



Laboratoire Bordelais de Recherche en Informatique
UMR 5800 - Université Bordeaux 1, 351, cours de la Libération,
33405 Talence CEDEX, France

Research Report RR-1440-07

On the Complexity of Distributed Graph Coloring with Local Minimality Constraints

Cyril Gavoille, Ralf Klasing, Adrian Kosowski,
Łukasz Kuszner, Alfredo Navarra

December 2007

On the Complexity of Distributed Graph Coloring with Local Minimality Constraints*

Cyril Gavoille[†] Ralf Klasing[†] Adrian Kosowski[‡]

Lukasz Kuszner[‡] Alfredo Navarra[§]

Abstract

Distributed Greedy Coloring is an interesting and intuitive variation of the standard Coloring problem. Given an order among the colors, a coloring is said to be *greedy* if there does not exist a vertex for which its associated color can be replaced by a color of lower position in the fixed order without violating the property that neighbouring vertices must receive different colors. We consider the problems of *Greedy Coloring* and *Largest First Coloring* (a variant of greedy coloring with strengthened constraints) in the Linial model of distributed computation, providing lower and upper bounds and a comparison to the $(\Delta + 1)$ -Coloring and *Maximal Independent Set* problems, with Δ being the maximum vertex degree in G .

Keywords: Distributed Computing, Graph Coloring, Greedy Algorithm, Randomisation

1 Introduction

To *color* the vertices of a graph G means to give each vertex a positive integer color value in such a way that no two adjacent vertices get the same color. In many practical considerations, it is desirable to minimise the number of colors used.

*The research was partially funded by the State Committee for Scientific Research (Poland) Grant 4 T11C 047 25, by the Future and Emerging Technologies Unit of EC (IST priority - 6th FP), under contract no. FP6-021235-2 (project ARRIVAL), the project “CEPAGE” of INRIA, and by European projects COST Action 293 “Graphs and Algorithms in Communication Networks” (GRAAL) and COST Action 295 “Dynamic Communication Networks” (DYNAMO). Some of the results of this paper were announced in [9, 14].

[†]LaBRI - Université Bordeaux 1 - CNRS, 351 cours de la Libération, 33405 Talence cedex, France. E-mail: {gavoille,klasing}@labri.fr

[‡]Department of Algorithms and System Modeling, Gdańsk University of Technology, Narutowicza 11/12, 80952 Gdańsk, Poland. Email: {kosowski,kuszner}@sphere.pl

[§]Dipartimento di Matematica e Informatica, Università degli Studi di Perugia, Via Vanvitelli 1, 06123 Perugia, Italy. Email: navarra@dipmat.unipg.it

Many applications ranging over code assignment in wireless networks [2], scheduling problems [8,20], track assignment in railway optimisation [4–6] and so forth, can be solved by means of minimum coloring.

If at most k colors are used, the result is called a k -coloring. The smallest possible positive integer k for which there exists a k -coloring of G is called the *chromatic number* $\chi(G)$. It has to be remembered that even in a centralised setting, approximating $\chi(G)$ within a factor of $n^{1-\varepsilon}$ is an NP-hard problem, for any $\varepsilon > 0$ [26].

In the context of distributed computing, the performance of a graph coloring algorithm on a system graph G is characterised by at least two main parameters: the number of colors used by the algorithm to color the graph G and the number of rounds required to obtain a coloring. Designing a fast distributed algorithm (i.e., an algorithm running within $o(D)$ rounds where D is the diameter of the graph G) which always uses a number of colors in some way bounded from above with respect to $\chi(G)$ appears to be difficult; e.g., in [19] it is proved that coloring trees of diameter D requires $\Omega(D)$ rounds even if we allow the algorithm to use $\sqrt{\Delta}$ colors. For this reason, it is natural to pose the problem of constructing distributed algorithms having properties analogous to certain centralised graph coloring heuristics, which are known to work well in practice. The class of greedy algorithms considered herein is of special significance due to the elegance of their formulation, and the number of graph classes for which they always produce optimal or near-optimal results.

1.1 Preliminaries: Greedy Coloring in a Centralised Setting

For a given graph G and a sequence K of all its vertices, $K = (v_1, v_2, \dots, v_n)$, we will use the term *greedy coloring* to describe the following procedure of locally minimal color assignment:

algorithm Greedy-Color(G, K):

for $v := v_1$ **to** v_n **do**

give vertex v the smallest possible color not used by any
of the already colored neighbours of v ;

Different graph coloring algorithms are obtained by choosing sequence K in a specific way and then applying the **Greedy-Color** procedure. Below we briefly recall some of the types of sequences most often applied in practice (cf. [15,16]).

- *General Greedy (G)* sequence: K is an arbitrarily chosen sequence of vertices.
- *Largest-First (LF)* sequence: K is formed by arranging the vertices of graph G in non-ascending order of degrees.
- *Smallest-Last (SL)* sequence: K is formed by iteratively removing a vertex of minimal degree from the graph and placing it at the end of K .

All the considered algorithms obviously lead to correct colorings of the graph, but the converse does not hold — not every coloring of a graph can be obtained by

applying a specific algorithm. We will call a coloring of a graph an *A-coloring* if it may be obtained by means of greedy coloring with a sequence of some type A.

