Linial for lists

Yannic Maus¹ · Tigran Tonoyan²

Received: 15 January 2021 / Accepted: 16 September 2021 / Published online: 17 May 2022
© The Author(s) 2022

Abstract
Linial’s famous color reduction algorithm reduces a given \( m \)-coloring of a graph with maximum degree \( \Delta \) to an \( O(\Delta^2 \log m) \)-coloring, in a single round in the LOCAL model. We give a similar result when nodes are restricted to choose their color from a list of allowed colors: given an \( m \)-coloring in a directed graph of maximum outdegree \( \beta \), if every node has a list of size \( \Omega(\beta^2(\log \beta + \log \log m + \log \log |C|)) \) from a color space \( C \) then they can select a color in two rounds in the LOCAL model. Moreover, the communication of a node essentially consists of sending its list to the neighbors. This is obtained as a corollary, we improve the state-of-the-art truly local \((\deg+1)\)-list coloring algorithm from Baranboim et al. (PODC, pp 437–446, 2018) by slightly reducing the runtime to \( O(\sqrt{\Delta \log \Delta} + \log^* n) \) and significantly reducing the message size (from \( \Delta^{O(\log^* \Delta)} \) to roughly \( \Delta \)). Our techniques are inspired by the local conflict coloring framework of Fraigniaud et al. (in: FOCS, pp 625–634, 2016).

Keywords Deterministic distributed coloring · Color reduction · Cover-free family · List coloring

1 Introduction

Symmetry breaking problems are a cornerstone of distributed graph algorithms in the LOCAL model.¹ A central question in the area asks: How fast can these problems be solved in terms of the maximum degree \( \Delta \), when the dependence on \( n \) is as mild as \( O(\log^* n) \)? [11]. That is, we are looking for truly local algorithms, with complexity of the form \( f(\Delta) + O(\log^* n) \).² The \( O(\log^* n) \) term is unavoidable due to the seminal lower bound by Linial [41] that, via simple reductions, applies to most typical symmetry breaking problems. \((\Delta + 1)\)-vertex coloring and maximal independent set (MIS) are the key symmetry breaking problems. Both can be solved by simple centralized greedy algorithms (in particular they are always solvable), and even more importantly in a distributed context, any partial solution (e.g. a partial coloring) can be extended to a complete solution. The complexity of MIS is settled up to constant factors with \( f(\Delta) = \Theta(\Delta) \), by the algorithm from [13] and a recent breakthrough lower bound by Balliu et al. [3]. In contrast, the complexity of vertex coloring, despite being among the most studied distributed graph problems [11], remains widely open, with the current best upper bound \( f(\Delta) = \tilde{O}(\sqrt{\Delta}) \) and lower bound \( f(\Delta) = \Omega(1) \). While most work has focused on the \((\Delta+1)\)-coloring problem, recent algorithms, e.g., [8, 12, 23, 37], rely on the more general list coloring problem as a subroutine, where each vertex \( v \) of a graph \( G \) has a list \( L_v \subseteq C \) of colors from a colorspace \( C \), and the objective is to compute a proper vertex coloring, but each vertex has to select a color from its list. Again, a natural case is the always solvable \((\deg+1)\)-list coloring problem, where the list of each vertex is larger than

---

¹ In the LOCAL model [41, 46] a communication network is abstracted as an \( n \)-node graph \( G = (V, E) \) with unique \( O(\log n) \)-bit identifiers. Communications happen in synchronous rounds. Per round, each node can send one (unbounded size) message to each of its neighbors. At the end, each node should know its own part of the output, e.g., its own color. In the CONGEST model, there is a limitation of \( O(\log n) \) bits per message.

² Not all symmetry breaking problems admit such a runtime, e.g., \( \Delta \)-coloring and sinkless orientation [17].
its degree. Our paper contributes to the study of list coloring problems. To set the stage for our results, let us start with an overview of truly local coloring algorithms.

In [41], besides the mentioned lower bound, Linial showed that an $O(\Delta^2)$-coloring can be done in $O(\log^* n)$ rounds. Szegedy and Vishwanathan [49] improved the runtime to $\frac{1}{2} \log^* n + O(1)$, and showed that in additional $O(\Delta \cdot \log \Delta)$ rounds, the $O(\Delta^3)$-coloring could be reduced to $(\Delta + 1)$-coloring. The latter result was rediscovered by Kuhn and Wattenhofer [39]. Barenboim, Elkin and Kuhn used defective coloring for partitioning the graph into low degree subgraphs and coloring in a divide-and-conquer fashion, and brought the complexity of $(\Delta + 1)$-coloring down to $O(\Delta + \log^* n)$ [13]. A simpler algorithm with the same runtime but without using defective coloring was obtained recently in [12, 42]. All of these results also hold for low outdegree colorings, which led to a simpler list coloring approach, with the difference that the graph is partitioned into subgraphs, each equipped with a color space size independence of Theorem 1 (see the technical overview below). It was not clear prior to our work whether the situation simplifies significantly if one restricts to ordinary list coloring because, even if the input to the algorithm in Theorem 1 is a list coloring problem, the intermediate stages of the algorithm fall back to the more general local conflict coloring.

The overarching goal of our paper is providing deeper understanding of the remarkable framework of [23], better connecting it with classic works in the area, and obtaining a simpler list coloring algorithm that also uses smaller messages, by moderately sacrificing generality.

1.1 Our contribution

Our main result is a simple algorithm that yields the following theorem.

Theorem 2 (Linial for Lists) In a directed graph with max. degree $\Delta$, max. outdegree $\beta$, and an input $m$-coloring, list coloring problem with lists $L_v$ from a color space $C$ and of size $|L_v| \geq l_0 = 4e\beta^2(4\log \beta + \log \log |C| + \log \log m + 8)$ can be solved in 2 rounds in LOCAL. Each node sends $l_0 + 1$ colors in the first round, and a $l_0/\beta^2$-bit message in the second.

The name “Linial for Lists” stems from the fact that Theorem 2 is a “list version” of one of the cornerstones of distributed graph coloring, Linial’s color reduction, which says that an $m$-coloring can be reduced to a $(5\Delta^2 \log m)$-coloring in a single round [41]. Moreover, our framework is itself a natural generalization of Linial’s approach of cover-free set families. Applied to equal lists, it yields an alternative
proof of Linial’s color reduction, in the form of a greedy construction of cover-free families (Linial proved their existence using the probabilistic method [41]; he also used an alternative construction from [21] via polynomials over finite fields, which however yields a weaker color reduction for $m \gg \Delta$) (see Sects. 2.3 and 6).

Compared with Theorem 1, we lose colorspace independence, and our algorithm does not extend to general local conflict coloring (although we use a kind of conflict coloring in the process). In exchange, we eliminate $\Delta$ from the bound on the list size, reduce the runtime to exactly 2 rounds, and dramatically reduce message size. This is achieved by a non-trivial paradigm shift in the local conflict coloring framework (see the technical overview below). The runtime cannot be reduced to 1 round, due to a lower bound of [49] (see Sect. 6).

Combining Theorem 2 with the partitioning algorithm of [12] as outlined above gives us a $(\deg + 1)$-list coloring algorithm. Note that any improvement in the list size bound in Theorem 2 (with little increase in runtime) would yield a faster $(\deg + 1)$-list coloring algorithm.

**Theorem 3** ($(\deg + 1)$-List Coloring) In a graph with max. degree $\Delta$, $(\deg + 1)$-list coloring problem with lists $L_v \subseteq C$ from a color space of size $|C| = 2^{\poly(\Delta)}$ can be solved in $O(\sqrt{\Delta \log \Delta}) + \frac{1}{\Delta} \cdot \log^* n$ rounds in LOCAL.

Furthermore, each node only needs to broadcast to its neighbors a single non-CONGEST message consisting of a subset of its list (i.e., of size $\poly(\Delta)$ which could be $\omega(\log n)$).

The bound on the color space size stems from the color space dependence in Theorem 2. As discussed in Sect. 6, it is possible to trade color space dependence with runtime in Theorem 2, which could improve or suppress the bound in Theorem 3. That, however, comes with the cost of having large messages.

Theorem 3 immediately provides the fastest known truly local $(\Delta + 1)$-coloring algorithm in LOCAL. Below we list further implications of our framework.

– **CONGEST (see Cor. 1):** We obtain an improved $(\Delta + 1)$-coloring algorithm in a low degree regime in CONGEST. In particular, if $\Delta = \tilde{O}(\log n)$ then $(\Delta + 1)$-coloring (more generally, $(\deg + 1)$-list coloring with colorspace of size $|C| = \poly(\Delta)$) can be solved in $\tilde{O}(\sqrt{\Delta}) + \frac{1}{\Delta} \cdot \log^* n$ rounds in CONGEST. Generally, if one allows messages of size $B$, this runtime holds for degree up to $\Delta = \tilde{O}(B)$. On the other hand, if $\Delta = \Omega((\log^2 + \log \log m)$, for an arbitrarily small constant $\epsilon > 0$, an algorithm from recent work [37] achieves runtime $O(\sqrt{\Delta})$ in CONGEST (if one recasts their dependency on $n$ as a $\Delta$-dependency). Thus, only for the regime of $\Delta \in \Omega((\log^{1+\epsilon} n) \cap O((\log^{2+\epsilon} n)$ we do not have a deterministic algorithm with runtime $\tilde{O}(\sqrt{\Delta})$ in CONGEST (with the current best being $O(\Delta^{3/4} + \log^* n)$ due to [8]).

– **Defective list coloring (see Theorem 6):** Our framework extends to $d$-defective list coloring, that is, list coloring where each node $v$ can have at most $d$ neighbors with the same colors as $v$: If lists are of size $\Omega((\Delta/(d + 1))^2 \cdot (\log \Delta + \log \log \Delta + \log \log m))$ we can compute a $d$-defective list coloring in 2 rounds in LOCAL. The result can be seen as the “list variant” of a defective coloring result in [36]. While we are not aware of an immediate application, defective list coloring with a better “colors vs. defect” tradeoff $(d \times O(\Delta/d))$ for line graphs has recently been used to obtain an edge coloring algorithm with complexity quasi poly $\log \Delta + O(\log^* n)$ [6].

– **\(\Delta\)-coloring:** The improvements obtained in Theorem 3 also imply respective improvements for several $\Delta$-coloring algorithms that use $(\deg + 1)$-list coloring as a subroutine [27].

