Then $\Gamma^1_G(v) = \Gamma_G(v)$. 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 which is induced by the nodes V', that is, $\deg_{V'}(v) := \deg_{G|_{V'}}(v)$.

We say that an interval v is *proper* if there is no other interval u such that $\Gamma_G\,[v] \subsetneq \Gamma_G\,[u]$. For an interval graph $G = (V, E)$, we denote by $G_P = (V_P, E|_{V_P})$ the subgraph of G that is induced by the proper intervals of G. It is easy to see that the subgraph G_P of a connected interval graph is connected, too. Then the following degree bound holds:

Lemma 1. *For all $v \in V_P$: $\deg_{G_P}(v) \leq 2\chi(G_P) - 2$.*

In order to prove Lemma 1, we use the following notations. Given the interval representation of an interval graph, we denote the intervals that intersect with an interval v on its left (resp. right) boundary by $\Gamma^L_G(v)$ (resp. $\Gamma^R_G(v)$). The set of intervals that are contained in an interval v is denoted by $\Gamma^C_G(v)$. Note that $\Gamma_G(v) = \Gamma^L_G(v) \cup \Gamma^C_G(v) \cup \Gamma^R_G(v)$.

Proof. Let $v \in V_P$. First, notice that for all $v' \in \Gamma^C_{G_P}(v) : \Gamma_{G_P}(v') = \Gamma_{G_P}(v)$ since otherwise v' would not be in V_P. Therefore, all intervals $v'' \in \Gamma^L_{G_P}(v) \cup \Gamma^R_{G_P}(v)$ intersect with all intervals in $\Gamma^C_{G_P}(v)$. Then, $\Gamma^C_{G_P}(v) \cup \Gamma^L_{G_P}(v)$ forms a clique as well as $\Gamma^C_{G_P}(v) \cup \Gamma^R_{G_P}(v)$. Since any clique in a graph G is at most of size $\chi(G)$, we obtain

$$|\Gamma_{G_P}(v)| \leq |\Gamma^C_{G_P}(v) \cup \Gamma^L_{G_P}(v)| + |\Gamma^C_{G_P}(v) \cup \Gamma^L_{G_P}(v)| - 2 \leq 2 \cdot (\chi(G_P) - 1).$$

□

Distributed Algorithms. In the following, we will reuse existing distributed algorithms. The deterministic distributed algorithm of Wattenhofer and Schneider [15] colors a bounded-independence graph using $\Delta + 1$ colors in $\mathrm{O}(\log^* n)$ time, and we will denote this algorithm by ColBI (BI stands for bounded independence). This algorithm can be implemented such that it returns a *canonical coloring*, i.e., a coloring such that no node could change its color to a smaller one. In the same work, Wattenhofer and Schneider show that in a bounded-independence graph, a maximal independent set can be deterministically computed in $\mathrm{O}(\log^* n)$ time, and we denote this algorithm by MisBI. Both, ColBI and MisBI, can be implemented in the $\mathcal{CONGEST}$ model.

3 O(1)-approximation for Coloring Interval Graphs in the $\mathcal{LOCAL}$ Model

In this section, we show that an interval graph G can be colored in $\mathrm{O}(\log^* n)$ time with $\mathrm{O}(\chi(G))$ colors in the $\mathcal{LOCAL}$ model. Our algorithm makes use of the distributed algorithms ColBI and MisBI for computing a coloring and an independent set in bounded-independence graphs. We run these algorithms on the subgraph

G_P of proper intervals. Unit disc graphs are of bounded independence [15]. Since the class of proper interval graphs is equivalent to the class of unit interval graphs, and unit interval graphs are a subclass of unit disc graphs, the following fact follows immediately:

Fact 1 *Proper interval graphs are of bounded independence.*

We present our algorithm in Subsection 3.1, its analysis in Subsection 3.2, and we discuss an implementation of the algorithm in the $\mathcal{CONGEST}$ model in Subsection 3.3.

3.1 Algorithm

1. **Identify the subgraph G_P of proper intervals:** Each node v determines if $v \in G_P$ by checking if there is a neighbor $u \in \Gamma_G(v)$ such that $\Gamma_G[u] \supsetneq \Gamma_G[v]$. If no such neighbor exists then v is in G_P. This involves one round of communication where each node sends the list of its neighbors to all its neighbors.
2. **Compute a maximal independent set J of G_P:** Using MisBI, we compute a maximal independent set J of the graph G_P in $\mathrm{O}(\log^* n)$ rounds. J is needed for the computation of a dominating set in the next step.
3. **Compute a dominating set $N \cup J$:** Algorithm 2 computes a set N such that $N \cup J$ is a dominating set of the graph G. Ties are broken arbitrarily. In step one, every node communicated already its list of neighbors to its neighbors, and hence no further communication is required.