Observe that all *G*-colorings have the much desired local minimality property described by Grundy [11], namely, no single vertex may have its color value decreased without affecting the color of some other (neighbouring) vertex. As a direct consequence, all *G*-colorings use not more than $\Delta + 1$ colors, with Δ being the maximum vertex degree in *G*. LF-colorings and SL-colorings are special types of *G*-colorings, thus they inherit this property.

For graph classes found in practice greedy colorings obtained according to specific sequences may admit even stronger bounds [16,22,25]. For example, any SL-coloring of a planar graph uses not more than 6 colors, whereas any LF-coloring is optimal or within a fixed number of colors of the optimum for numerous graph classes, including complete *k*-partite graphs, caterpillars, crowns, and bipartite wheels [16].

1.2 Model of Distributed Computation

We consider restricted variants of the vertex coloring problem in a distributed network, assuming the so-called Linial model of computation, which is widely used in previous research on the subject [3, 18, 19, 24]. Such a distributed network consists of a set *V* of processors and a set *E* of bidirectional communication links between pairs of processors. It can be modeled by an undirected graph $G = (V, E)$. We denote $n = |V|$.

Vertices are identified by means of unique labels. Each vertex has its local state, described by a certain number of integer variables. The system functions in so-called *synchronized rounds*, consisting of three steps: a vertex first reads its own local state variables and the local state variables of all its neighbours, then performs an arbitrary amount of local computations, and finally updates its local state variables accordingly.

Although local computations are unbounded, the case of randomised algorithms, i.e, algorithms that can make use of a so-called *coin-flip* function, is usually separately studied. Section 4 is devoted to such a variant.

In all further considerations we assume that two global parameters are known to all vertices: some constant-factor upper bounds on the number of vertices *n* and on the maximum vertex degree Δ . This assumption is used only to provide a simple mechanism for executing subroutines of known complexity with respect to *n* and Δ and waiting a known number of rounds for their completion.

For convenience of notation, we will assume that the special local state variable *c* always stores the outcome of the algorithm; for example, in a coloring process, $c(v)$ is the color value assigned to vertex *v*. The algorithm is considered to be complete when the values of all variables *c* are correctly set and the algorithm will not modify them in any subsequent round.

For each vertex *v* define its *neighborhood* $N(v) = \{u : \{u, v\} \in E\}$ and *vertex degree* $\deg_G(v) = |N(v)|$. The maximum degree among all vertices in *G* is denoted by $\Delta = \max_{v \in V} \{\deg_G(v)\}$. In order to distinguish among neighbours of higher degree, we will use the symbol $N_{\geq}(v) = \{u \in N(v) : \deg(u) \geq \deg(v)\}$ and similarly

$N_{>}(v) = \{u \in N(v) : \deg(u) > \deg(v)\}$. The *distance- d neighbourhood* $N_d(v)$ of vertex v is defined as the set of all vertices at a distance between 0 and d from v . For any subset of vertices $S \subseteq V$, we denote the subgraph of G induced by vertex set S as $G[S]$. The length of the shortest cycle in graph G is known as its *girth* and denoted by $g(G)$.

1.3 Problem Definitions

In this paper we consider two variants of the graph coloring problem in a distributed setting, those of obtaining a **G**-coloring and of obtaining an **LF**-coloring of the system graph, denoted **G-COL** and **LF-COL** respectively. As mentioned in the Subsection 1.1, **G-COL** and **LF-COL** are refinements of the problem of finding a $(\Delta + 1)$ -coloring of the system graph G (simply denoted **COL**), and thus are not easier in the sense of computation time. In fact, **G-COL** and **LF-COL** are also not easier than the problem of finding a maximal independent set in G (denoted **MIS**), since for any **G**-coloring of G the set of all vertices having color 1 is clearly a maximal independent set.

Nevertheless, it is possible to put forward a definition of **G-COL** and **LF-COL** which demonstrates the local nature of the imposed constraints; for completeness we provide local formulations of **COL** and **MIS** as well.

Definition 1. *The considered distributed problems are defined by the following constraints on the local variable c at any vertex v :*

$$(\Delta + 1)\text{-Coloring (COL):} \quad c(v) \in \{1, \dots, \Delta + 1\} \setminus c(N(v)).$$

$$\text{Maximal Independent Set (MIS):} \quad c(v) \neq 0 \Leftrightarrow c(N(v)) = \{0\}.$$

$$\text{Greedy Coloring (G-COL):} \quad c(v) = \min \{1, \dots, \Delta + 1\} \setminus c(N(v)).$$

$$\text{Largest-First Coloring (LF-COL):} \quad c(v) = \min \{1, \dots, \Delta + 1\} \setminus c(N_{\geq}(v)).$$

The proof of the equivalence of the above definitions with the prior characterisations is straightforward. For the **G-COL** problem, consider any **G**-coloring c of graph G , and an arbitrarily chosen vertex v . Since at the time of coloring some of the vertices from $N(v)$ are already colored and v obtains the minimum possible color, we have $c(v) \leq \min \{1, \dots, \Delta + 1\} \setminus c(N(v))$, but also $c(v) \notin c(N(v))$ by the legality of the coloring, thus $c(v) = \min \{1, \dots, \Delta + 1\} \setminus c(N(v))$. Conversely, given a set of color values fulfilling $c(v) = \min \{1, \dots, \Delta + 1\} \setminus c(N(v))$ we can always construct a corresponding sequence K for the **Greedy-Color** procedure simply by ordering vertices according to non-decreasing values of $c(v)$. For the definition of **LF-COL** the argument is analogous; an appropriate sequence K can be constructed by ordering vertices according to non-increasing values of $\deg(v)$, breaking ties according to non-decreasing values of $c(v)$.