### 1.2 Technical overview

At their core, the proofs of Theorems 1 and 2 are based on three important concepts: conflict coloring, problem amplification and 0-round solvability. A conflict coloring problem is a list coloring problem where every node has a list of available colors but two colors can conflict even if they are not equal (in particular, a color can conflict with multiple colors). The associated conflict degree is the maximum number of conflicts per color a node can have. Problem amplification transforms one conflict coloring problem instance into another, as follows: given an input to a problem $A$, each node computes its input to another problem $B$ (perhaps by exchanging information along the way), with the property that, having a solution to $B$, a simple one round algorithm computes a solution to $A$, and 2. the list-size-to-conflict-degree $(l/d)$ ratio of $B$ is larger than that of $A$. Note that the first property essentially determines the conflicts in $B$, and usually a color in $B$ is a set of colors in $A$. The importance of the second property stems from the concept of 0-round solvability: an instance of a problem $B$ with large enough $l/d$ ratio can be solved in 0 rounds, i.e., with no communication.

From here, the plan is simple: take problem $P_0$, which is the list coloring problem, recast it as a conflict coloring problem, and amplify it into problems $P_1, \ldots, P_t$, so that $P_t$ is 0-round solvable. Then we can cascade down to a solution of problem $P_0$, in $t$ rounds. Crucially, in order to do the above, we need $P_0$ to have sufficiently large $l/d$ ratio to begin with (which explains the particular list size requirements in our theorems). The input $m$-coloring is used for tie-breaking in the 0-round solution of $P_t$.

In [23], local conflict coloring is the main problem type, where the conflict between two colors depends on who the
colors belong to, i.e., two colors can conflict along one edge of the graph and not conflict on another one. Their framework allows solving any local conflict coloring problem, and by re-modeling a problem in an arbitrary colorspace via mapping each list to an interval [1, l] of natural numbers, one can redefine local conflicts and “forget” about the real size of the colorspace (hence colorspace independence). When computing the input of \( P_t \) (given \( P_{t-1} \)), in order to maintain manageable conflict degree, nodes exchange messages to filter out colors in \( P_t \) that cause too much conflict with any neighboring node. These messages are large (recall that a color in \( P_t \) is a set of colors in \( P_{t-1} \)), although this can be implemented more efficiently by pre-collecting lists from a neighborhood of appropriate radius. Thus, the input to \( P_t \) is usually the topology, \( P_0 \)-lists and conflicts in the \( i \)-hop neighborhood of a node. The goal towards 0-round solvability is then to find a problem \( P_t \) whose \( l/d \) ratio is larger than the number of all \( r \)-hop neighborhood patterns (i.e., inputs). The complicated nature of the input to \( P_t \) also makes the 0-round solvability proof rather conceptually involved. The number \( t \) of problems required is about \( 3 \log^* (m + \Delta) \).

Our framework, on the other hand, is based on special global conflict coloring instances, where the conflict relation of two colors does not depend on the edge across which they are. This limits us to solving only ordinary list problems \( P_0 \).

Our key insight (see Sect. 3.3), which sets Theorems 1 and 2 apart, is that in our setting nodes do not need to communicate for computing the input to problems \( P_1, \ldots, P_t \). To achieve this, we show that when forming the lists for \( P_t \) from the input to \( P_{t-1} \), it suffices to drop “universally bad” colors (sets of colors in \( P_{t-1} \)), whose absence is enough to ensure moderate conflict degree towards any (!) other node. We achieve this by crucially exploiting the symmetry of the particular conflict coloring problems arising from ordinary list coloring.

Thus, the input of a node in \( P_t \) is just its input in \( P_0 \). This makes the 0-round solution (of \( P_t \)) particularly simple. The only communication happens when we cascade down from a solution of \( P_t \) to that of \( P_0 \). With \( t = 2 \), we get our main theorem. Since here we have only two problems, the message size is limited (the first round is needed to learn the \( P_0 \)-lists of neighbors, while the second one consists of a small auxiliary message). Taking larger \( t \) would reduce the requirement on the initial list size but increase message size (see Sect. 6). Since \( t = 2 \) is sufficient for our applications, we limit our exposition to that case. Setting \( t = 1 \) does not give anything non-trivial for list coloring, since the \( l/d \) ratio is not large enough, but when all \( P_0 \)-lists are equal, it gives an alternative proof of Linial’s color reduction (Sect. 2.3). In fact, \( P_1 \) is essentially the problem of finding a low intersecting set family, which Linial’s algorithm is based on, while \( P_2 \) is a “higher-dimensional” variant of it. Thus, at the core of our result there is an (offline) construction of certain set families over the given color space: given those, the algorithm is easy. This way, we believe our paper also provides a deeper insight into the framework of Theorem 1. Our result can also be seen as a bridge between the results of [23] and the recently popular concept of speedup (see Sect. 6).

1.3 Further related work

Most results on distributed graph coloring until roughly 2013 are covered in the excellent monograph by Barenboim and Elkin [11]. An overview of more recent work can be found in [37]. Due to the large volume of published work on distributed graph coloring, we limit this section to an informative treatment of a selected subset. While we have covered most literature on truly local vertex coloring algorithms, there are many known algorithms that trade the high \( \Delta \)-dependence in the runtime with lower \( n \)-dependence. All deterministic algorithms in this category (for general input graphs) involve a \( \Omega(\log n) \) factor. From the early 90s until very recently, the complexity of (deg + 1)-list coloring (and \( (\Delta + 1) \)-coloring) in terms of \( n \) was \( 2^{O(\sqrt{\log n})} [1, 45] \), with algorithms based on network decomposition (into small diameter components). A recent breakthrough in network decomposition algorithms [47] brought the runtime of (deg + 1)-list coloring down to poly \( \log n \) in LOCAL (it also applies to many other symmetry breaking problems; see [26, 29, 47]). A poly log \( n \)-round CONGEST algorithm appeared a little later [7].

Historically, decompositions into subgraphs that are equipped with small outdegree orientations as used in our results, in [23], and in [8] are closely related to the notion of arboricity. To the best of our knowledge, orientations were first introduced as a tool for distributed graph coloring by Barenboim and Elkin [9]. They showed how to compute \( O(n) \)-outdegree acyclic orientations in graphs with arboricity \( a \) in \( O(\log n) \) rounds, and used those to devise several algorithms to color graphs with bounded arboricity. They also noticed that the degree bound of \( \Delta \) in Linial’s color reduction can be replaced with a bound on the outdegree. In another paper, they designed algorithms to recursively partition an input graph into small-arboricity subgraphs, which were used to obtain an \( O(\log \Delta \log n) \)-round algorithm for \( O(\Delta^{1+\epsilon}) \)-coloring and an \( O(\Delta^\epsilon \log n) \)-round algorithm for \( O(\Delta) \)-coloring [10]. Recently, this recursive technique was extended to (deg + 1)-list coloring, giving an algorithm with runtime \( (2^{O(\sqrt{\log n})} \log n) [37] \); this runtime has hidden \( \Delta \)-dependence on the color space. While the algorithms from [9, 10, 37] have an inherent \( O(\log n) \)-factor in their runtime, it was shown in [8] that one can decompose a graph into small arboricity subgraphs [5] (equipped with a small outdegree orientation) without incurring a log \( n \) factor, yielding the first

\[ 5 \text{ This has been applied for different parameter settings, where the graph is decomposed into } t = O(\Delta^\alpha) \text{ subgraphs of arboricity } O(\Delta/t), \text{ for a constant } 0 < \alpha < 1. \]
sublinear in $\Delta$ algorithm for $\Delta + 1$ coloring. The runtime for computing the underlying decompositions (as well as the conceptual complexity of the algorithm) were later improved in [12,42]. This immediately improves the results of [23] and [8], as well as gives the current form of our result.

Note that our results, as well as those of [23] and [8], only require partitioning into subgraphs of appropriately bounded outdegree, and are oblivious to their arboricity. While a bound on outdegree provides a similar bound on arboricity, computing a bounded outdegree orientation given a graph with bounded arboricity requires $\Omega(\log n)$ rounds [9].

Recent randomized coloring algorithms rely on the graph shattering technique [14]. In the shattering phase, a randomized algorithm computes a partial coloring of the graph, after which every uncolored connected component of the graph has small size (i.e., poly log $n$). In the post-shattering phase, deterministic $(\deg+1)$-list coloring is applied on all uncolored components in parallel. The runtime of the shattering phase has progressed from $O(\log \Delta)$ [14], over $O(\sqrt{\log \Delta})$ [33] to $O(\log^{\ast} \Delta)$ [20]. Combined with the poly log $n$-round list coloring algorithm of [47], this gives the current best runtime poly log log $n$, for $(\Delta + 1)$-coloring [20], and $O(\Delta \log n)$ for $(\deg + 1)$-list coloring [14].

While special graph classes are out of the scope of this paper, we mention the extensively studied case of distributed edge coloring. Here, poly log $n$-round algorithms were designed for progressively improving number of colors, from $(2 + \varepsilon)\Delta$ [28,31] to $(2\Delta - 1)$ [22,32], then to $(1 + \varepsilon)\Delta$ [30,32,48]. In a very recent manuscript, Bernshteyn gives a deterministic $(\Delta + 1)$-edge coloring algorithm with runtime polynomial in $\Delta$ and $\log n$ [15]. This matches the bound given by Vizing’s theorem [50]. The truly local complexity of $(2\Delta - 1)$-edge coloring has improved from $O(\Delta)$ [44] to $2^{O(\sqrt{\log \Delta})}$ [17] then to quasi poly log $\Delta$ [6] (in addition to $O(\log^{\ast} n)$).

Little is known on coloring lower bounds (in contrast to other symmetry breaking problems, e.g., maximal matching, MIS or ruling sets [3,4,38]). Linial’s $\Omega(\log^{\ast} n)$ lower bound is extended to randomized algorithms in [43]. The deterministic bound has recently been re-proven in a topological framework [24]. A lower bound of $\Omega(\Delta^{1/3})$ for $O(\Delta)$-coloring holds in a weak variant of the LOCAL model [34]. Several works characterized coloring algorithms which can only spend a single communication round [34,39,49]. None of these results gives anything non-trivial for two rounds. The speedup technique (e.g., [2-4,16-18]), which has been very successful for MIS lower bounds, is poorly understood for graph coloring. We briefly discuss the technique and its relation to our result in Sect. 6. There are lower bounds for more restricted variants of coloring. There is a $\Omega(\log n)$ ($\Omega(\log \log n)$) lower bound for deterministic [17] (randomized [19]) $\Delta$-coloring, as well as for $(\Delta - 2)$-defective 2-coloring [5]. An $\Omega(\log n/\log \log n)$ lower bound is known for greedy coloring [25]. Similar bounds hold for coloring trees and bounded arboricity graphs with significantly fewer than $\Delta$ colors [9,41].