Algorithm 2 Computation of a dominating set

```
for all $v \in J$ do
    $u_1 \leftarrow \arg\max_{u \in \Gamma_{G_P}(v)} |\Gamma_G(u) \setminus \Gamma_G(v)|$
    $u_2 \leftarrow \arg\max_{u \in \Gamma_{G_P}(v)} |\Gamma_G(u) \setminus (\Gamma_G(v) \cup \Gamma_G(u_1))|$
    $N \leftarrow N \cup \{u_1, u_2\}$
end for
```

4. **Find a distance-3 coloring of $G|_{N \cup J}$ and obtain color classes $(I_i)_{i \geq 1}$:** We argue in the analysis that the maximal degree in the vertex induced graph $G|_N$ is 4, and hence the maximal degree in $G|_{N \cup J}$ is 5. Therefore, the size of the 4-neighborhood of every node is bounded by some constant C. We build the graph H on vertex set $N \cup J$ where nodes are adjacent if they are at distance at most 3 in G_P. This involves two additional rounds of communication to establish knowledge about the 3-neighborhood of each node. We run ColBI to color H in time $\mathrm{O}(\log^* n)$ and we obtain a constant number of color classes $(I_i)_{i \geq 1}$. This coloring is a distance-3 coloring of $G|_{N \cup J}$.
5. **Coloring.** After each of the following iterations, in one round of communication, each node that has received a color notifies its neighbors about its own color. This guarantees that a not-yet colored node always knows the palette of still available colors that it may be colored with.
   Iterate over the sets $(I_i)_{i \geq 1}$ and do the following:

   Every node $u' \in I_i$ coordinates the coloring of not-yet colored nodes $u \in \Gamma_G[u']$ as follows: Nodes u send the palette of possible colors with which they may be colored to u'. The node u' is unique for u: As I_i is a color class of a distance-3 coloring, every other node $u'' \in I_i \setminus \{u'\}$ is at distance at least 2 from u. Then, u' determines a canonical coloring of all nodes u respecting the color restrictions of the nodes, and notifies the nodes u about their color.

3.2 Analysis

Lemma 2. *The following properties hold:*

1. *$J \cup N$ is a dominating set in G.*

2. The maximal degree of a node in the graph $G|_N$ is 4.

Proof. First, we prove Item 1. Let $v \in V \setminus V_P$. Suppose that v is not adjacent to any node in J since otherwise the statement is trivially true. Consider the interval representation of the graph. Since $v \notin V_P$, there is a node u in V_P that is adjacent to v such that $\Gamma_G[u] \supset \Gamma_G[v]$. Since J is a maximal independent set in G_P, J is also a dominating set in G_P. Therefore, there is at least one node $w \in J$ that is adjacent to v. Consider Algorithm 2 for node w and consider the interval representation of the vertices. Algorithm 2 basically selects an interval $i_1 \in \Gamma^L_{G_P}(w)$ that reaches furthest out to the left, and an interval $i_2 \in \Gamma^R_{G_P}(w)$ that reaches furthest out to the right. Therefore, for either $i = i_1$ or $i = i_2$, we have $\Gamma_G(i) \setminus \Gamma_G(w) \supseteq \Gamma_G(u) \setminus \Gamma_G(w)$, and hence node v is covered by either i_1 or i_2 which proves Item 1.

Second, we prove Item 2. Consider the interval representation of J. Since J is an independent set, J is a collection of non-overlapping intervals. Consider three adjacent intervals $i_1, i_2, i_3 \in J$ such that i_2 lies in between i_1 and i_3. For any i_j $(j = 1, 2, 3)$, Algorithm 2 selects the intervals that reach out furthest to the left of right. Denote by i^1_j and i^2_j the two selected intervals for i_j. Note that each interval i^k_j $(k = 1, 2)$ may overlap with at most one other interval from J different from i_j. i^k_j cannot completely include such an interval since $J \subseteq G_P$ and G_P is the subgraph of proper intervals. Therefore, at most 4 selected intervals may overlap: The two selected intervals from i_2, one interval from i_1 and one interval from i_3. The maximal degree is hence 4. □

Lemma 3. *For every $u \in V$ there is at least one set I_i s.t. $|\Gamma_G[u] \cap I_i| = |\Gamma_G[\Gamma_G[u]] \cap I_i| = 1$. That is, each node is dominated by a node in some I_i but has then no other node in I_i in its 2-neighborhood.*

Proof. Let u be a vertex in V. Since $J \cup N = \cup_i I_i$ is a dominating set, u is adjacent to at least one node u' of $J \cup N$. So, $|\Gamma_G[\Gamma_G[u]] \cap I_i| \geq |\Gamma_G[u] \cap I_i| \geq 1$. Let i be the index with $u' \in I_i$.

Suppose that $|\Gamma_G[\Gamma_G[u]] \cap I_i| \geq 2$. Then, there is a vertex v in V that is adjacent to both u and u'', for some u'' in $I_i \setminus \{u'\}$. There is a vertex $\hat{u}$ $(\hat{v})$ in G_P (not necessarily distinct from u (v)) corresponding to a proper interval that contains the interval of u $(\hat{v})$, and the neighbors of u (v) are also neighbors of $\hat{u}$ $(\hat{v})$, respectively. Thus, $u', \hat{u}, \hat{v}, u''$ is a path of length 3 in G_P, contradicting the distance-3 coloring property. □

Lemma 3 shows that every node will be correctly colored in Step 5 of the algorithm. $|\Gamma_G[u] \cap I_i| = 1$ shows that all nodes will be considered in the coloring step of the algorithm, and $|\Gamma_G[\Gamma_G[u]] \cap I_i| = 1$ guarantees that the computed colorings of the different nodes of I_i do not interfere with each other. We conclude with the main theorem:

Theorem 2. *In the $\mathcal{LOCAL}$ model, there is a deterministic $\mathrm{O}(1)$-approximation algorithm that computes a canonical coloring of an interval graph and runs in time $\mathrm{O}(\log^* n)$.*