As a side note, it is interesting to observe that the problem of finding an **SL**-coloring of graph G does not admit any local definition. Indeed, when G is a ring of even length, any such coloring uses exactly 2 colors. Taking into account that the 2-coloring of a ring requires $\Omega(n)$ rounds (cf. e.g. [19]), this means that finding a distributed **SL**-coloring of a graph may also require $\Omega(n)$ rounds; such a problem is of little interest in a distributed setting. The same also holds for well-known algorithms with dynamic sequences, such as **DSATUR** [15].

1.4 State-of-the-art Results

In terms of n , the fastest distributed algorithm for MIS is $2^{O(\sqrt{\log n})}$ [24]. The same bound holds for COL, taking into account the simple algorithm proposed by Johanson [13] which finds a $(\Delta+1)$ -coloring of an arbitrary graph by direct reduction to one iteration of any algorithm for MIS. The current best lower bound is only $\Omega(\log^* n)$ for COL [19] and $\Omega\left(\sqrt{\frac{\log n}{\log \log n}}\right)$ for MIS [17]. The distributed time complexity of COL and MIS is thus widely open.

For the general graph coloring problem some extremely fast algorithms have been described. Linial in [19] gave an algorithm working in $O(\log^* n)$ time but using $O(\Delta^2)$ colors. Algorithms having strong bounds on the number of colors usually only work for some specific graph classes. For example in [10] a technique for coloring triangle-free graphs using $O(\Delta/\log \Delta)$ colors was proposed, but the algorithm may fail for some instances of the problem (i.e., for some triangle-free graphs).

To the best of our knowledge, the first distributed approach to greedy graph coloring was proposed by Panconesi and Rizzi [23] who used a forest decomposition technique to achieve a coloring fulfilling the constraints of G-COL in $O(\Delta^2 \log^* n)$ time. Recently an algorithm motivated by sequential LF-coloring was described in [12]. Analysis shows that it runs in $O(\Delta^2 \log n)$ time. The solution obtained by this algorithm is always greedy (solving the G-COL problem), but does not satisfy the constraints of the LF-COL problem for some instances.

1.5 Our Contribution

We provide lower and upper bounds on the time complexity of Greedy Coloring (G-COL) and Largest First Coloring (LF-COL) with respect to Coloring (COL) and Maximal Independent Set (MIS).

A summary of the results is contained in Table 1. The obtained lower and upper bounds for the G-COL and LF-COL problems are expressed in terms of parameters n and Δ . In particular, we prove a lower bound of $\Omega\left(\frac{\log n}{\log \log n}\right)$ for G-COL and $\Omega(\sqrt{n})$ for LF-COL, an improvement upon the current MIS lower bound [17]. We note that our lower bounds also apply for randomized algorithms (cf. Section 4).

The rest of the paper is organised as follows. The next section provides lower bounds for the considered problems. Upper bound results for deterministic algorithms can be found in Section 3, while a discussion concerning upper bound results for randomised algorithms can be found in Section 4. Finally, Section 5 provides conclusive remarks.

2 Lower Bounds

2.1 The G-COL Problem

For a given graph G , let $C_G(v)$ denote the set of all possible colors which may be assigned to vertex $v \in V$, taken over all greedy colorings of G . For trees, we have

Lower bounds (deterministic and randomized)

COL	MIS	G-COL	LF-COL
$\Omega(\log^* n)$ [18]	$\Omega\left(\sqrt{\frac{\log n}{\log \log n}}\right)$ [17]	$\Omega\left(\frac{\log n}{\log \log n}\right)$ [Thm 5]	$\Omega(\sqrt{n})$ [Thm 6]
	$\Omega\left(\frac{\log \Delta}{\log \log \Delta}\right)$ [17]		$\Omega(\Delta)$ [Thm 6]

Upper bounds (deterministic algorithms)

COL	MIS	G-COL	LF-COL
$2^{O(\sqrt{\log n})}$ [1, 24]	$2^{O(\sqrt{\log n})}$ [1, 24]	$2^{O(\sqrt{\log n})}$ [Thm 8]	$O(\sqrt{n} \cdot \mathcal{T}_{\text{MIS}})$ [Thm 12]
$O(\Delta \log n)$ [1] $O(\Delta^2 + \log^* n)$ [23]	$O(\Delta + \mathcal{T}_{\text{COL}})$ [Pro 9]	$O(\Delta + \mathcal{T}_{\text{COL}})$ [Pro 9]	$O(\Delta \cdot \mathcal{T}_{\text{MIS}})$ [Thm 11]

Upper bounds (randomised algorithms)