### 1.4 Roadmap

Section 2.1 introduces our version of conflict coloring together with the 0-round solvability lemma. Section 2.2 defines the problems $P_0$ and $P_1$ and provides further notation. Section 2.3 contains the first result of our framework: an alternative proof of Linial’s algorithm. Theorem 2 (Linial for Lists) is proved in Sect. 3. Theorem 3 ((deg + 1)-list coloring) is proved in Sect. 4. Theorem 6 (Defective list coloring) is proved in Sect. 5. We conclude with a discussion of the results and open problems in Sect. 6.

### 2 Basic setup and Linial’s color reduction

In this section, we first introduce the conflict coloring framework that is the basis of our algorithm, then we show how it quickly implies an alternative variant of Linial’s color reduction algorithm. For a set $S$ and an integer $k \geq 0$, let $P(S)$ and $\binom{S}{2}$ denote the set of all subsets and all size-$k$ subsets of $S$, respectively. For a map $f$ we use $f^{(i)}$ to denote the $i$-fold application of $f$, e.g., $P^{(2)}(S) = P(P(S))$.

#### 2.1 Global conflict coloring

A list family $F \subseteq P(C)$ is a set of subsets of a color space $C$. Given a symmetric conflict relation $\mathcal{R} \subseteq \{(c, c') : c, c' \in C\}$, the conflict degree of a family $\mathcal{F}$ in $\mathcal{R}$ is the maximum number of colors in a list $L$ that conflict with a single color in a list $L'$ (possibly same as $L$), i.e., $d_{\mathcal{R}}(\mathcal{F}) = \max_{L,L' \in \mathcal{F},c \in L} |\{(c', \in L' : (c, c') \in \mathcal{R}\}|. An instance $\Psi = (C, \mathcal{R}, \mathcal{F}, \mathcal{L})$ of the global conflict coloring problem on the graph $G$ is given by a color space $C$, a symmetric conflict relation $\mathcal{R}$ on $C$, a list family $\mathcal{F}$, and an assignment $\mathcal{L} : V \rightarrow \mathcal{F}$ of lists $\mathcal{L}(v) \in \mathcal{F}$ of colors to each vertex $v$. The goal is to assign each vertex a color from its list such that no pair of neighboring vertices get conflicting colors $\{c, c' \in \mathcal{R}\}$. The conflict degree of $\Psi$ is $d_{\mathcal{R}}(\mathcal{F})$. Note that the conflict degree does not depend on $G$ or $\mathcal{L}$.

**Lemma 1** (Zero Round Solution) An instance $(C, \mathcal{R}, \mathcal{F}, \mathcal{L})$ of the conflict coloring problem on a graph $G$ can be solved without communication if $G$ is $m$-colored, $m, \mathcal{R}, \mathcal{F}$ are $\mathcal{R}$-

---

6 Formally, $G$ is also part of the problem, but we omit it since it is always clear from the context. These definitions crucially differ from local conflict coloring in [23], where a pair of colors can conflict along one edge and not conflict along another.
globally known, and every list in \( \mathcal{F} \) has size greater than 
\((m \cdot |\mathcal{F}| - 1) \cdot d_{\mathcal{R}}(\mathcal{F})\).

**Proof**
Every vertex \( v \) has a type \((\psi_v, \mathcal{L}(v)) \in [m] \times \mathcal{F}\), which is uniquely determined by its input color \( \psi_v \) and list \( \mathcal{L}(v) \). Note that adjacent vertices have distinct types, and there are \( t = m \cdot |\mathcal{F}| \) (globally known) possible types. Below, we show how to greedily assign each type a color from its list s.t. different types get non-conflicting colors. The conflict coloring problem is then solved by running this algorithm locally and consequently by all vertices, where each vertex gets the color assigned to its type.

Let \( \{T_i = (m_i, L_i)\}_{i=1}^t \) be a fixed ordering of \([m] \times \mathcal{F}\). Assign \( T_1 \) a color \( \phi(T_1) \in L_1 \) arbitrarily. For any \( i \geq 1 \), given the colors \( \phi(T_1), \ldots, \phi(T_i) \) of preceding types, assign \( T_{i+1} \) a color from \( L_{i+1} \) that does not conflict with \( \phi(T_1), \ldots, \phi(T_i) \). This can be done since each of the \( i \) fixed colors conflicts with at most \( d_{\mathcal{R}}(\mathcal{F}) \) colors in \( L_{i+1} \), i.e., there are at most \( i \cdot d_{\mathcal{R}}(\mathcal{F}) \leq (m|\mathcal{F}| - 1)d_{\mathcal{R}}(\mathcal{F}) \) colors that \( T_{i+1} \) cannot take, and this is less than the size of \( L_{i+1} \), as assumed. \( \square \)

### 2.2 Basic problems: \( P_0 \) and \( P_1 \)

Let \( \mathcal{C} \) be a fixed and globally known color space (which may depend on the graph \( G \)). An \( i \)-list is a subset \( L \subseteq \mathcal{P}^{(i)}(\mathcal{C}) \); e.g., the initial color list \( L_v \subseteq \mathcal{C} \) of a vertex \( v \) is a 0-list. Below, we introduce two problems. Problem \( P_0 \) is the standard list coloring problem, which we would like to solve via Lemma 1. However, the Lemma may not apply, if the lists \( L_v \) are not large enough. We then introduce problem \( P_1 \), with parameters \( 0 < \tau \leq k \), which is a low intersecting sublist selection problem. On the one hand, \( P_1 \) can be reformulated as a conflict coloring problem with larger lists and color space (hence could be solvable via Lemma 1), and on the other hand, a solution to \( P_1 \) can be used to solve \( P_0 \). The input of a node \( v \) in both problems contains its list \( L_v \). Formally, we have, for parameters \( \tau \) and \( k \),

- **\( P_0 \) (list coloring):** Node \( v \) has to output a color \( c(v) \in L_v \) such that adjacent nodes’ colors do not conflict, i.e., they are not equal.

- **\( P_1 \) (low intersecting sublists):** Node \( v \) has to output a 0-list \( C_v \subseteq L_v \) such that \( |C_v| = k \) and adjacent nodes’ 0-lists do not \( \tau \)-conflict. Two 0-lists \( C, C' \subseteq \mathcal{C} \) do \( \tau \)-conflict if \( |C \cap C'| \geq \tau \).

Note that problems \( P_0 \) and \( P_1 \) are not conflict coloring problems in the formal sense defined above (e.g., we do not define a list family \( \mathcal{F} \) or a list assignment \( \mathcal{L} : V \to \mathcal{F} \)). The aim with such definitions is to have a higher level and more intuitive (but still formal) problem statement. As the name suggests, in \( P_1 \) each node needs to compute a subset of its list such that the outputs form a low intersecting set family. \( P_1 \) can be reduced to a formal conflict coloring problem \( \mathcal{P}_1 \) whose solution immediately solves the \( P_1 \) instance (see Theorem 4).

#### 2.3 Warmup: Linial’s color reduction (without lists)

As a demonstration, we use the introduced framework to re-prove Linial’s color reduction theorem [41, Thm. 4.1] (which was extended to directed graphs in [9]). An \( r \)-cover-free family of size \( k \) over a set \( U \) is a collection of \( k \) subsets \( C_1, \ldots , C_k \subseteq U \) such that no set \( C_i \) is a subset of the union of \( r \) others. The obtained algorithm is essentially a greedy construction (via Lemma 1) of an \( r \)-cover-free family (with appropriate parameters) whose existence was proved via the probabilistic method in [41]. This greedy construction was first obtained in [35] but, to our knowledge, remained unnoticed in the distributed computing community.

While our aim is to make the proof below reusable for the later sections (hence the general statement in list coloring terms), we note that similar ideas can be used to obtain a less technical proof of Linial’s color reduction (see “Appendix A”).

**Theorem 4** ([9,41]) Let the graph \( G \) be \( m \)-colored and oriented, with max outdegree \( \beta \). All nodes have an identical color list \( L_v = \mathcal{C} \) from a color space \( \mathcal{C} \). Further, \( \mathcal{C} \), \( m \), and \( \beta \) are globally known. If \( |L_v| = l_0 \geq 2e \cdot \beta^2 \cdot \log m \) holds then in 1 round in LOCAL, each node can output a color from its list \( l_0 \)-s.t. adjacent nodes output distinct colors.

**Proof** Our goal is to solve \( P_0 \) with the given lists. To this end, it suffices to solve \( P_1 \) with parameters \( \tau = \lceil \log m \rceil > 1 \) and \( k = \beta \cdot \tau \), without communication. Indeed, having selected the sublists \( (C_v)_{v \in V} \), we need only one round to solve \( P_0 \): Node \( v \) learns sublists of its outneighbors and outputs any color \( c(v) \in C_v \setminus \cup_{u \in \text{Nout}(v)} C_u \). This can be done since for any outneighbor \( u \), \( C_v \) and \( C_u \) do not \( \tau \)-conflict (\( |C_v \cap C_u| \leq |C_v| - 1 \)) and hence \( |C_v \cap C_u| \geq k - (\tau - 1) \beta > 0 \).

We recast the given \( P_1 \) instance with (identical) input lists \((L_v)_{v \in V}\) as a conflict coloring problem \( \mathcal{P}_1 = (\mathcal{G}, \mathcal{R}_1, \mathcal{F}_1, \mathcal{L}_1) \) with color space \( \mathcal{C}_1 = \mathcal{P}(\mathcal{C}) \) and the \( \tau \)-color relation as \( \mathcal{R}_1 \). The list of a node \( \mathcal{L}_1(v) = (L_v^i)_{i \leq k} \) in \( \mathcal{P}_1 \) consists of all \( k \)-sized subsets of its input list \( L_v \). As each \( L_v \) is identical to \( \mathcal{C} \), we have that \( \mathcal{L}_1 \) maps each node \( v \) to the same list \( (L_v^i)_{i \leq k} = (\mathcal{C}_1^i) \) and the list family \( \mathcal{F}_1 = \{ (\mathcal{C}_1^i) \} \) consists of that singleton. A solution to \( \mathcal{P}_1 \) immediately solves \( P_1 \).

**Claim 1** The conflict degree of \( \mathcal{P}_1 \) is upper bounded by \( d_1 = \binom{k}{\tau} \cdot \binom{\lfloor \frac{\log m}{\tau} \rfloor - 1}{\tau - 1} \).