Proof. Concerning correctness of the algorithm, we showed in Lemma 3 that every node $v \in G$ will be colored in Step 5. By construction, the algorithm computes a canonical coloring, i.e., it always assigns the smallest color possible to a node. Therefore, the total amount of required colors can be bounded by the fact that any canonical coloring of an interval graph uses at most $8 \cdot \chi(G)$ colors [6]. The runtime of the algorithm is $\mathrm{O}(\log^* n)$ since we essentially run a constant number of times the algorithms MisBI and ColBI whose runtimes are $\mathrm{O}(\log^* n)$. □

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

Suppose that every node $v_i \in V$ is aware of its interval representation and knows its interval boundaries a_i, b_i. We assume that the numbers a_i, b_i require space $\mathrm{O}(\log n)$ to be written down. Then the previous algorithm can be implemented in the $\mathcal{CONGEST}$ model: Concerning Step 1, exchanging interval boundaries and the number of neighbors is enough to determine whether a node $v \in V$ is also in V_P. Step 2 remains unchanged. Since each node $v \in V_P$ knows the interval boundaries of its neighbors, Step 3 is simplified and v simply selects incident intervals that reach out furthest to the left and to the right. Step 4 remains unchanged. Since the maximal degree in H is bounded by a constant, all messages sent in order to compute the 3-neighborhood of every node are still of size $\mathrm{O}(\log n)$. Concerning Step 5, note that it is impossible that every node u sends its palette of still available colors to the coordinator u' with a message of size $\mathrm{O}(\log n)$. We therefore give up on obtaining a canonical coloring, and, instead, for each coloring round we use a set of new colors (for instance round i uses the colors $\{(i-1)n+1, in\}$. Since, however, each coloring round uses $\mathrm{O}(\chi(G))$ colors and there are only a constant number of sets I_i, we still obtain a constant factor approximation.

Theorem 3. *There is a deterministic* $\mathrm{O}(1)$*-approximation algorithm that computes a coloring of an interval graph in the* $\mathcal{CONGEST}$ *model and runs in time* $\mathrm{O}(\log^* n)$ *if each node knows its interval boundaries.*

3.4 Lower Bound For Coloring Interval Graphs In The $\mathcal{LOCAL}$ Model

Linial's lower bound shows that any distributed algorithm for coloring the n-cycle with three colors requires time $\Omega(\log^* n)$. This lower bound even holds if each node is aware that the graph to be colored is an n-cycle, and the n nodes have distinct labels in $\{1,\dots,n\}$. The difficulty stems from the fact that the n nodes may have arbitrary labels from the set $\{1,\dots,n\}$. For our reduction in the proof of Theorem 5, it is important that there is one arbitrary node that can distinguish itself from the other ones. We assume that nodes have labels from the set $\{1,\dots,n\}$, and we select the node with label 1 as a distinct one. Linial's lower bound still holds in that case.

Theorem 4. *[7] Every possibly randomized distributed algorithm that colors the n-cycle with at most three colors requires time $\Omega(\log^* n)$.*

We require a well-known color reduction in our proof. The following lemma is well-known and we prove it here for completeness.

Lemma 4. *Let γ denote a coloring of the graph $G=(V,E)$ that uses $\alpha \geq \Delta+2$ colors. Then, in one round, a coloring can be computed that uses $\alpha-1$ colors.*

Proof. In one round, every node communicates its current color to its neighbors. The set of nodes with color α forms an independent set. This allows every vertex $v \in V$ with color α to picks a new color from $\{1,\dots,\Delta+1\} \setminus \gamma(\Gamma_G(v))$. Since a node has at most Δ neighbors, there is always an available color. □

Theorem 5. *Every possibly randomized distributed algorithm that colors an interval graph $G=(V,E)$ with $|V|=n$ using $o(\log^*(n)\chi(G))$ colors requires time $\Omega(\log^* n)$.*

Proof. Suppose that there is a distributed algorithm A which colors an interval graph using $o(\log^*(n)\chi(G))$ colors in $o(\log^* n)$ time. We will show that this leads to a contradiction to Theorem 4 as with the help of such an algorithm, we can color the n-cycle with 3 colors in $o(\log^* n)$ time as follows:

Given an n-cycle, the vertex with label 1 removes one of its edges which transforms the n-cycle to a line of length n. Denote by s the vertex with label 1 and by t the other endpoint of the line. Note that the chromatic number of a line is 2. A line of length n is an interval graph, and we run algorithm A on this graph to color it with $o(\log^* n)$ colors. By Lemma 4, in $o(\log^* n)$ rounds, this coloring of the line can be reduced to a 3-coloring. Finally, if vertex s and t have the same color, then vertex s picks a different color that goes along with its two neighbors. The runtime of this procedure is clearly $o(\log^* n)$. □

4 Simple Coloring Algorithm

We show now that an algorithm that follows the scheme of Algorithm 1 can be used to compute a $(k \log n)$-coloring on graphs G that have the property that every induced subgraph on α nodes has at most αk edges (Theorem 6). This property is fulfilled by k-degenerate graphs since clearly k-degenerate graphs have at most kn edges, and k-degeneracy inherits to induced subgraphs. Furthermore, it is easy to see that the degeneracy k of an inductive d-independent graph is bounded as $k \leq d\chi(G)$. Theorem 1 as stated in the introduction follows hence immediately from Theorem 6.