COL	MIS	G-COL	LF-COL
$O(\log n)$ [13, 21]	$O(\log n)$ [21]	$2^{O(\sqrt{\log n})}$ [Cor 13]	$O(\sqrt{n} \log n)$ [Cor 14]
$O(\Delta^2 + \log^* n)$ [23]	$O(\Delta + \mathcal{T}_{\text{COL}}^R)$ [Cor 13]	$O(\Delta + \mathcal{T}_{\text{COL}}^R)$ [Cor 13]	$O(\Delta \log \Delta \log n)$ [Cor 14]

Table 1: The time complexity of Greedy Coloring and LF-Coloring with respect to other well-known problems, $(\Delta + 1)$ -Coloring and Maximal Independent Set, in the distributed setting. For a problem P, the notation \mathcal{T}_P (resp. \mathcal{T}_P^R) describes a known upper bound on the complexity of an algorithm (resp. randomized algorithm) solving P.

the following property.

Lemma 1. *Let T be a tree and v some vertex of T such that the maximum distance from v to another vertex of T is d . Then $\max C_T(v) \leq d + 1$.*

Proof. Consider an arbitrary greedy coloring of T . If a vertex v_c receives some color $c > 1$, then it must be adjacent to a vertex v_{c-1} which is assigned color $c - 1$. By induction we easily prove that v_c must be the endpoint of a path in T with c vertices. Since by assumption the longest path of T with one endpoint in v has $d + 1$ vertices, clearly the color assigned to v in the coloring cannot exceed $d + 1$. \square

Lemma 2. *Suppose that there exists an n -vertex graph G such that for some ℓ , $g(G) \geq \ell$ and $\chi(G) > \ell/2$. Then, there does not exist a distributed algorithm for the G-COL problem which colors all graphs of at most n vertices in less than $\ell/2 - 1$ rounds.*

Proof. Suppose that there exists a distributed algorithm A which for all graphs of at most n vertices produces a greedy coloring c_A in at most d rounds. Algorithm A

must be able to assign for any vertex $v \in V$ a color $c_A(v) \in C_G(v)$, using information about the graph structure at distance at most d only. Thus, if for two graphs $G = (V, E)$ and $G' = (V', E')$ with distinguished vertices $v \in V$, $v' \in V'$ we have $G[N_d(v)] = G'[N_d(v')]$, then the coloring of graph G produced by algorithm A has the property that $c_A(v) \in C_G(v) \cap C_{G'}(v')$.

Now, let G be the graph of girth $g(G) \geq \ell$ described in the assumption of the lemma. Let $v \in V$ be arbitrarily chosen. We now apply the proven property for graphs G and $G'_v \equiv G[N_{\frac{\ell}{2}-1}(v)]$ with vertices v and $v' \equiv v$, obtaining $c_A(v) \in C_G(v) \cap C_{G'_v}(v)$. However, the graph G'_v is clearly a tree (since it is acyclic and connected) and all vertices of G'_v are at a distance of at most $\ell/2 - 1$ from v . Hence by Lemma 1 we obtain $\max C_{G'_v}(v) \leq \ell/2$, and we conclude that for graph G , $c_A(v) \leq \ell/2$. Since v was arbitrarily chosen, we have $c_A(v) \leq \ell/2$ for all $v \in V$. We have thus obtained a coloring of G using at most $\ell/2$ colors, whereas by assumption the chromatic number $\chi(G)$ is larger than $\ell/2$, a contradiction. \square

Lemma 3 ([7]). *For all $n, k \in \mathbb{N}$ ($4 \leq k \leq n$) there exists an n -vertex graph G , such that $\chi(G) \geq k$ and $g(G) \geq \frac{1}{4} \frac{\log n}{\log k}$.*

Corollary 4. *For all values of $n \geq 2^{16}$ there exists a graph G of order n , such that $\chi(G) \geq \frac{\log n}{\log \log n}$ and $g(G) \geq \frac{1}{4} \frac{\log n}{\log \log n}$.*

Proof. For any given value of $n \geq 2^{16}$, put $k = \left\lceil \frac{\log n}{\log \log n} \right\rceil$ in Lemma 3. We immediately obtain that there exists a graph G of order n such that the following conditions are fulfilled:

$$\chi(G) \geq k = \left\lceil \frac{\log n}{\log \log n} \right\rceil \geq \frac{\log n}{\log \log n}, \quad (1)$$

$$g(G) \geq \left\lceil \frac{1}{4} \frac{\log n}{\log k} \right\rceil \geq \frac{1}{4} \frac{\log n}{\log \log n - \log \log \log n + 1} \geq \frac{1}{4} \frac{\log n}{\log \log n}, \quad (2)$$

which completes the proof. \square

Combining Corollary 4 with Lemma 2 (putting $l = \frac{1}{4} \frac{\log n}{\log \log n}$) we immediately obtain that for all values of $n \geq 2^{16}$, no distributed algorithm can solve the G-COL problem in less than $\frac{1}{8} \frac{\log n}{\log \log n} - 1$ rounds for all graphs of order at most n . Thus, we may write the following theorem.