**Proof** Consider two arbitrary lists \( L, L' \in \mathcal{F}_1 \) and some 0-list \( C \in L \) (that is of size \( k \)). Each 0-list \( C' \in L' \) that \( \tau \)-conflicts with \( C \) can be constructed by first choosing \( \tau \) elements of \( C \), and then adding \( k - \tau \) elements from the rest of \( \mathcal{C} \) which is of size \( l_0 - \tau \). This can be done in at most \( d_1 \) many ways. \( \square \)
Claim 2 Let \( l_1 = \binom{0}{k} \). For any \( k \geq \tau > 1 \), if \( l_0 \geq 2ek^2/\tau \) then \( l_1/d_1 > 2^\tau \).

Proof We have
\[
\frac{l_1}{d_1} = \binom{0}{k} \left( \frac{l_0}{\binom{0}{k}} \right)^\tau \cdot \left( \frac{\tau}{ek^2} \right) \geq \frac{l_0\tau}{ek^2} \geq 2^\tau,
\]
where in the first inequality, we used the well-known approximation \( \binom{L}{k} \leq (ek/\tau)^k \), and the following inequality, applied to \( \binom{0}{k} / \binom{0}{k-1} \); for integers \( L > K > x > 0 \),
\[
\binom{L}{K} \frac{(L-x)}{(K-x)} = \frac{L!(K-x)!(L-K)!}{K!(L-K)!(L-x)!} = \frac{L(L-1)\ldots(L-x+1)}{K(K-1)\ldots(K-x+1)} > \left( \frac{L}{K} \right)^x,
\]
which follows as \( (L-i)/(K-i) > L/K \) holds for \( 0 < i \leq x \) \( \Box \).

Since \( \tau \geq \lfloor \log m \rfloor \) and \( |\mathcal{F}| = 1 \) the last claim implies \( |L_i(v)| = l_1 \geq m|\mathcal{F}|d_1/F_{i}(\mathcal{F}_i) \), hence we can solve \( P_1 \) (and thus also \( P_1 \)) without communication, using Lemma 1 \( \Box \).

What we did above is greedily forming a \( \Delta \)-cover-free family \( C_1, \ldots, C_m \subseteq \{O(\Delta^2 \log m)\} \) of size \( m \). This was done in [41], using the probabilistic method. Having such a family globally known, every vertex of input color \( x \) picks, in \( 0 \) rounds, the set \( C_x \) as its candidate output colors (neighboring vertices get distinct sets). Then, every vertex of color \( x \) learns the sets \( C \) of its neighbors, and based on the \( \Delta \)-cover-free property and the fact that there are at most \( \Delta \) neighbors, can select a color \( c \in C_x \) that is not a candidate for any neighbor.

3 Linial for lists

The goal of this section is to prove the following theorem.

Theorem 2 (Linial for Lists). In a directed graph with max. degree \( \Delta \), max. outdegree \( \beta \), and an input \( m \)-coloring, list coloring with lists \( L_v \) from a color space \( C \) and of size
\[
|L_v| \geq l_0 = 4e\beta^2(4\log \beta + \log \log |C| + \log \log m + 8)
\]
can be solved in 2 rounds in LOCAL. Each node sends \( l_0 \cdot \log \lvert C \rvert \) and \( \lceil \log \lvert C \rvert \rceil \) bits in the first round and \( \lceil l_0/4e\beta^2 \rceil \) bits in the second.

Assumption Throughout this section, we assume that the list of each node is exactly of size \( l_0 \); if a node has a list of size \( l > l_0 \) then it removes \( l - l_0 \) arbitrary colors from its list.

3.1 Problem \( P_2 \) (low intersecting sublist systems)

Recall that when applying our framework to the case where all lists were equal (Theorem 4), we essentially constructed a \( \Delta \)-cover-free family over the color space, and this was sufficient because we only needed a family of size \( m \); one set for each possible input pair \((x, L)\) of a color \( x \) and list \( L \), with the same \( L \) for all nodes. In order to replicate the construction for list coloring, we would need to construct a cover-free family with a set \( C_{x,L} \) for every combination of color \( x \) and list \( L \), and such that \( C_{x,L} \subseteq L \). It is not hard to see that such a family does not exist.\(^7\) Instead, we introduce problem \( P_2 \), whose goal is to assign every input \((x, L)\) a collection of candidate subsets \( K_{x,L} = \{C_{x,L,1}, C_{x,L,2}, \ldots \} \), where each \( C_{x,L,i} \subseteq L \). Further, we need that for every pair of distinct collections, there are not many pairs of subsets from the two collections that intersect much (in a sense formally defined below). This ensures that having such \( K_{x,L} \), the nodes can compute the desired \( \Delta \)-cover free family with one communication round, and use it to choose a color in another round.

Problem \( P_2 \) depends on parameters \( 0 < \tau \leq k \leq l_0 \) and \( 0 < \tau' \leq k' \), and each node has a list \( L_v \subseteq C \) in its input. Instead of a color (as in \( P_0 \)) or a sublist (as in \( P_1 \)), each vertex \( v \) now needs to output a collection \( K_v = \{C_1, C_2, \ldots \} \) of sublists of \( L_v \), each of size \( k \).

\( P_2 \) (low intersecting sublist systems): Node \( v \) has to output a 1-list \( K_v \subseteq \mathcal{P}(L_v) \) s.t. adjacent nodes’ 1-lists do not \((\tau',\tau)\)-conflict and \( |K_v| = k' \) and \( |C| = k \) for all \( C \in K_v \).

Two 1-lists \( K, K' \subseteq \mathcal{P}(C) \) do \((\tau',\tau)\)-conflict if there are two sequences \( C_1, \ldots, C_{\tau'} \in K \) and \( C_1', \ldots, C_{\tau'}' \in K' \), where at least one of the sequences has \( \tau' \) distinct elements, and for every \( 1 \leq i \leq \tau' \), \( C_i \) and \( C_i' \) \( \tau \)-conflict.

We prove in Lemma 2 that with a suitable choice of the parameters a solution of \( P_2(\tau, k, \tau', k') \) yields a solution of \( P_1(\tau, k) \) and \( P_0 \), where we implicitly impose (and throughout this section assume) that \( P_0 \), \( P_1 \) and \( P_2 \) receive the same input lists \( (L_v)_{v \in V} \).

3.2 Algorithm

Under the assumptions of Theorem 2, fix the following (globally known) four parameters:

\( ^7 \) Consider lists of size \( \ell \) from a colorspace of size \( C \geq 10\ell \). All members of such a cover-free family must be distinct and have size at most \( \ell \), so the family cannot be larger than \( \sum_{i \leq \ell} \binom{C}{i} < 2^{\binom{C}{2}} \). This is less than the size \( m \binom{C}{2} \) we need.

\( ^8 \) The sets in each sequence are not necessarily distinct.
\[ \tau = [8 \log \beta + 2 \log \log |C| + 2 \log \log m] + 14 \]
\[ \tau' = 2^{\tau - \lfloor \log (2e \beta^2) \rfloor} \]
\[ k = \beta \cdot \tau \]
\[ k' = \beta \cdot \tau'. \]

Note that, \( \tau', k, k' \) are determined by \( \tau \) and \( \beta \). \(^9\) We have the bound \( l_0 \geq 2ek^2/\tau \) on list size.

The algorithm consists of two phases. In the first phase, nodes locally and without any communication compute a solution \((K_v)_{v \in V}\) of \( P_2 \) consisting of 1-lists (see Lemma 3).

The second phase has two rounds of communication. In the first round, each node \( v \) learns the solution \( K_u \) to \( P_2 \) of each outneighbor \( u \in N_{out}(v) \), and selects a 0-list \( C_v \in K_v \) that does not \( \tau \)-conflict with the 0-lists in \( K_u \) for \( u \in N_{out}(v) \), and thus is a solution to \( P_1 \) (Lemma 2). In the second round, node \( v \) learns the lists \( C_u \) of outneighbors, and selects a color \( c(v) \in C_v \) that does not appear in \( C_u \), for \( u \in N_{out}(v) \) (Lemma 2). This solves \( P_0 \).

**Lemma 2** \( (P_2 \rightarrow P_1 \rightarrow P_0) \) Given a solution \((K_v)_{v \in V}\) of \( P_2 \) (a solution \((C_v)_{v \in V}\) of \( P_1 \)), a solution of \( P_1 \) (of \( P_0 \), resp.) can be computed in one round.

**Proof** \( P_2 \rightarrow P_1 \): As \( K_v \) and \( K_u \) do not \((\tau', \tau)\)-conflict for any \( u \in N_{out}(v) \), there are at most \( \tau' - 1 \) 0-lists \( C \in K_v \) that \( \tau \)-conflict with a 0-list in \( K_u \). By removing all \( C \) from \( K_v \) that \( \tau \)-conflict with any \( C' \in K_u \) for any outneighbor \( u \in N_{out}(v) \), at most \( |K_v| - \beta \cdot (\tau' - 1) = k' - \beta \cdot (\tau' - 1) \geq 1 \) outputs remain; let \( C_v \) be any such 0-list. As the conflict relation is symmetric, \( P_1 \) is solved.

\( P_1 \rightarrow P_0 \): Since \( C_v, C_u \) do not \( \tau \)-conflict, removing from \( C_v \) all the colors from the 0-lists of the outneighbors leaves at least \( k - \beta \cdot (\tau - 1) \geq 1 \) colors that \( v \) can select as \( c(v) \).

\( \square \)

### 3.3 Zero round solution to \( P_2 \)

The results in this section hold for parameters \( \tau, \tau', k' \) fixed as in Sect. 3.2, and for any \( \tau \leq k \leq \beta \tau \). While we set \( k = \beta \tau \) for solving \( P_0 \), we will use another value of \( k \) for our defective coloring result (see Sect. 5). Note that we still have the bound \( l_0 \geq 2ek^2/\tau \) on list size, for any such \( k \). The goal of this section is to prove the following lemma.

**Lemma 3** \( (P_2 \text{ in zero rounds}) \) Under the assumptions of Theorem 2, the problem \( P_2(\tau, k, \tau', k') \) can be solved in zero rounds.

To prove Lemma 3, we reduce (without communication) an instance of \( P_2 \) to a conflict coloring instance \( \mathcal{P}_2 \) that can be solved in zero rounds with Lemma 1.

---

\(^9\) It may also be helpful to note the similarity between this parameter setting and that in Theorem 4.