In order to color a graph with a limited number of edges in each induced subgraph with an algorithm of type Algorithm 1, we use the following sequence of probabilities: we start with probability $p_1 = 1$, and we repeat it $\frac{32e^2 \log n}{p_1}$ times. Then, we halve this probability, i.e., $p_2 = p_1/2$ and we repeat it $\frac{32e^2 \log n}{p_2}$ times. This procedure of halving the previous probability $p_{i+1} = p_i/2$ and repeating it $\frac{32e^2 \log n}{p_{i+1}}$ times continues until all nodes are colored. We will prove now Theorem 6. We note again that this type of proof was already used in [18] and [17].

Theorem 6. *There is an algorithm that follows the scheme of Algorithm 1 and colors graphs $G = (V, E)$ that have the property that every induced subgraph on α vertices has at most αk edges with $\mathrm{O}(k \log n)$ colors and rounds w.h.p. Thus, the algorithm uses $O(d\chi(G) \log n)$ colors, where d is the inductive independence.*

Proof. We use the previously described sequence of probabilities. Let i be such that $k \le \frac{1}{p_i} < 2k$, and let U be the subset of vertices of V that are not yet colored just before the first iteration with probability p_i. The graph $G|_U$ has at most $k|U|$ edges, and, therefore, by the Markov Inequality, at least $\frac{1}{2}|U|$ nodes of U have a degree of at most $2k$. Let U' be this subset. The probability that a node $u \in U'$ is colored in an iteration where the current probability is p_i is:

$$\begin{aligned}\mathbb{P}\left[u \text{ is colored}\right] &= \mathbb{P}\left[u \text{ is pre-selected}\right] \cdot \mathbb{P}\left[\text{ no neighbor of } u \text{ is pre-selected}\right] \\ &= p_i \cdot (1 - p_i)^{\deg_{U'}(u)} \ge \frac{1}{2k} \cdot \left(1 - \frac{1}{k}\right)^{2k} \ge \frac{1}{2k}\left(\frac{1}{2e}\right)^2,\end{aligned}$$

where we used $(1 - 1/x)^x \ge 1/(2e)$ for any $x \ge 2$. Let $U_0 = U$ and denote by U_l the number of uncolored nodes after l iterations with probability p_i. Then:

$$\begin{aligned}\mathbb{E}|U_l| &\le \mathbb{E}|U_{l-1}| - \frac{1}{2k}\left(\frac{1}{2e}\right)^2 \frac{1}{2}\mathbb{E}|U_{l-1}| = \mathbb{E}|U_{l-1}|\left(1 - \frac{1}{k16e^2}\right) \\ &= |U_0|\left(1 - \frac{1}{k16e^2}\right)^l \le ne^{-\frac{l}{k16e^2}}.\end{aligned}$$

We have $\mathbb{E}|U_t| \le n^{-1}$ for $t \ge 32e^2 k \log n$. Therefore, by the Markov Inequality, we conclude that

$$\mathbb{P}\left[|U_t| \ge 1\right] \le \mathbb{E}|U_t| = n^{-1}.$$

Therefore, running $32e^2 k \log n \le \frac{32e^2 \log n}{p_i}$ iterations of the algorithm with probability p_i will color all nodes with high probability. The iterations with probabilities p_j with $j < i$ do only account to a constant factor in the total number of iterations since we double the number of iterations each time we increase the probability. □

5 Implementation in the Beep Model

In the discrete beeping communication model as introduced in [19], nodes of a network modeled by a graph $G = (V, E)$ communicate with each other via *beeps.* Nodes are not aware of their neighborhoods. In each round, a node $v \in V$ has the choice between two actions: Either v transmits a beep signal (v beeps), or v is in listening mode. If v is in listening mode, then v receives a signal only if at least one of its neighbors transmits a beep. The reception of a beep signal does not allow v to determine the number of its neighbors that transmitted it. Node v can only distinguish between the situation where none of its neighbors transmitted, or at least one of its neighbors transmitted. While in [19] asynchronous wake-up times of nodes are considered, we assume a synchronous model where all nodes are awake at time 0. Furthermore, we assume that nodes know a polynomial upper bound on n, the number of nodes. We assume that they have only $O(\log n)$ memory.

Despite the fact that the discrete beeping model is very restrictive, many non-trivial problems can be solved in this model. It models aspects of wireless networks (carrier sensing) and biological phenomena.

Algorithms that can be implemented in this model can certainly be implemented in many other distributed models.

We will show now that the scheme of Algorithm 1 can be implemented in the discrete beeping model. In Line 9 of Algorithm 1, a pre-selected node has to determine whether either none of its neighbors pre-selected themselves, or whether there is at least one neighbors that pre-selected itself. Note that if we gave a node the ability to beep and listen at the same time, Line 9 of Algorithm 1 could be implemented in one communication round. The main difficulty for an implementation in the discrete beeping model stems from the fact that if a node decides to beep it cannot receive any information. Therefore, the pre-selected nodes cannot simply beep simultaneously in one round since the beep of a node wouldn't be heard by another beeping node. We will show, however, that this task can be computed in $\mathrm{O}(\log n)$ rounds of communication. In the following, we denote by beep() the action that a node decides to beep, and by listen() the action that a node is in listening mode. The function listen() returns *true*, if at least one neighboring node beeped, otherwise it returns *false*. Algorithm 3 implements one round of Algorithm 1 in the discrete beeping model.