Theorem 5. *The distributed time complexity of G-COL is $\Omega\left(\frac{\log n}{\log \log n}\right)$.*

2.2 The LF-COL Problem

Theorem 6. *The distributed time complexity of LF-COL is $\Omega(\sqrt{n})$ and $\Omega(\Delta)$.*

Proof. Consider a family of graphs $\{G_d\}_{d=4,5,6,\dots}$ having $n(G_d) = \frac{5}{2}d^2 - \frac{1}{2}d + 1$ and $\Delta(G) = 2d$, whose representative is depicted in Figure 1. Vertices v_0, v_1, \dots, v_d induce a path of length d . Some additional components are connected to particular vertices to ensure that vertex v_i obtains a color $d - i + 1$ in each LF-coloring of

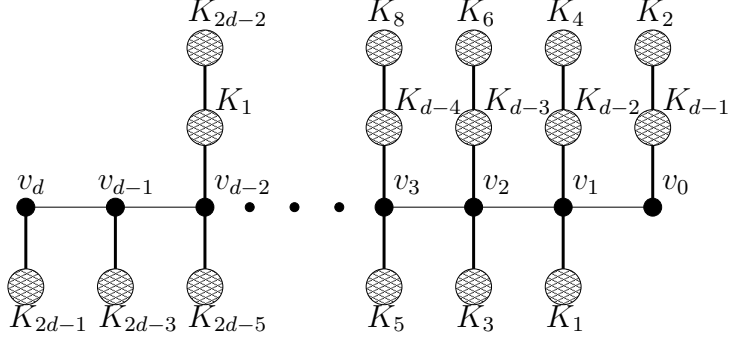


Figure 1: A graph which requires $\Omega(d)$ time to LF-color in the distributed model

G_d , that is component K_r depicts a complete graph with r vertices and a bold line between such a component and v_i illustrates that each of the vertices of K_r is connected to v_i . Similarly, when two components K_{r_1} and K_{r_2} are connected, each vertex from K_{r_1} is connected to each vertex from K_{r_2} , thus forming a clique $K_{r_1+r_2}$. We have $\deg(v_i) = d + i$, hence the vertices of the path appear in the LF sequence in the order $v_d, v_{d-1}, \dots, v_1, v_0$. Moreover, we have $|N_{>}(v_i)| = d - i$, and it is easy to obtain inductively that in any LF-coloring of G_d the set of colors used in $N_{>}(v_i)$ is $\{1, 2, \dots, d - i\}$, hence the only possible color for v_i in an LF-coloring is $d - i + 1$. However, if vertex v_d were to be removed from the graph, the colors of all other vertices of the path would decrease by 1. Thus, we have shown that the color of v_0 depends on the existence of vertex v_d : if v_d exists, v_0 must obtain color $d + 1$, otherwise it must obtain color d . As vertices v_d and v_0 are at a distance of d from each other, and information in our model can propagate only at the speed of one vertex per round, the coloring cannot be completed in less than d rounds, and the claim follows. \square

3 Upper Bounds for Deterministic Algorithms

3.1 The G-COL Problem

We recall after [1,24] that a (j, k) -decomposition of a graph $G = (V, E)$ is a partition $\{C_1, C_2, \dots\}$ of V such that the following conditions are fulfilled:

- each subgraph $G[C_i]$, called a *cluster*, is connected and of diameter at most j ,
- the so called *cluster graph*, obtained from G by contracting each C_i into one vertex, is vertex colored with at most k colors.

Lemma 7. *Given a (j, k) -decomposition of a graph, there exists a distributed $O(j \cdot k)$ time algorithm for solving the G-COL problem.*

Proof. Once the (j, k) -decomposition is given, we have a situation in which the clustered graph is already colored (perhaps not greedily). The algorithm then proceeds according to the assigned colors: first clusters of color 1 are processed in parallel, then clusters of color 2 are processed in parallel, and so forth. Inside each cluster, a leader is elected in order to collect all the information from the related cluster. This operation clearly requires $O(j)$ rounds to retrieve all the information about the cluster. The leader then greedily assigns a suitable color to each vertex of its cluster respecting the greedy coloring property, i.e., each vertex gets the smallest color possible with respect to its already colored neighbours from other clusters. Again this requires $O(j)$ rounds. Note that clusters of color 1 have no constraints with respect to the other clusters. In the case of clusters of color 2, the greedy coloring may begin only after waiting $O(k)$ steps for all the neighbours from clusters of color 1 to be colored. Consequently, a coloring of the last clusters is complete after $O(j \cdot k)$ rounds. \square

The following theorem is obtained directly from Lemma 7, taking into account the results of [24] where a $(2^{O(\sqrt{\log n})}, 2^{O(\sqrt{\log n})})$ -decomposition of any n -vertex graph is provided in $2^{O(\sqrt{\log n})}$ distributed time.