---

Reducing \( P_2 \) to a conflict coloring instance (no communication): Given input lists \((L_v)_{v \in V}\) and parameters \( 0 < \tau \leq k \leq l_0 \) and \( 0 < \tau' \leq k \), the conflict coloring instance \( \mathcal{P}_2 \) is given by the colorspace \( \mathcal{P}^{(2)}(C) \), the \((\tau', \tau)\)-conflict relation \( R_2 \) on 1-lists, the list family \( \mathcal{F}_2 = \{ \overline{L}_2(L) \mid L \in \binom{C}{l_0} \} \) and list \( L(v) \in L_2(L_v) \) for node \( v \), where \( L_2 : \binom{C}{l_0} \rightarrow \mathcal{P}^{(3)}(C) \) maps \( l_0 \)-sized subsets of \( C \) to 2-lists and is defined below. The map \( L_2 \), the colorspace, the conflict relation and the set family \( \mathcal{F}_2 \) are global knowledge and no communication is needed to compute the list \( L(v) \) of a node in \( \mathcal{P}_2 \).

To define the map \( L_2 \) we need another definition. For an integer \( t \geq 0 \) and a 2-list \( T \), a 1-list \( K \in T \) is \((t, \tau')\)-good if there are less than \( t \) 1-lists \( K' \in T \) such that \( K \) and \( K' \) do \((\tau', \tau)\)-conflict. We define maps \( L_1, L_2, L_2 \), as follows. For \( S \in \binom{C}{l_0} \),

\[ L_1(S) = \binom{S}{k} \quad \text{(elements} C \text{ are 0-lists)} \]
\[ L_2(S) = \binom{L_1(S)}{k'} \quad \text{(elements} K \text{ are 1-lists)} \]
\[ L_2(S) = \{ K \in L_2(S) \mid K \text{ is } (L_2(S), d_2, \tau', \tau) \text{-good} \}, \]

where \( d_2 \) is chosen as in Lemma 5. \(^{10}\) Due to the definition of the \((\tau', \tau)\)-conflict relation and the map \( L_2 \), solving \( \mathcal{P}_2 \) immediately solves \( P_2 \).

The sizes of \( L_1(S), L_2(S) \) and \( L_2(S) \) do not depend on \( S \). Let \( l_1 = |L_1(S)| = \binom{l_0}{k} \), and \( l_2 = |L_2(S)|/2 = \binom{l_0}{k}/2 \).

We will later show that \( |L_2(S)| \geq l_2 \). Let \( \mathcal{F}_1 = \{ L_1(S) \mid S \in \binom{C}{l_0} \} \).

**Some intuition:** In the conflict coloring instance \( \mathcal{P}_2 \), every node \( v \) has a list \( \{ K_1, K_2, \ldots \} \) of 1-lists, each a collection of subsets of its input list \( L_v \). To ensure small conflict degree, but still large list size, it is enough that \( v \) only takes \( K \)'s that are “good”, as defined above. Since being “good” only depends on \( L_v \), node \( v \) can also compute its \( \mathcal{P}_2 \)-list locally.

In Lemmas 4, 5, 6, we show that lists \( L_2(L_v) \) are large and that \( \mathcal{P}_2 \) has small conflict degree. Before that, let us see how these lemmas imply 0-round solvability of \( P_2 \) (Lemma 3).

**Proof (of Lemma 3)** To solve an instance of \( P_2 \) on input lists \((L_v)_{v \in V}\), nodes locally set up the conflict coloring instance \( \mathcal{P}_2 \). Lemmas 4 and 5 show that the conflict degree of \( \mathcal{P}_2 \) is bounded by \( d_2, \mathcal{F}_2 \leq d_2 \), and that every list in \( \mathcal{F}_2 \) has size at least \( l_2 \). Note that \( \mathcal{F}_2 \) is globally known and \( |\mathcal{F}_2| = \binom{C}{l_0} < |C|^{l_0} \), since each element in \( \mathcal{F} \) can be written as \( L_2(S) \) for some \( S \in \binom{C}{l_0} \). Using Lemma 6 we obtain

\(^{10}\) The precise value is not important to understand how \( L_2 \) is formed.
\[ \frac{l_2}{d_2} \geq \frac{1}{8} 2^{s - \log(4e\beta^2)} \geq m \cdot \abs{\mathcal{C}}^{\ell_0} > m \cdot \abs{\mathcal{F}_2}, \] where the second inequality follows by a routine calculation (following hereafter) using the definition of \( \tau \) and \( \ell_0 \). Thus, Lemma 1 holds, and \( \mathcal{Q}_2 \) and \( \mathcal{P}_2 \) can be solved in zero rounds.

Next, we perform the missing calculation. We use \( \tau = [8 \log \beta + 2 \log \log \abs{\mathcal{C}} + 2 \log \log m] + 14 \), and \( l_0 = 2e\beta^2(\tau + 1) \), to deduce that \( (1/8) \cdot 2^{s - \log(4e\beta^2)} \geq m \abs{\mathcal{C}}^{\ell_0} \) holds. We have (explanation below):

\[
\log \left( 8m \cdot \abs{\mathcal{C}}^{\ell_0} \right) \\ \leq \log \log m + \log \log C + \log l_0 + \log 3 \\ \leq \log m + \log \log C + \log 6e + 2 \log \beta + \log(\tau + 1) \\ \leq \frac{\tau}{2} + \log \log m + \log 6e + 2 \log \beta + \log \log C \\ \leq \tau/2 + \tau/2 - \log(4e\beta^2) = \tau - \log(4e\beta^2),
\]

where in the first inequality, we used \( \log(x + y) \leq \log x + \log y \), for \( x, y \geq 1 \), in the second one we used \( l_0 = 2e\beta^2(\tau + 1) \), in the third we used \( \log(\tau + 1) \leq \tau/2 \) (holds for \( \tau \geq 6 \)), while in the last one we used \( \tau/2 \geq 4 \log \beta + \log \log |C| + \log \log m + 7 \).

Exponentiating (with base 2) both sides twice gives us the claim. \( \square \)

We continue with proving Lemmas 4, 5, 6. First, we bound the conflict degree of \( \mathcal{Q}_2 \) by \( d_2 \) (from the definition of \( L_2 \)). Note that it is a property of the list family \( \mathcal{F}_2 \) and the conflict relation \( \mathcal{R}_2 \), and is independent of the graph and list assignment. From the definition of \( L_2 \), the value \( d_2 \) can be seen as a bound on the number of 1-lists in \( L_2(X) \) that conflict with a given 1-list \( K \) in \( L_2(X) \). The intuition behind the lemma is that \( L_2(Y) \), for \( Y \neq X \), should have fewer conflicts with \( K \), since \( Y \) can be obtained from \( X \) by removing some elements and adding new ones (which do not appear in \( K \)). The proof involves establishing an isomorphism between \( L_2(X) \) and \( L_2(Y) \), which preserves their common elements. For this, it is crucial to have \( |X| = |Y| \). This is why we need all input lists to have same size \( |L_v| = l_0 \).

**Lemma 4** (Conflict Degrees) Let \( X, Y \in \binom{\mathcal{C}}{\ell_0} \) be 0-lists. Let \( d_1 = \binom{\ell_0}{k} \cdot \binom{l_0 - \ell_0}{k - \ell_0 - 1} \).

1. For any 0-list \( C \in L_1(X) \), there are at most \( d_1 \) 0-lists in \( L_1(Y) \) that \( \tau \)-conflict with \( C \).
2. For any 1-list \( K \in L_2(X) \), there are at most \( d_2 \) 1-lists in \( L_2(Y) \) that \( (\tau', \tau') \)-conflict with \( K \). In particular, \( d_{2c}(\mathcal{F}_2) \leq d_2 \), and this holds irrespective of the value of \( d_2 \).

**Proof** The proof of the first claim is along the same lines as the proof of Claim 1, so we only prove the second claim here. Let \( X_1 = L_1(X), X_2 = L_2(X) \) and \( X_2 = L_2(X) \), and define \( Y_1, Y_2, \tilde{Y}_2 \) similarly. As \( |X| = |Y| \), there is a bijection between \( X \) and \( Y \) that is the identity on \( X \cap Y \). Let \( \alpha : X \rightarrow Y \) be an arbitrary such bijection; if \( c \in X \cap Y \) then \( \alpha(c) = c \). Further, since \( X_1 = \binom{\mathcal{C}}{\ell_0} \) and \( Y_1 = \binom{\mathcal{C}}{\ell_0} \), we have the bijection \( \beta : X_1 \rightarrow Y_1 \) given by \( \beta(|c_1, \ldots, c_k|) = |\alpha(c_1), \ldots, \alpha(c_k)| \), and since \( X_2 = \binom{\mathcal{C}}{\ell_0} \) and \( Y_2 = \binom{\mathcal{C}}{\ell_0} \), we have the bijection \( \gamma : X_2 \rightarrow Y_2 \), where \( \gamma(|c_1, \ldots, c_k|) = \beta(c_1, \ldots, \beta(c_k)) \).

We show that the claim holds for any \( t \geq 0 \) and for any \( K \in X_2 \) that is \( (X_2, t, \tau', \tau') \)-good (which demonstrates that the actual value of \( d_2 \) is irrelevant). As \( Y_2 \subseteq \tilde{Y}_2 \), it suffices to show that \( K \) does \( (\tau', \tau') \)-conflict with at most \( t \) 1-lists in \( \tilde{Y}_2 \). Towards a contradiction, let \( K \in \tilde{X}_2 \) \( (\tau', \tau') \)-conflict with each of \( t \) distinct 1-lists \( K'_1, K'_2, \ldots, K'_t \in \tilde{Y}_2 \) and define \( K_i = \gamma^{-1}(K'_i) \in \tilde{X}_2 \). We show that \( K \) also \( (\tau', \tau') \)-conflicts with each of the distinct \( \gamma \) (which is a bijection) \( K_1, \ldots, K_t \in \tilde{X}_2 \), which is a contradiction to \( K \) being \( (\tilde{X}_2, t, \tau', \tau') \)-good: To ease notation, let us focus on \( K \) and \( K_1 \). Assume that there are \( \tau' \) distinct (case 2: not necessarily distinct) 0-lists \( C'_1, C'_2, \ldots, C'_s \) in \( K_1 \), and \( \tau' \) not necessarily distinct (case 2: distinct) 0-lists \( C_1, C_2, \ldots, C_s \) in \( K \), such that \( C_i \) and \( C'_i \) \( \tau' \)-conflict. Then \( \beta^{-1}(C'_i) \) and \( C_i \) 1-conflict, since \( \alpha \) is the identity on \( C_i \cap C'_i \); \( \beta^{-1}(C'_i) \) are all distinct (since \( \beta \) is a bijection) and belong to \( K_1 \), therefore \( K \) and \( K_1 \) \( (\tau', \tau') \)-conflict. \( \square \)

Next, we show that at most half of the elements \( K \in \tilde{L}_2 \) fail to be good; this lemma crucially depends on the value of \( d_2 \). Below, we use the conflict degree \( d_1 \) from Lemma 4.