Algorithm 3 Iteration i in the beep model

```
Require: $p_i$ {Probability $p_i$, integer $C \geq 4$}
1: if coin($p_i$) then
2:     Select uniformly at random $S \subset \{1, 2, \ldots, C \log n\}$ such that $|S| = \frac{C \log n}{2}$
3:     $B \leftarrow$ false
4:     for $l = 1 \ldots C \log n$ do
5:         if $l \in S$ then beep() else $B \leftarrow B \vee$ listen() end if
6:     end for
7:     if $B =$ false then $\gamma(v) \leftarrow i$ end if
8: end if
```

Lemma 5. *Let $C \geq 4$ be an integer. The probability that a pre-selected node $v \in V$ colors itself in Line 7, despite having a pre-selected neighbor, is at most $\frac{1}{n^{C-3}}$ assuming that $n > 2C$.*

Proof. Let $v \in V$ be a node that pre-selected itself (coin(p_i) in Line 1 evaluated *true*). Let $V' \subseteq \Gamma_G(v)$ denote the neighbors of v that also pre-selected themselves, and suppose that $|V'| > 0$. Denote by S the subset of indices that v has chosen, and for a pre-selected neighbor $u \in V'$ of v, denote by S_u the indices that u has chosen. Then, v colors itself only if $\forall u \in V' : S_u = S$. Denote by $N = |\Gamma_G(v)|$ the size of the neighborhood of node v. We bound this probability as follows:

$$\mathbb{P}\left[\forall u \in V' : S_u = S\right] = \sum_{i \geq 1} \mathbb{P}\left[\forall u \in V' : S_u = S \,|\, |V'| = i\right] \cdot \mathbb{P}\left[|V'| = i\right] \leq \sum_{i \geq 1}^{N} \left(\frac{1}{\binom{C \log n}{\frac{1}{2} C \log n}} \right)^{i}, \tag{1}$$

since clearly $\mathbb{P}\left[|V'| = i\right] \leq 1$ for all i. By the formula for the central binomial coefficient, we have $\binom{C \log n}{\frac{1}{2} C \log n} \geq 2^{C \log(n) - \log(2C) - \log \log n}$. Using this in Inequality 1, we obtain

$$\sum_{i \geq 1}^{N} \left(\frac{1}{\binom{C \log n}{\frac{1}{2} C \log n}} \right)^{i} \leq \sum_{i \geq 1} 2^{-i(C \log(n) - \log(2C) - \log \log n)} \leq N 2^{-C \log(n) + \log(2C) + \log \log n} \leq \frac{1}{n^{C-3}},$$

for $n \geq 2C$. □

Since every node may get pre-selected at most $\mathrm{O}(d\chi(G) \log n)$ times in the algorithm of Theorem 1, selecting a large enough value for C (for instance $C \geq 7$) guarantees that the overall error probability is small enough when implementing this algorithm in the discrete beeping model. From Lemma 5 and Theorem 1 we obtain the following corollary:

Corollary 1. *There is an algorithm that follows the scheme of Algorithm 1 that can be implemented in the discrete beeping model with* $\mathrm{O}(d\chi(G)\log^2 n)$ *rounds and colors a graph with* $\mathrm{O}(d\chi(G)\log n)$ *colors w.h.p. where* d *is the inductive independence number of a graph.*

6 Lower Bound For Algorithms of Type Algorithm 1

We discuss now a hard instance showing that no algorithm that follows the scheme of Algorithm 1 can achieve an approximation ratio of $o(\frac{\log n}{\log\log n})$ on interval graphs. We present the hard instance graph $G_{T,b} = (V, E)$ in its interval representation, where T and b are parameters as follows: As basic building blocks of our construction we use cliques of size $T = o(n)$ (we determine the precise value of T later). Their adjacency relations follow a tree structure with branching factor $\log^b n$ for an integer $b \geq 6$ (we set $b = 6$, but any constant $b \geq 6$ equally works), and we obtain a *containment interval graph* as in Figure 2, i.e., an interval graph where the set $\{\Gamma_{G_{T,b}}(v) \mid v \in V\}$ forms a laminar family. The vertex set V is decomposed into layers $V_0, \ldots, V_k$. We have $|V_i| = T \cdot (\log n)^{ib}$, and, therefore, $k = \Theta(\frac{\log n}{\log\log n})$ in order to have a total of n vertices. The chromatic number of this graph is $\chi(G_{T,b}) = Tk$. We aim to construct the hard instances for a given chromatic number, and we therefore set the parameter T to be $T = \chi(G_{T,b})/k$. Let us summarize the values of our parameters: We consider the n-vertex graph $G_{T,b}$ with chromatic number $\chi(G_{T,b})$ and we set $T = \chi(G_{T,b})/k = \Theta(\chi(G_{T,b}) \log\log n/ \log n)$ and $b = 6$.

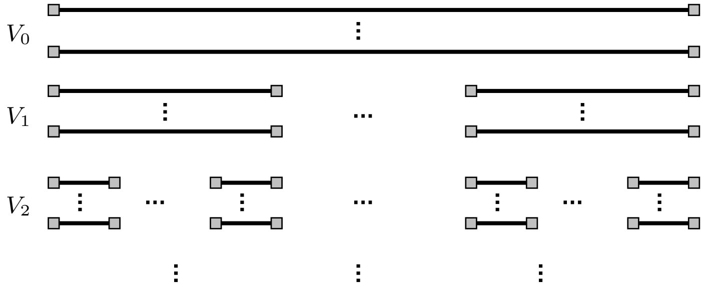

Fig. 2. Hard instance $G_{T,b} = (V, E)$. V_0 is a clique of size T, V_1 are $\log^b(n)$ cliques each of size T. This construction continues recursively until level $k = \Theta(\frac{\log(\frac{n}{T})}{\log\log n})$.