Theorem 8. *There exists a distributed algorithm for the G-COL problem running in $2^{O(\sqrt{\log n})}$ time.*

Proposition 9. *There exists a distributed $O(\Delta + \mathcal{T}_{\text{COL}})$ algorithm for G-COL, where \mathcal{T}_{COL} denotes a known upper bound on the execution time of an algorithm for the COL problem in G .*

Proof. Suppose that a $(\Delta + 1)$ -coloring of graph G is already provided. This induces a partition of the vertex set into independent sets $V = IS_1 \cup IS_2 \cup \dots \cup IS_{\Delta+1}$. It is easy to see that in the i -th round we can simultaneously assign the minimum possible color to all vertices from IS_i , taking into account the constraints imposed by the colors of vertices from $IS_1 \cup \dots \cup IS_{i-1}$. Thus, a greedy coloring of the graph can be obtained $\Delta + 1$ rounds after the initial $(\Delta + 1)$ -coloring is known to be complete. \square

3.2 The LF-COL Problem

Before discussing the details of the distributed implementation of the LF algorithm, we present an equivalent characterisation of a correct LF-coloring. For a given coloring of G , let $IS_{(j,k)} \in V$, for any $1 \leq j \leq \Delta$, $1 \leq k \leq \Delta + 1$, denote the independent set of vertices of G of degree j and colored with color k . We define $H_{(j,k)} \subseteq G$ as the subgraph induced by the set of all vertices $v \in V$ such that $\deg_G(v) = j$ and $v \notin \left(\bigcup_{k_i < k} IS_{(j,k_i)} \right) \cup N \left(\bigcup_{j_i > j} IS_{(j_i,k)} \right)$.

Lemma 10. *Given an assignment of colors $c : V(G) \rightarrow \mathbb{N}$, if for all j, k the set $IS_{(j,k)}$ is a maximal independent set in $H_{(j,k)}$, then c is an LF-coloring of G .*

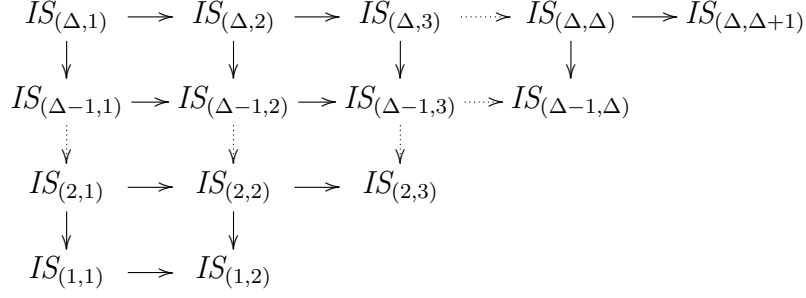


Figure 2: Illustration of the time ordering of independent set construction

Proof. Observe that if a coloring c of G fulfills the assumption of the lemma, then it is identical to the coloring c' obtained using the greedy algorithm with the largest-first sequence of vertices: $K = (IS_{(\Delta,1)}, IS_{(\Delta,2)}, \dots, IS_{(\Delta,\Delta+1)}, IS_{(\Delta-1,1)}, IS_{(\Delta-1,2)}, \dots, IS_{(\Delta-1,\Delta)}, \dots, IS_{(1,1)}, IS_{(1,2)})$, where the elements of each independent set may be enumerated in arbitrary order. Indeed, suppose that for all vertices u which appear in sequence K before some fixed vertex v both colorings are identical, $c(u) = c'(u)$. Putting $j = \deg_G(v)$, for all $j_i < c(v)$, $IS_{(j,k_i)}$ is a maximal independent set in $H_{(j,k_i)}$, $v \in V(H_{(j,k_i)})$, and $v \notin IS_{(j,k_i)}$, hence $IS_{(j,k_i)} \cup \{v\}$ is not an independent set in G . This means that v has at least one neighbour $u \in IS_{(j,k_i)}$ which appears earlier in K , and $c'(u) = k_i$, so $c'(v) \neq k_i$. In this way we obtain that $c'(v) \geq c(v)$. On the other hand, all vertices u which appear before v in K and have color $c'(u) = c(v)$ either belong to the same independent set $IS_{(j,c(v))}$ as v , or to some independent set $IS_{(j_i,c(v))}$, $j_i > j$, and $v \notin N(IS_{(j_i,c(v))})$. Consequently, color $c(v)$ is the smallest legal color for v in the greedy coloring, so we have $c'(v) = c(v)$, which completes the proof by an inductive argument. \square

We now propose a distributed algorithm which defines the maximal independent sets $IS_{(j,k)} \subseteq H_{(j,k)}$. Let r be a known upper bound on the number of rounds required to compute an independent set in any subgraph of G . Then the execution of the proposed algorithm is divided into $\Delta + 1$ steps, each of which lasts r rounds. In the i -th step, $1 \leq i \leq \Delta + 1$, we compute at once all of the independent sets $IS_{(j,k)}$ such that $1 \leq j \leq \Delta$, $1 \leq k \leq \Delta + 1$, and $j - k = \Delta - i$. Thus, in step 1 we compute only a maximal independent set $IS_{(\Delta,1)}$ in graph $H_{(\Delta,1)}$, i.e. in the subgraph of G induced by all vertices of degree Δ . In step 2 we compute a maximal independent set $IS_{(\Delta-1,1)}$ in the subgraph of G induced by all vertices of degree $\Delta - 1$, and at the same time maximal independent set $IS_{(\Delta,2)}$ in the subgraph of G induced by all vertices of degree Δ not belonging to $IS_{(\Delta,1)}$, and so on. An illustration of the time ordering of the independent set construction is shown in Fig. 2. By Lemma 10, when all vertices have been assigned to some independent set, the result can be interpreted as a distributed LF-coloring of the graph. The time complexity of the algorithm is determined by $\Delta + 1$ steps of an algorithm for the MIS problem, hence we have the following theorem.