**Lemma 5** \( (L_2 \) is large) Let \( d_2 = 4 \binom{k}{d'_1} \binom{l_0 - \ell_0}{k - \ell_0 - 1} \). For any \( S \subseteq \binom{\mathcal{C}}{\ell_0} \), we have \( \abs{L_2(S)} \geq l_2 \).

**Proof** Fix \( S \subseteq \binom{\mathcal{C}}{\ell_0} \) and consider the digraph \( H = (V_H, E_H) \) over the vertex set \( V_H = \tilde{L}_2(S) \), where \( (K, K') \in E_H \) if \( K \) contains at least \( \tau' \) lists, each in \( \tau' \)-conflict with a list in \( K' \) (in particular, for every \( K, (K, K) \in E_H \). Note that a 1-list \( K \) is \( (\tilde{L}_2(S), d_2, \tau', \tau') \)-good iff its undirected degree in \( H \) is at most \( d_2 \).

**Claim 3** The maximum outdegree of a node \( K \in V_H \) is at most \( d_2/4 \).

**Proof** Consider a fixed \( K \in V_H \). Let \( X \subseteq K \) be the set of 0-lists in \( L_1(S) \) that \( \tau' \)-conflict with a 0-list in \( K \). By Lemma 4 part 1, every \( C \in K \) \( \tau' \)-conflicts with at most \( d_1 \) 0-lists, hence \( \abs{X} \leq \abs{K} \cdot d_1 = k'd_1 \). Every 1-list \( K' \), such that there are at least \( \tau' \) 0-lists in \( K \) that are in \( \tau' \)-conflict with a 0-list in \( K' \), can be obtained by first choosing \( \tau' \) 0-lists from \( X \), and adding an arbitrary subset of \( \tilde{k' - \tau'} \) other 0-lists. Clearly, this can be done in at most \( \binom{\tilde{k'}}{\tilde{k' - \tau'}} \binom{l_0 - \ell_0}{k - \ell_0 - 1} = \frac{d_2}{4} \) many ways. \( \square \)

The Claim implies that \( \abs{E_H} \leq \abs{V_H} \cdot d_2/4 \), hence the undirected average degree of a node in \( H \) is at most
2|E_H|/|V_H| ≤ d_2/2$, and by Markov’s inequality, at most half of the nodes have degree greater than $d_2$. Since $L_2(S)$ is the set of nodes of degree at most $d_2$, we conclude that $|L_2(S)| ≥ |V_H|/2 = |L_2(S)|/2 = l_2$. 

Finally, we bound the ratio $l_2/d_2$ based on the values of the remaining parameters.

**Lemma 6** (I/d Ratio) If $k ≥ τ ≥ \log(2eβ^2)$, $l_0 ≥ 2ek^2/τ$, $τ' = 2τ−\log(2eβ^2)$, and $k' = βτ'$, then

$$l_2/d_2 > 2^τ−\log(4k^2β^2)/8.$$

**Proof** First, we get $l_1/d_1 ≥ τ'$, as in Eq. (1). Then, with $\binom{k^2d_1}{τ} ≤ \left(\frac{ek^2}{τ}\right)^τ$, and $\binom{l_1}{τ'}$ applied to $\binom{l_1}{k'−τ'}$, we lower bound $l_2/d_2$ as

$$\frac{l_2}{d_2} = \frac{1}{8} \frac{\binom{l_1}{τ'}}{\binom{k'−τ'}{τ'}} ≥ \frac{1}{8} \frac{l_1}{k'} · \frac{τ'}{e(k'd_1)}$$

$$≥ \frac{1}{8} \left(\frac{2τ}{eβ^2τ'}\right) ≥ \frac{2τ'}{8} ≥ \frac{2^τ−\log(4k^2β^2)}{8},$$

where the third and fourth inequalities hold since $τ' ≤ 2τ/2eβ^2 ≤ 2τ'$.

3.4 Proof of the main theorem

**Proof (Theorem 2)** Nodes solve $P_2$ without communication (Lemma 3), then use two rounds of communication to solve the input list coloring problem $P_0$ (see algorithm description and Lemma 2).

We bound the messages sent by a node $v$ during the algorithm. In the first round, $v$ needs to send $K_v$ to its neighbors. Note that $K_v$ is uniquely determined by the list $L_v$ and the input color $ψ_v$ (see the proof of Lemma 1), so it suffices to send $(ψ_v, L_v)$, which can be encoded in $l_0[\log |C|] + \log (|m|)$ bits. In the second round, $v$ needs to send $C_v$. Since $C_v ∈ K_v$, and the neighbors know $K_v$, it suffices to send the index of $C_v$ in $K_v$ (in a fixed ordering). Recall that $|K_v| = k' < 2^τ$, so $v$ only needs to send $τ ≤ l/4eβ^2$ bits.

**Remark 1** Note that in both communication rounds of Theorem 2 each node only needs to send messages to its in-neighbors. In contrast, the results in Sects. 4 and 5 require bi-directional communication.

4 Application: ($Δ + 1$)-coloring and (deg + 1)-List Coloring

**Theorem 3** (Restatement). In a graph with max. degree $Δ$, (deg + 1)-list coloring problem with lists $L_v ⊆ C$ from a color space of size $|C| = 2^{poly(Δ)}$ can be solved in $O(\sqrt{Δ} \log Δ) + \frac{1}{2} \cdot \log^* n$ rounds in LOCAL. Furthermore, each node only needs to broadcast to its neighbors a single non-CONGEST message consisting of a subset of its list.

The proof combines Theorem 2 with the graph partitioning provided by [12], following the high level description in Sect. 1. A variant of this framework was also used in [6,37]. We nevertheless present a proof for completeness, and also due to subtle but important differences from [23] (we have an additional finishing phase that is not present there).

The graph partitioning given by [12] aims at arbdefective colorings, as introduced in [10], but the main technical object provided by [12] (and which is all we need here) is a low outdegree partition of a graph. For a graph $H = (V, E)$, the collection $H_1, \ldots, H_4$ of directed graphs $H_i = (V_i, E_i)$ is a β-outdegree partition of $H$ if their vertices span $V$, i.e., $V = V_1 ∪ \ldots ∪ V_4$, the underlying undirected graph of $H_i$ is the induced subgraph $H[V_i]$ (so it is indeed a partition), and the max. outdegree of a node in $H_i$ is at most $β$, for all $i$.

**Lemma 7** (Lemmas 6.1-6.3, [12,42]) There are constants $c, c' > 0$, s.t. for every $β ≥ c$, given a graph $H$ with an $m$-coloring, there is a deterministic algorithm that computes a β-outdegree partition $H_1, \ldots, H_4$ with $k = c' Δ/β$ in $O(k + \log^* m)$ rounds in CONGEST.

**Proof (of Theorem 3)** We begin with computing an $m = O(Δ^2)$-coloring in $\frac{1}{4} \log^* n + O(1)$ rounds [41,49]. The main algorithm consists of $t = \log_2(Δ/Δ^4)$ phases. After phase $j$, we have colored a subset of vertices, s.t. the maximum degree $Δ_j$ of the graph $G[U_j]$ induced by uncolored vertices is upper bounded as $Δ_j ≤ Δ/2^j$. Before describing a phase $j$, let us show how we finish the coloring after the phase $t$, in a final phase. Consider the graph $G[U_t]$ at the end of phase $t$. Note that it has maximum degree $Δ_t = O(Δ^4/Δ)$. We compute an $m'$-coloring of $G[U_t]$ with $m' = O(Δ^2) = O(\sqrt{Δ})$ from the initial $m$-coloring in $O(1)$ rounds [41]. In each of the final $m'$ rounds $i = 1, \ldots, m'$, vertices with color $i$ pick a color from their list not picked by a neighbor (can be done since each node has more available colors than colored neighbors, and no two neighbors pick simultaneously). The runtime of the final phase is $O(\sqrt{Δ})$.

The following happens in phase $j = 1, \ldots, t$. At the beginning of the phase, we have the set $U_{j−1}$ of uncolored vertices, where $U_0 = V(G)$. Let $X = 4e \cdot (4log Δ + log log |C| + log log m + 8) = O(log Δ)$ and for $j = 0, \ldots, t − 1$ let $β = √j/(2X)$ and $k_j = c' Δ/β$, where $c'$ is the constant in Lemma 7. We partition $G[U_{j−1}]$ into $β_j$-outdegree subgraphs $H_1, H_2, \ldots, H_k$, using Lemma 7.

11 In order to apply the lemma, we need $β_j ≥ c$. Since $β_j ≥ β = ζ(√j/\log Δ)$, $β_j ≥ c$ holds if $Δ$ is large enough. For $Δ = O(1)$, Theorem 3 holds via an $O(Δ) + 1/2 \log^* n$ round algorithm (see e.g. [12]).
The phase consists of $k_j$ stages $i = 1, \ldots, k_j$, each consisting of 3 rounds. In stage $i$, we partially color $H_i$, as follows. For every uncolored vertex $v \in H_i$, let $L_{v,i,j}$ be the set of colors in $L_v$ that have not been taken by a neighbor of $v$. Let $W_i = \{ v \in H_i : |L_{v,i,j}| \geq \beta_j^2 X \}$. Color the graph $H_i[W_i]$ using Linial for Lists (Theorem 2) with color space $C$, the $\beta_j$-outdegree orientation and the $m$-coloring. This is a valid application of the theorem, by the definition of $X$, $\beta_j$ and $W_i$. In the third round of the stage, all nodes in $W_i$ send their color to their neighbors. This completes the algorithm description. Clearly, phase $j$ takes $3k_j$ rounds.

It remains to show that $\Delta_j \leq \Delta/2^j$. We do this by induction, with base $j = 0$, $\Delta_0 = \Delta$. Assume $\Delta_j \leq \Delta/2^j$ holds for some $j \geq 0$. Let $v \in U_j$ be a node that is uncolored at the end of stage $j$. We know that $|L_{v,j,i}| < \beta_j^2 X = \Delta_j/2$, in a stage $i$. Recall that $L_{v,j,i}$ is the set of colors in $L_v$ not taken by a neighbor of $v$. Since $|L_v|$ is larger than the number of neighbors of $v$, $|L_{v,j,i}|$ is larger than the number of uncolored neighbors of $v$. Therefore $v$ has at most $|L_{v,j,i}| < \Delta_j/2 \leq \Delta/2^{j+1}$ neighbors in $U_j$, which proves the induction: $\Delta_{j+1} \leq \Delta/2^{j+1}$.