We shall prove now that any algorithm following the scheme of Algorithm 1 requires $\Omega(\chi(G_{T,b})k)$ iterations on graph $G_{T,b}$. However, due to space restrictions, all proofs of this section are omitted and can be found in the full version of this article. Our argument is as follows: Let $p_1, p_2, \ldots$ be the sequence of probabilities chosen by the algorithm, where p_i is the probability chosen in round i. We will argue that for any $k/2 \leq i < k$, layer V_{i+1} will be eliminated by the algorithm before the elimination of at most 1/10 of the nodes of layer V_i since the presence of layer V_{i+1} induces high degrees to all nodes in layer V_i. For the nodes in V_i, this reduces the probability of being selected and colored. We show that the elimination of a layer takes time $\Omega(\chi(G_{T,b}))$ for any choice of probabilities. Since there are $k = \Theta(\frac{\log n}{\log\log n})$ layers, the result follows.

Denote by $V_i^j \subseteq V_i$ the set of not-yet colored nodes after iteration j. Then $V_i^0 = V_i$. Let $V^j = \bigcup_i V_i^j$. Denote by t_i the least number of iterations of the algorithm such that at least one clique of V_i lost at least $1/2T$ of its vertices, i.e., at least half of the vertices of at least one clique of layer i have disappeared. In any iteration j, any node $v \in V^j$ gets pre-selected with probability p_j. Then it is colored only if none of its neighbors have been preselected. Therefore, the probability of v being chosen and colored is $p_i(1-p_i)^{\deg_{V^j}(v)}$.

Next, we show that it is very unlikely that a node of layer i is colored before iteration t_{i+1}.

Lemma 6. *Consider graph $G_{T,b}$. Let $0 \leq i < k$. For every iteration $j < t_{i+1}$, every $v \in V_i^j$, and large enough n:*

$$\mathbb{P}\left[v \textit{ is colored in iteration } j\right] \leq \frac{1}{T \log^{b-2} n}.$$

Proof. The probability for v to be colored in round j is $p_j(1-p_j)^{\deg_{V^j}(v)}$. Since $j < t_{i+1}$, we have $\deg(v) \geq T/2 \log^b(n)$, and hence

$$\mathbb{P}\left[v \text{ is colored in iteration } j\right] \leq p_j(1-p_j)^{T/2\log^b n} \leq \min\{p_j, (1-p_j)^{T/2\log^b n}\}. \tag{2}$$

We consider the cases $p_j \leq \frac{1}{T\log^{b-2} n}$ and $p_j > \frac{1}{T\log^{b-2} n}$ separately. If $p_j \leq \frac{1}{T\log^{b-2} n}$ then the RHS of Inequality 2 is clearly also smaller than $\frac{1}{T\log^{b-2} n}$. Suppose that $p_j > \frac{1}{T\log^{b-2} n}$. Then the RHS of Inequality 2 is bounded by

$$(1-p_j)^{T/2\log^b n} \leq e^{-\frac{T\log^b(n) p_j}{2}} < e^{-\frac{\log^2 n}{2}} < \frac{1}{n^{\frac{1}{2}\log n}},$$

where we used $1 - p_j \leq e^{-p_j}$. Therefore, for large enough n, Inequality 2 is bounded by $\frac{1}{T\log^{b-2} n}$. □

This fact is then used in the following lemma. With high probability, all cliques of layer i are still of size at least 9/10 of its initial size just after iteration t_{i+1}.

Lemma 7. *Consider graph $G_{T,b}$. Suppose that $t_{i+1} \leq T\log^2 n$. Then with probability at least $1 - \mathrm{O}\left(\frac{1}{n^{\log^{b-5}(n)T-1}}\right)$ and n large enough, the size of the smallest clique in V_j after iteration t_{i+1} is at least $\frac{9}{10}T$ for any $j \leq i$.*

Proof. For any moment $j < t_{i+1}$, by Lemma 6, the probability that a node $v \in V_j$ is colored is at most $\frac{1}{T\log^{b-2} n}$. Using the assumption $t_{i+1} \leq T\log^2 n$, we obtain

$$\mathbb{P}\left[v \in V_j \text{ colored in the first } t_{i+1} \text{ rounds }\right] \leq \frac{t_{i+1}}{T\log^{b-2} n} \leq \frac{1}{\log^{b-4} n}.$$

Consider any clique $C \in V_j$. Then $C \setminus V_j^{t_{i+1}}$ is the number of nodes of clique C that have been colored in the first t_{i+1} rounds. By the previous inequality, we obtain

$$\mathbb{E}|C \setminus V_j^{t_{i+1}}| \geq T \cdot \frac{1}{\log^{b-4} n},$$

and we apply the Chernoff bound (for $\delta = \frac{\log^{b-4}(n)}{10} - 1$):

$$\begin{aligned}\mathbb{P}\left[|C \setminus V_j^{t_{i+1}}| \geq 1/10T\right] &\leq e^{-1/3\cdot(\frac{\log^{b-4}(n)}{10}-1)^2\cdot T\frac{1}{\log^{b-4} n}} \\ &< e^{-\frac{\log^{b-4}(n)T}{300}} = \Theta\left(\frac{1}{n^{\log^{b-5}(n)T}}\right).\end{aligned}$$

Since there are at most n/T cliques in V_j for all j, we use the union bound and we obtain

$$\mathbb{P}\left[\exists \text{clique } C \in V_j : |C \setminus V_j^{t_{i+1}}| \geq 1/10T\right] \leq \frac{n}{T}\Theta\left(\frac{1}{n^{\log^{b-5}(n)T}}\right) = \mathrm{O}\left(\frac{1}{n^{\log^{b-5}(n)T-1}}\right).$$

□

Using Lemma 7 we conclude that the number of rounds between t_{i+1} and t_i is $\Theta(\chi(G))$.