Theorem 11. *There exists a distributed $O(\Delta \cdot \mathcal{T}_{\text{MIS}})$ algorithm for LF-COL, where \mathcal{T}_{MIS} denotes a known upper bound on the execution time of an algorithm for the*

MIS problem in graphs of order at most n and degree at most Δ .

Theorem 12. *There exists a distributed $O(\sqrt{n \cdot \mathcal{T}_{\text{MIS}}})$ algorithm for LF-COL, where \mathcal{T}_{MIS} denotes a known upper bound on the execution time of an algorithm for the MIS problem in graphs of order at most n and degree at most Δ .*

Proof. Consider a partition of the vertex set $V(G) = V_1 \cup V_2$, where V_1 contains all vertices of degree at least a , and V_2 contains all vertices of degree less than a , for some value of parameter a . Let H be the spanning subgraph of G with all edges having at least one endpoint of degree at least a , $H = G \setminus E(G[V_2])$. Let $P = (v_1 v_2 \dots v_d) \subseteq H$ be a shortest path in H connecting some two vertices v_1 and v_d . In P we must have at least $d/4$ vertices which belong to V_1 and whose neighbourhoods are pairwise disjoint in H , hence also in G . Since each of these vertices is of degree at least a , we obtain $a \cdot d/4 \leq n$, which means that the diameter of each of the connected components of H is at most $4n/a$.

The coloring of graph G is obtained by first coloring all vertices from V_1 , and then all vertices from V_2 . The former stage can be performed by electing a leader in each of the connected components of H , computing a centralised LF-coloring of all vertices from V_1 within this component, and disseminating the coloring to all vertices. In view of the obtained bound on the diameter, this step takes $O(n/a)$ time. In the latter stage, the coloring of V_2 is easily completed using an approach similar to that described by Theorem 11 in $O(a \cdot \mathcal{T}_{\text{MIS}})$ time, since all vertices are of degree less than a and obtain color at most a . The overall complexity of the algorithm is therefore $O(n/a + a \cdot \mathcal{T}_{\text{MIS}})$, and the proof of the claim is complete when we put $a = \sqrt{n/\mathcal{T}_{\text{MIS}}}$. \square

4 Notes on Randomised Algorithms

The introduction of randomisation consists in allowing local computations of vertices to involve a random coin-flip function, which returns a value 0 or 1 (a so-called *random bit*) with equal probability. We consider randomised algorithms that always stop with a correct result, whereas the time complexity of an algorithm is its *expected* running time.

Although randomized solutions to some problems may be significantly faster than the best known distributed approaches (e.g. MIS in Table 1), certain lower time bounds related to the speed of dissemination of information in the network still apply. In particular, the arguments used when proving lower bounds in Section 2 are still valid: if the distance d -neighbourhoods of a vertex v in some two system graphs G_1 and G_2 containing v are identical, and the sets of permissible values of color $c(v)$ are disjoint for graphs G_1 and G_2 , then for any coloring algorithm, for at least one of the graphs G_1, G_2 , the probability of obtaining a correct solution within d rounds cannot exceed $1/2$. Hence, by Markov's inequality, the expected execution time of any algorithm is $\Omega(d)$. In this way we obtain that all the lower bounds in Table 1 also hold for expected execution times in the randomized model.

On the other hand, deterministic algorithms can also be applied in the randomized model. Hence, all upper complexity bounds in Table 1 which are expressed in

terms of n and Δ still hold. In particular, the G-COL problem can still be solved in $2^{O(\sqrt{\log n})}$ time. The $O(\Delta + \mathcal{T}_{\text{COL}})$ -time algorithm for G-COL given by Proposition 9 is also easily adapted to allow for any randomised subroutine for the COL problem. Indeed, the initial $(\Delta + 1)$ -coloring is used only to define independent sets $\{IS_i : 1 \leq i \leq \Delta + 1\}$, and instead of waiting for the COL algorithm to complete, we can start the greedy coloring phase even before the independent sets are fully defined (for example, in round $(\Delta + 1)k + i$, $k \in \mathbb{N}$, we activate all vertices from set IS_i none of whose neighbours belong to IS_i).

Corollary 13. *There exists a distributed randomised $O(\Delta + \mathcal{T}_{\text{COL}}^R)$ expected time algorithm for G-COL, where $\mathcal{T}_{\text{COL}}^R$ denotes the complexity of a randomised algorithm for the COL problem in G .*

In the case of algorithms for the LF-COL (Theorems 11 and 12), it is possible to apply a randomised subroutine for MIS which has the property that at every stage of the algorithm the partial solution described by local variables c of those vertices which have already completed the algorithm induces a (not necessarily maximal) independent set. The randomised distributed algorithm for MIS proposed by Luby [21] has this property. Although the complexity of this algorithm is given as $O(\log n)$, when computing an LF-coloring, the subroutine for MIS is called $O(\Delta^2)$ times. Since the time distribution of executions of the Luby algorithm is dominated by the negative binomial distribution (a direct corollary of [21] Thm 1), the slowest of the iterations of MIS will almost certainly complete in $O(\log n \log \Delta)$ time. We may use this value (or $O(\log^2 n)$ for simplicity) in place of \mathcal{T}_{MIS} in Theorems 11 and 12.