Recall that $X = O(\log \Delta)$ and bound the runtime as follows:

$$\frac{1}{2} \log^* n + O(1) + \sum_{j=1}^{t} 3k_j + m' $$

$$= \frac{1}{2} \log^* n + \sum_{j=1}^{t} 3c j' \sqrt{X \Delta \over 2 j - 1} + O(\sqrt{\Delta})$$

$$= \frac{1}{2} \log^* n + O(\sqrt{\Delta \log \Delta}).$$

The second claim easily follows, recalling the message complexity of Linial for Lists.

Note that the final phase in the algorithm above is necessary as otherwise, if the recursion continued until the maximum degree of uncolored nodes was, say, $O(\log(3) \Delta)$, their reduced list size would be similarly small, and we could no longer apply Theorem 2, which requires lists of size $\Omega(\log \log |C|) = \Omega(\log \log \Delta)$, as the color space does not change in the recursion.

**Corollary 1** In a graph with max. degree $\Delta = \tilde{O}(\log n)$, $(\deg + 1)$-list coloring problem with lists $L_v \subseteq C$ from a color space of size $|C| = \text{poly}(\Delta)$ can be solved in $\tilde{O}(\sqrt{\Delta}) + \frac{1}{2} \log^* n$ rounds in CONGEST.

**Proof** In the algorithm of Theorem 3 each vertex only participates in a single instance of Linial for lists, and all other steps of the algorithm can be implemented in CONGEST. Thus, as stated above, the only non-CONGEST message $M$ of a node consists of a subset of its list. Note that we can always limit the lists to size $\Delta + 1$, so the number of colors in $M$ is at most $\Delta + 1$. Each color can be encoded in $O(\log |C|) = O(\log \Delta)$ bits. Thus, $M$ can be encoded in $O(\Delta \log \Delta)$ bits. By the assumption of the claim, we have $\Delta \leq a \log n \cdot (\log \log n)^b$, for constants $a \geq 1, b \geq 0$. It follows that $\Delta \log \Delta \leq a \log n \cdot (\log \Delta)^{b+1}$, as otherwise we would have $\Delta > a \log n (\log \Delta)^b > a \log n (\log \log n)^b$. Thus, each node sends at most $O(\Delta \log \Delta/\log n) = O((\log \Delta)^{b+1})$ messages more than in the (LOCAL) algorithm, and the runtime increases by the corresponding factor. $\square$

### 5 Defective (list) coloring

A $d$-defective $c$-coloring is a $c$-coloring where each vertex $v$ can have at most $d$ neighbors with the same color as $v$. A $d$-defective list coloring is a list coloring where each vertex $v$ can have at most $d$ neighbors with the same color as $v$. As proven in [36], one can compute a $d$-defective $O((\Delta/(d + 1))^2 \log m)$-coloring, given an $m$-coloring, in one round in CONGEST. As a warm up to our “list version”, we re-prove this result by adapting the proof of Theorem 4.

**Theorem 5** ([36]) Let $d \geq 0$ be an integer. In a graph with max. degree $\Delta > d$ and an input $m$-coloring, $d$-defective coloring problem with $2e \cdot [\Delta/(d + 1)]^2 \cdot \log m$ colors can be computed in one round in CONGEST.

**Proof** We adapt the proof of Theorem 4. Note that the proof holds when $G$ is not directed, and $\beta$ is replaced with $\Delta$. For a consistent notation, let $\beta = \Delta$. Also, let $P_{\beta}(a, b)$ denote problem $P_1$ under the parameter setting $\tau = a$ and $\kappa = b$. First, we solve problem $P_{\beta}(\tau, \kappa)$ with parameters $\tau = [\log m]$, $\kappa = [\beta/(d + 1)] \tau$. As Claim 2 holds for any $\log m \geq \tau \geq \log m$ and $l_0 \geq 2ek^2/\tau$, the corresponding $P_{\beta}(\tau, \kappa)$ can be solved locally with list size $l_0 \geq 2ek^2/\tau = 2e[\beta/(d + 1)]^2 \tau$, which holds by our theorem assumption. Let $(C_v)_{v \in V}$ be the solution to $P_1$, with $(C_v) = k$. Let $v$ be a node. Since the solution is conflict free, we have $(C_v) < \tau$, for every neighbor of $v$. For a color $c \in C_v$, let $f(c)$ be the number of neighbors $u$ s.t. $c \in C_u$. It follows that $\sum_{c \in C_v} f(c) < \beta \tau$, hence there is a color $c \in C_v$ with $f(c) < \beta \tau/|C_v| \leq d + 1$. Each node $v$ picks a color $c \in C_v$ with minimum $f(c)$. $\square$

Note that, as for Theorem 4, the set systems in the proof of Theorem 5 are computed greedily. The theorem also works with a $d_1$-defective $m$-coloring as input. The defect of the output coloring is then $d + d_1$. Applying this result iteratively and combining it with another set system construction based on polynomials over finite fields, one can compute a $d$-defective $O((\Delta/(d + 1))^2)$-coloring in $O(\log^* m)$ rounds from a given $m$-coloring [36], [13, Section 3.2].

**Theorem 6** (Defective List Coloring) Let $d \geq 0$ be an integer. In a graph with max. degree $\Delta > d$ and an input
m-coloring, d-defective list coloring problem with lists \( L_v \) from a color space \( C \) and of size \( |L_v| \geq 1 = 4e(\Delta/d + 1)^2 \cdot (4 \log \Delta + \log |C| + \log \log m + 8) \) can be solved in 2 rounds in LOCAL, if \( C, m \) and \( \Delta \) are globally known.

Proof As in Theorem 5, observe that our whole analysis of in 2 rounds in LOCAL, if \( \beta = \Delta \), as before. First, we solve problems \( P_2 \) and \( P_1 \) with parameters \( \tau, \tau', k' \) as in Sect. 3.3, and \( k = \lfloor \beta/(d + 1) \rfloor \tau \) (as observed in the beginning of Sect. 3.3, \( P_2 \) and \( P_1 \) can be solved for any \( \tau \leq k \leq \beta \tau \)). Thus, we require the list size of each node to be at least \( l_0 \geq 2ek^2/\tau \), which holds. Given a solution of \( P_1 \), we can obtain the \( d \)-defective coloring in the same manner as in Theorem 5. □

6 Discussion

We conclude with several observations on our results, as well as open problems.

1. It is possible to define problems \( P_3, \ldots, P_t \) for any \( t \), as we defined \( P_1 \) and \( P_2 \). Lemma 4 extends naturally to these problems, so the input of a node \( v \) in \( P_t \) is again only its initial list \( L_v \). We need \( t \) rounds, instead of 2, to derive a solution of \( P_0 \) from a solution of \( P_t \) (which also implies larger messages). On the other hand, we have somewhat smaller list size requirement: \( c\beta^2 (\log \beta + \log^0 |C| + \log^0 m) \), for a constant \( c > 0 \). In particular, one can list color in \( O(\log^* \max(|C|, m)) \) rounds if lists are at least \( c\beta^2 \log \beta \) for a sufficiently large constant \( c > 0 \).

2. Unlike in [23], our bound on the list size does not depend on \( \Delta \). This result implies that, e.g., given a graph with a \( \beta \)-outdegree orientation and an input coloring with \( 2^{\text{poly}(\beta)} \) colors and list sizes of at least \( c\beta^2 \cdot \log \beta \) from a color space of size \( 2^{\text{poly}(\beta)} \), for a constant \( c > 0 \), it is possible to list-color the graph in 2 rounds. By the remark above, one can have even larger color space, by increasing the runtime accordingly.

3. A lower bound in [49] suggests that the coloring in Theorem 2 cannot be done in a single round. In particular, if one is willing to keep the doubly-logarithmic dependence on \( m \) in the list size, then one has to pay a factor exponential in \( \beta \). On the other hand, we do not know how to eliminate the \( \log \beta \) term, even if we use more communication.

4. The recently popular speedup technique has mostly been used to prove lower bounds, e.g., [2–4,16–18]. Here, a problem \( P_0 \) is mechanically (and without communication!) transformed into a problem \( P_1 \) whose complexity is exactly one round less. Then, if \( P_1 \) cannot be solved locally one deduces that \( P_0 \) cannot be solved in 1 round. By iterating this process, one can derive larger lower bounds. However, the description complexity of derived problems grows exponentially, and it is very important to be able to simplify the problem description, in order to iterate the process. If \( P_0 \) is the \( (\Delta + 1) \)-vertex coloring problem, this process has only been understood in the special case of \( \Delta = 2 \), which corresponds to Linial’s \( \Omega(\log^* n) \) lower bound [40,41]. While [23] also performs a similar transformation, it is different from the speedup technique, since the transformation is not mechanical, requiring nodes to communicate for building the new problems. It may rather be seen as a transformation of problem instances (that depend on the graph) than problems. In contrast, our transformations are mechanical, and the input and output labels live in the same universe as it is the case for mechanical speedup.

5. While our present treatment of the proof of Theorem 2 in terms of conflict coloring problems and problems \( P_0, P_1 \) and \( P_2 \) has the aim of connecting to the framework of [23] as well as to the speedup framework, we note that the proof can be stated entirely in terms of set systems, just like the proof of Linial’s color reduction.

6. Linial used the probabilistic method to show the existence of large low intersecting set families required for solving \( P_1 \) in the setting of identical lists. As we saw in Sect. 2.3, such families can also be constructed greedily. We also used a greedy algorithm for the construction of the low intersecting set systems for problem \( P_2 \). It is therefore an interesting question whether the probabilistic method can be used to show the existence of low intersecting set systems with similar or better parameters.

– Open Problem: Remove the \( \log \beta \) term in Theorem 2 while keeping the runtime \( o(\sqrt{\log \beta}) \).

This question is particularly of interest because \( \log \beta \) is the source of the \( \sqrt{\log \Delta} \) factor in in Theorem 3 (note that the terms depending on \( m, |C| \) can be reduced, by the remarks above). The non-list \( O(\Delta^2) \)-coloring by Linial uses, in addition to his main color reduction, an \( O(\Delta^3) \)-to-\( O(\Delta^2) \) color reduction, using polynomials over finite fields [41]. With a more sophisticated use of polynomials [8] constructs a cover-free family for list coloring but it requires a much smaller outdegree. It is not clear if polynomials help with our question.