Lemma 8. *Suppose that $G_{T,b}$ is such that $\chi(G_{T,b}) \geq \log^5 n$. Then for any $i \geq k/2$, a small enough but constant c, and n large enough:*

$$\mathbb{P}\left[t_i - t_{i+1} \leq c\chi(G)\right] = \mathrm{O}\left(\frac{1}{n^{\log n - 1}}\right).$$

Proof. Let E denote the event that every gadget $C \in V_j$ is of size at least $(9/10)T$ before iteration t_{j+1}. Then by Lemma 7, we have $\mathbb{P}\left[E\right] \geq 1 - \mathrm{O}\left(\frac{1}{n^{\log^{b-5}(n)T-1}}\right)$. Conditioned on E, for any $t_{i+1} - 1 \leq j \leq t_{k/2}$, the degree of any node $v \in V_i^j$ is therefore at least $\frac{9Tk}{20}$. Since the probability that any node v is colored in round i is $p_i(1-p_i)^{\deg_{V^i}(v)}$, the probability that a node $v \in V_i^j$ for $t_{i+1} - 1 \leq j \leq t_{k/2}$ is colored in round j is at most:

$$\mathbb{P}\left[v \text{ is colored in round } j \,|\, E\right] = p_j(1-p_j)^{\frac{9Tk}{20}} \leq p_j e^{-\frac{9Tkp_j}{20}} \leq \frac{20}{9Tk},$$

for any p_j, where we used $1+x \leq e^x$ for any x in the first inequality, and we applied Lemma 9 for the second inequality. Let t be an integer. Then conditioned on the event E, the probability for one node $v \in V_i^j$ with $t_{i+1} - 1 \leq j \leq t_{k/2} - t$ to be colored in t rounds is at most $t \cdot \frac{20}{9Tk}$. Consider now a clique $C \in V_i$. Then the expected number of eliminated nodes of C between round $t_{i+1} - 1$ and round $t_{i+1} - 1 + t$ is bounded as follows:

$$\mathbb{E}\left[|C \setminus V_i^{t_{i+1}-1+t}| - |C \setminus V_i^{t_{i+1}-1}|\right] \leq Tt \cdot \frac{20}{9Tk} = \frac{20t}{9k}.$$

For any $t < cTk$ and for a small enough but constant c, we obtain by the Chernoff bound:

$$\mathbb{P}\left[|C \setminus V_i^{t_{i+1}-1+t}| - |C \setminus V_i^{t_{i+1}-1}| \geq 4/10T\right] \leq e^{-\frac{\left(\frac{18kT}{100t}\right)^2 20t}{3 \cdot 9k}} = \frac{1}{e^{\Theta(\frac{T^2}{tk})}},$$

and we conclude that conditioned on the event E:

$$\mathbb{P}\left[|C \setminus V_i^{t_{i+1}-1+t}| \geq 5/10T \,|\, E\right] \leq \frac{1}{e^{\Theta(\frac{T^2}{tk})}}.$$

Now, since there are at most $\frac{n}{T}$ gadgets in any level i, we apply the union bound and obtain:

$$\mathbb{P}\left[\text{every gadget } C \in V_i \text{ is such that} |C \setminus V_i^{t_{i+1}-1+t}| \geq 5/10T \,|\, E\right] \leq \frac{n}{T}\frac{1}{e^{\Theta(\frac{T^2}{tk})}}.$$

Since for any two events A, B, we have $\mathbb{P}\left[A\right] \leq \mathbb{P}\left[A \,|\, B\right] + (1 - \mathbb{P}\left[B\right])$, we obtain

$$\begin{aligned}
&\mathbb{P}\left[\text{every gadget } C \in V_i \text{ is such that} |C \setminus V_i^{t_{i+1}-1+t}| \geq 5/10T\right] \leq \\
&\qquad \mathbb{P}\left[\text{every gadget } C \in V_i \text{ is such that} |C \setminus V_i^{t_{i+1}-1+t}| \geq 5/10T \,|\, E\right] + (1 - \mathbb{P}\left[E\right]) \leq \\
&\qquad\qquad \frac{n}{T}\frac{1}{e^{\Theta(\frac{T^2}{tk})}} + \mathrm{O}\left(\frac{1}{n^{\log^{b-5}(n)T-1}}\right). \qquad (3)
\end{aligned}$$

Note that $T = \chi(G)/k$. Furthermore, $k = \mathrm{O}(\log n)$ and we assumed that $\chi(G) = \Omega(\log^5 n)$. We furthermore assumed that $t \leq c\chi(G)$. Then, the first addend of the right side of Inequality 3 is bounded as follows:

$$\frac{n}{T}\frac{1}{e^{\Theta(\frac{T^2}{tk})}} = \frac{nk}{\chi(G)}\frac{1}{e^{\Theta(\frac{\chi(G)^2}{tk^3})}} = \mathrm{O}\left(\frac{n}{\log^4 n}\frac{1}{e^{\log^2 n}}\right) = \mathrm{O}\left(\frac{1}{n^{\log n - 1}}\right).$$

This expression also bounds Inequality 3 which proves the lemma. □

The following lemma was used in the proof of Lemma 8.