Corollary 14. *There exist distributed randomised algorithms for LF-COL running in $O(\Delta \log \Delta \log n)$ and $O(\sqrt{n} \log n)$ expected time.*

5 Conclusion

The number of colors used by most distributed algorithms for $(\Delta + 1)$ -coloring, such as Johansson's algorithm [13], is close to Δ even if the graph is bipartite. This is not surprising, since such algorithm has no mechanism for economizing on the number of colors. Distributed greedy coloring (G-COL) is a natural approach for optimizing the number of colors. We have shown that such greedy coloring can be obtained within a time bound as good as the best known $(\Delta + 1)$ -coloring algorithm [24]. However, and maybe surprisingly, no polylog randomized algorithm for greedy coloring is known, unlike the case of $(\Delta + 1)$ -coloring.

References

- [1] B. Awerbuch, A. V. Goldberg, M. Luby, and S. A. Plotkin. Network decomposition and locality in distributed computation. In *30th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 364–369, 1989.

- [2] R. Battiti, A. A. Bertossi, and M. A. Bonuccelli. Assigning codes in wireless networks: bounds and scaling properties. *Wireless Networks*, 5(3):195–209, 1999.
- [3] P. Chaudhuri. Algorithms for some graph problems on a distributed computational model. *Inf. Sci.*, 43(3):205–228, 1987.
- [4] S. Cicerone, G. D’Angelo, G. Di Stefano, D. Frigioni, and A. Navarra. Robust algorithms and price of robustness in shunting problems. In *7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS)*, pages 175–190, 2007.
- [5] S. Cornelsen and G. Di Stefano. Track assignment. *Journal of Discrete Algorithms*, 5(2):250–261, 2007.
- [6] E. Dahlhaus, P. Horak, M. Miller, and J. F. Ryan. The train marshalling problem. *Discrete Applied Mathematics*, 103(1-3):41–54, 2000.
- [7] P. Erdős. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959.
- [8] M. Garey and D. Johnson. *Computers and Intractability*. W.H. Freeman and Company, 1979.
- [9] C. Gavoille, R. Klasing, A. Kosowski, and A. Navarra. On the complexity of distributed greedy coloring. In *Proc. 21st International Symposium on Distributed Computing (DISC)*, volume 4731 of *LNCS*, pages 482–484, 2007.
- [10] D. A. Grable and A. Panconesi. Fast distributed algorithms for Brooks-Vizing colorings. *J. Algorithms*, 37(1):85–120, 2000.
- [11] P. M. Grundy. Mathematics and games. *Eureka*, 2:6–8, 1939.
- [12] J. C. Hansen, M. Kubale, Ł. Kuszner, and A. Nadolski. Distributed largest-first algorithm for graph coloring. In *Proc. 10th European Conference on Parallel Computing (Euro-Par)*, pages 804–811, 2004.
- [13] Ö. Johansson. Simple distributed $(\Delta + 1)$ -coloring of graphs. *Inf. Process. Lett.*, 70(5):229–232, 1999.
- [14] A. Kosowski and Ł. Kuszner. On greedy graph coloring in the distributed model. In *Proc. 12th European Conference on Parallel Computing (Euro-Par)*, volume 4128 of *LNCS*, pages 592–601, 2006.
- [15] A. Kosowski and K. Manuszewski. *Graph Colorings*, volume 352, chapter Classical Coloring of Graphs, pages 1–20. AMS, USA, Providence, 2004.
- [16] M. Kubale. *Introduction to Computational Complexity and Algorithmic Graph Coloring*. GTN, Gdańsk, 1998.

- [17] F. Kuhn, T. Moscibroda, and R. Wattenhofer. What cannot be computed locally! In *23rd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 300–309, 2004.
- [18] N. Linial. Legal coloring of graphs. *Combinatorica*, 6(1):49–54, 1986.
- [19] N. Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992.
- [20] Z. Lotker and A. Navarra. Managing Random Sensor Networks by means of Grid Emulation. In *Proceedings of the The 5th International IFIP-TC6 Networking Conference*, volume 3976 of *Lecture Notes in Computer Science*, pages 856–867. Springer-Verlag, 2006.
- [21] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM J. Comput.*, 15(4):1036–1053, 1986.
- [22] S. Olariu and J. Randall. Welsh-powell opposition graphs. *Inf. Process. Lett.*, 31(1):43–46, 1989.
- [23] A. Panconesi and R. Rizzi. Some simple distributed algorithms for sparse networks. *Distributed Computing*, 14(2):97–100, 2001.
- [24] A. Panconesi and A. Srinivasan. On the complexity of distributed network decomposition. *J. Algorithms*, 20(2):356–374, 1996.
- [25] J. S. Turner. Almost all k -colorable graphs are easy to color. *J. Algorithms*, 9(1):63–82, 1988.
- [26] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *38th ACM Symposium on Theory of Computing (STOC)*, pages 681–690, 2006.