– Open Problem: More generally, prove or rule out a truly local \( (\Delta + 1) \)-coloring algorithm with \( \Delta \)-dependence \( f(\Delta) = o(\sqrt{\Delta}) \).
A color reduction via greedy construction of cover-free families

For an integer \( \Delta \geq 1 \), a \( \Delta \)-cover free family \( \mathcal{F} \subseteq \mathcal{P} \) over a universe \( U \) is a collection of subsets of \( U \), such that no set in \( \mathcal{F} \) is contained in the union of \( \Delta \) others.

To be self contained we quickly repeat how [41] use such families to reduce a coloring: Suppose a graph \( G \) with \( \Delta \)-coloring and there is a globally known \( \Delta \)-cover free family \( \mathcal{F} = \{ C_1, C_2, \ldots \} \) with \( |\mathcal{F}| \geq m \) over a universe \( U = [m] \). Then \( G \) can be recolored with \( m' \) colors, as follows. Every node \( v \) of input color \( x \) selects \( C(v) = C_x \subseteq U \), sends \( x \) to its neighbors, after which every node picks a color \( y \in C(v) \setminus \cup_{u \in N(v)} C(u) \) : \( y \) exists since \( |N(v)| \leq \Delta \) and \( \mathcal{F} \) is \( \Delta \)-cover free. By the choice of \( y \), the coloring is proper.

Theorem 7 For integers \( \Delta \geq 2 \) and \( m \geq 2 \), there is a greedy algorithm that constructs a \( \Delta \)-cover free set family \( \mathcal{F} \) of size \( m \) over a universe of size at most \( 5.2\Delta^2 \log_2(em) \).

Proof Let \( z = \lfloor \ln m \rfloor \), \( x = \Delta z \) and \( l = \lfloor e^2 x/z \rfloor \), and note that \( z \leq x \). Let \( \mathcal{P} \) be the set of all functions from \( [x] \) to \( [I] \). Two functions conflict if they agree on at least \( z \) inputs. Consider the following greedy procedure for selecting a conflict-free set \( \mathcal{R} \) of functions:

- \( \mathcal{R} \leftarrow \emptyset \), \( T \leftarrow \mathcal{P} \),
- while \( T \neq \emptyset \)
  - \( \mathcal{R} \leftarrow \mathcal{R} \cup \{ f \} \) for arbitrary \( f \in T \)
  - \( T \leftarrow T \setminus \{ g \in T \mid g \text{ conflicts with } f \} \).

No pair of functions in \( \mathcal{R} \) conflict: After each iteration of the while-loop the set \( T \) only contains functions that do not conflict with any function in \( \mathcal{R} \). Thus, in each iteration the function \( f \in T \) that we add to \( \mathcal{R} \) does not conflict with any previously picked function in \( \mathcal{R} \).

\[ |\mathcal{R}| \geq m \]: For a given function \( f \), there are at most \( d = \binom{l}{z} \) functions in \( \mathcal{P} \) that conflict with \( f \); every such function can be obtained by making it agree with \( f \) at some \( z \) inputs, and choosing the value on the rest of the inputs arbitrarily. Therefore, the number of functions removed from \( T \) at every step is at most \( d \). Thus, the process runs for at least \( |\mathcal{P}|/d \) iterations and using \( \binom{l}{z} \geq \left( \frac{e^2}{z} \right)^z \geq e \) and \( z \geq \ln m \) we obtain \( |\mathcal{R}| \geq \frac{|\mathcal{P}|}{d} \geq \left( \frac{e^2}{z} \right)^z \geq e \geq m \).

Now, the \( \Delta \)-cover free family \( \mathcal{R} \) over the universe \( U = \{ x \times I \} \) consists of sets \( S_f = \{(i, f(i)) \mid i \in [x] \} \subseteq U \), for all \( f \in \mathcal{R} \). Let \( f \neq g \in \mathcal{R} \). Since \( f \) and \( g \) do not conflict, there are at most \( z - 1 \) values \( i \in [x] \), s.t. \( f(i) = g(i) \). Therefore, \( |S_f \cap S_g| \leq z - 1 \). Since \( |S_f| = x \times \Delta \), for every \( \Delta \) distinct functions \( g_1, \ldots, g_{\Delta} \in \mathcal{R} \setminus \{ f \} \), we have \( |S_f \cap \bigcup_{i=1}^{\Delta} S_{g_i}| \geq x - (\Delta - 1) = \Delta \). The size of the universe is \( |U| = x \cdot I \leq 7.4\Delta^2 z \leq 7.4\Delta^2 \log(\text{em}) \leq 5.2\Delta^2 \log_2(\text{em}) \).

References

1. Awerbuch, B., Goldberg, A.V., Luby, M., Plotkin, S.A.: Network decomposition and locality in distributed computation. In: FOCS, pp. 364–369 (1989)
2. Balliu, A., Brandt, S., Efron, Y., Hirvonen, J., Maus, Y., Olivetti, D., Suomela, J.: Classification of distributed binary labeling problems. In: DISC, pp. 17:1–17:17 (2020)
3. Balliu, A., Brandt, S., Hirvonen, J., Olivetti, D., Rabie, M., Suomela, J.: Lower bounds for maximal matchings and maximal independent sets. In: FOCS, pp. 481–497 (2019)
4. Balliu, A., Brandt, S., Olivetti, D.: Distributed lower bounds for ruling sets. In: FOCS, pp. 365–376 (2020)
5. Balliu, A., Hirvonen, J., Lenzen, C., Olivetti, D., Suomela, J.: Locality of not-so-weak coloring. In: SIROCCO, pp. 37–51 (2019)
6. Balliu, A., Kuhn, F., Olivetti, D.: Distributed edge coloring in time quasi-polynomial in delta. In: PODC, pp. 289–298 (2020)
7. Bamberger, P., Kuhn, F., Maus, Y.: Efficient deterministic distributed coloring with small bandwidth. In: Y. Emeek, C. Cachin (eds.) PODC, pp. 243–252. ACM (2020)
8. Barenboim, L.: Deterministic (\( \Delta+1 \))-coloring in sublinear (in Delta) time in static, dynamic, and faulty networks. J. ACM \textbf{63}(5), 47:1–47:22 (2016)
9. Barenboim, L., Elkin, M.: Sublogarithmic distributed MIS algorithm for sparse graphs using Nash-Williams decomposition. Distrib. Comput. \textbf{22}(5–6), 363–379 (2010)
10. Barenboim, L., Elkin, M.: Deterministic distributed vertex coloring in polylogarithmic time. J. ACM \textbf{58}(5), 23:1–23:25 (2011)
11. Barenboim, L., Elkin, M.: Distributed Graph Coloring: Fundamentals and Recent Developments. Morgan & Claypool Publishers (2013)
12. Barenboim, L., Elkin, M., Goldenberg, U.: Locally-Iterative Distributed (\( \Delta+1 \))-coloring below Szegedy-Vishwanathan barrier, and applications to self-stabilization and to restricted-bandwidth models. In: PODC, pp. 437–446 (2018)
13. Barenboim, L., Elkin, M., Kuhn, F.: Distributed (\( \Delta+1 \))-coloring in linear (in delta) time. SIAM J. Comput. \textbf{43}(1), 72–95 (2014)
14. Barenboim, L., Elkin, M., Pettie, S., Schneider, J.: The locality of distributed symmetry breaking. J. ACM \textbf{63}(3), 20:1-20:45 (2016)
15. Bernshteyn, A.: A fast distributed algorithm for (\( \Delta+1 \))-edge-coloring. CoRR abs/2006.15703 (2020). arXiv:2006.15703
16. Brandt, S.: An automatic speedup theorem for distributed problems. In: PODC, pp. 379–388 (2019)
17. Brandt, S., Fischer, O., Hirvonen, J., Keller, B., Lempääläinen, T., Rybicki, J., Suomela, J., Uitto, J.: A lower bound for the distributed lovász local lemma. In: STOC, pp. 479–488 (2016)
18. Brandt, S., Olivetti, D.: Truly tight-in-Δ bounds for bipartite maximal matching and variants. In: PODC, pp. 69–78. ACM (2020)
19. Chang, Y., Kopelowitz, T., Pettie, S.: An exponential separation between randomized and deterministic complexity in the LOCAL model. SIAM J. Comput. 48(1), 122–143 (2019)
20. Chang, Y., Li, W., Pettie, S.: An optimal distributed (Δ+1)-coloring algorithm? In: STOC, pp. 445–456 (2018)
21. Erdös, P., Frankl, P., Füredi, Z.: Families of finite sets in which no set is covered by the union of r others. Israel J. Math. 51, 79–89 (1985)
22. Fischer, M., Ghaffari, M., Kuhn, F.: Deterministic distributed edge-coloring via hypergraph maximal matching. In: FOCS, pp. 180–191 (2017)
23. Fraigniaud, P., Heinrich, M., Kosowski, A.: Local conflict coloring. In: FOCS, pp. 625–634 (2016)
24. Fraigniaud, P., Paz, A.: The topology of local computing in networks. In: ICALP, pp. 128:1–128:18 (2020)
25. Gavoille, C., Klasing, R., Kosowski, A., Kuszner, L., Navarra, A.: On the complexity of distributed graph coloring with local minimality constraints. Networks 54(1), 12–19 (2009)
26. Ghaffari, M., Harris, D.G., Kuhn, F.: On derandomizing local distributed algorithms. In: FOCS, pp. 662–673 (2018)
27. Ghaffari, M., Hirvonen, J., Kuhn, F., Maus, Y.: Improved distributed delta-coloring. In: PODC, pp. 427–436 (2018)
28. Ghaffari, M., Hirvonen, J., Kuhn, F., Maus, Y., Suomela, J., Uitto, J.: Improved distributed degree splitting and edge coloring. In: DISC, pp. 19:1–19:15 (2017)
29. Ghaffari, M., Kuhn, F., Maus, Y.: On the complexity of local distributed graph problems. In: STOC, pp. 784–797 (2017)
30. Ghaffari, M., Kuhn, F., Maus, Y., Uitto, J.: Deterministic distributed edge-coloring with fewer colors. In: STOC, pp. 418–430 (2018)
31. Ghaffari, M., Su, H.: Distributed degree splitting, edge coloring, and orientations. In: SODA, pp. 2505–2523 (2017)
32. Harris, D.G.: Distributed local approximation algorithms for maximum matching in graphs and hypergraphs. In: FOCS, pp. 700–724 (2019)
33. Harris, D.G., Schneider, J., Su, H.: Distributed (Delta +1)-coloring in sublogarithmic rounds. J. ACM 65(4), 19:1–19:21 (2018)
34. Hefetz, D., Kuhn, F., Maus, Y., Steger, A.: Polynomial lower bound for distributed graph coloring in a weak LOCAL model. In: DISC, pp. 99–113 (2016)