Lemma 9. *For any real number x and any value $0 \leq p \leq 1$, the following inequality holds:*

$$p \cdot e^{-xp} \leq \frac{1}{x}.$$

Proof. Clearly, for any real number y, we have $y \leq e^y$. Let $y = xp$ and we obtain $xp \leq e^{xp}$. This implies the result. □

Lemma 8 allows us to obtain our lower bound result.

Theorem 7. *Suppose that $G_{T,b}$ is such that $\log^5 n \leq \chi(G_{T,b}) \leq n^{1-\epsilon}$ for any $\epsilon > 0$ and let n be sufficiently large. Then any algorithm that follows the scheme of Algorithm 1 requires $\Omega(\chi(G_{T,b})\frac{\log n}{\log\log n})$ colors to color $G_{T,b}$ with high probability.*

Proof. It follows from Lemma 8 that the elimination of one level $i \geq k/2$ takes at least $\Omega(\chi(G))$ rounds with probability at least $1 - \mathrm{O}\left(\frac{1}{n^{\log n - 1}}\right)$. Therefore, by the union bound, the elimination of the first $k/3$ levels takes $\Omega(\chi(G)k)$ rounds with probability at least $1 - \log n \cdot \mathrm{O}\left(\frac{1}{n^{\log n - 1}}\right) = 1 - \mathrm{O}\left(\frac{1}{n^{\log n - 2}}\right)$. □

References

1. Marathe, M., Breu, H., III, H.B.H., Ravi, S.S., Rosenkrantz, D.J.: Simple heuristics for unit disk graphs. Networks **25** (1995) 59–68
2. Breu, H., Kirkpatrick, D.G.: Unit disk graph recognition is NP-hard. Comput. Geom. Theory Appl. **9**(1-2) (1998) 3–24
3. Halldórsson, M.M.: Wireless scheduling with power control. ACM Trans. Algorithms **9**(1) (2012) 7:1–7:20
4. Karp, R.M.: Reducibility Among Combinatorial Problems. In Miller, R.E., Thatcher, J.W., eds.: Complexity of Computer Computations. Plenum Press (1972) 85–103
5. Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. In: Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing. STOC '06, New York, NY, USA, ACM (2006) 681–690
6. Smith, D.A.: The First-fit Algorithm Uses Many Colors on Some Interval Graphs. PhD thesis, Tempe, AZ, USA (2010) AAI3428197.
7. Linial, N.: Locality in distributed graph algorithms. SIAM J. Comput. **21**(1) (1992) 193–201
8. Kuhn, F., Wattenhofer, R.: On the complexity of distributed graph coloring. In: Proceedings of the Twenty-fifth Annual ACM Symposium on Principles of Distributed Computing. PODC '06, New York, NY, USA, ACM (2006) 7–15
9. Kothapalli, K., Scheideler, C., Onus, M., Schindelhauer, C.: Distributed coloring in $\sqrt{\log n}$ bit rounds. In: Proceedings of the 20th International Conference on Parallel and Distributed Processing. IPDPS'06, Washington, DC, USA, IEEE Computer Society (2006) 44–44
10. Barenboim, L., Elkin, M.: Deterministic distributed vertex coloring in polylogarithmic time. J. ACM **58**(5) (2011) 23:1–23:25
11. Métivier, Y., Robson, J.M., Saheb-Djahromi, N., Zemmari, A.: On the time and the bit complexity of distributed randomised anonymous ring colouring. Theor. Comput. Sci. **502** (2013) 64–75
12. Luby, M.: A simple parallel algorithm for the maximal independent set problem. In: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing. STOC '85, New York, NY, USA, ACM (1985) 1–10
13. Barenboim, L., Elkin, M., Pettie, S., Schneider, J.: The locality of distributed symmetry breaking. In: Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science. FOCS '12, Washington, DC, USA, IEEE Computer Society (2012) 321–330
14. Cole, R., Vishkin, U.: Deterministic coin tossing with applications to optimal parallel list ranking. Inf. Control **70**(1) (1986) 32–53
15. Schneider, J., Wattenhofer, R.: An Optimal Maximal Independent Set Algorithm for Bounded-Independence Graphs. Distributed Computing **22** (2010)
16. Fanghänel, A., Kesselheim, T., Vöcking, B.: Improved algorithms for latency minimization in wireless networks. Theor. Comput. Sci. **412**(24) (2011) 2657–2667

17. Kesselheim, T., Vöcking, B.: Distributed contention resolution in wireless networks. In: Proceedings of the 24th International Conference on Distributed Computing. DISC'10, Berlin, Heidelberg, Springer-Verlag (2010) 163–178
18. Halldórsson, M.M., Mitra, P.: Nearly optimal bounds for distributed wireless scheduling in the SINR model. In: Proceedings of the 38th International Conference on Automata, Languages and Programming - Volume Part II. ICALP'11, Berlin, Heidelberg, Springer-Verlag (2011) 625–636
19. Cornejo, A., Kuhn, F.: Deploying wireless networks with beeps. In: Proceedings of the 24th International Conference on Distributed Computing. DISC'10, Berlin, Heidelberg, Springer-Verlag (2010) 148–162
20. Halldórsson, M.M., Holzer, S., Mitra, P., Wattenhofer, R.: The power of non-uniform wireless power. In: Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013. (2013) 1595–1606
21. Ye, Y., Borodin, A.: Elimination graphs. ACM Trans. Algorithms **8**(2) (2012) 14:1–14:23