Locality Based Graph Coloring

Mario Szegedy
AT&T Bell Labs
Murray Hill, NJ 07974
600 Mountain Ave
rm 2D-151
ms@research.att.com

*Sundar Vishwanathan**
Computer Science Department
University of Chicago
1100 E. 58th Street
Chicago, IL 60637
sundar@cs.uchicago.edu

Abstract

We study the problem of locality based graph coloring. This problem is motivated by the problem of assigning time slots for broadcast in mobile packet radio networks. This problem has also been studied in the context of distributed and parallel graph coloring [4, 6, 9, 8].

In this problem, one has to design a coloring algorithm that assigns a color to a vertex based on the label of the vertex and the labels on its neighbours. Linial proved an upper bound of $O(\Delta^2 \log n)$ and a lower bound of $\Omega(\log\log n)$ on the number of colors needed to locally color an n-vertex graph with maximum vertex degree Δ [9, 8]. His main motivation was that repeated application of local coloring gives a fast algorithm for distributed coloring. He proved that one could get a Δ^2 coloring in $O(\log^* n)$ steps this way.

In this paper we improve upon the bounds for the problem of local coloring. Using a new characterization in terms of a family of set systems we design a randomized algorithm for the problem and prove an upper bound of $O(\Delta \cdot 2^{\Delta} \log\log n)$. An important question left open in Linial's paper was the case of large Δ. The best lower bound was $\Delta + 1$. Linial observed that a result of Erdős, Frankl and Füredi implied that his method cannot be applied to reduce the number of colors to below $\binom{\Delta+2}{2}$. We obtain lower bounds that match the upper bounds within a factor that is poly-logarithmic in terms of these bounds. Of particular interest we have very precise bounds for the case when $\Delta > 2\sqrt{n}$. These bounds are useful to obtain a heuristic estimate on the number of steps necessary to reduce the size of the color set from Δ^2 to $\Delta + 1$, when local coloring algorithms are used iteratively. The number of steps turns out to be $\Theta(\Delta \log \Delta)$.

*Research supported in part by Ketan Mulmuley's Packard Fellowship.

25th ACM STOC '93-5/93/CA,USA
© 1993 ACM 0-89791-591-7/93/0005/0201...$1.50

1 Introduction

Consider a set of mobile packet radio stations. At any given time a station can see at most Δ others. We associate a graph with each such configuration. The set of vertices correspond to the radio stations and two stations are adjacent if they can see each other.

The problem is to schedule transmissions in time to avoid interference. To achieve this the time axis is divided into units called frames. Each frame is divided into slots and each radio station is assigned a slot in which to transmit its information. So, in each frame, a radio station transmits at most one message. In order to avoid interference one has to assign slots to stations so that adjacent stations do not get the same slot. Since the configuration changes it is desirable if each station can decide on its slot based on who its neighbours are.

This problem of slot allocation can be abstracted to the problem of local graph coloring. Informally, we need an algorithm (one for every vertex) that on input a labeled graph looks at the label of a vertex and the labels on its neighbours and decides on the color the vertex gets (a more formal definition can be found in the next section). The combinatorial question here is a bound on the number of colors needed so that such a coloring can be accomplished. The algorithmic question is to devise an algorithm that accomplishes this task.

The problem of transmition scheduling was studied in [1] and [2], where the authors considered a slightly different model from ours.

Another motivation for this problem comes from the problem of coloring in the distributed model of computation. We refer the reader to the introduction section of Linial's paper for a description of this model [8, 9]. One algorithm for distributed coloring is to repeatedly apply a local coloring algorithm at every vertex. The algorithm consists of phases. In the beginning of the i^{th} phase each vertex is assigned a color from the color set $\{1, \ldots, n_i\}$. The vertices then simultaneously compute their own new color from their own and their neighbour's old color (the multiplicity with which a color appears on the neighbours does not count). The new colors are taken from the set $\{1, \ldots, n_{i+1}\}$. Typically $n_{i+1} < n_i$. Throughout the procedure the condition that neighboring vertices do not get the same color should be maintained. Let us denote by $k(n, \Delta)$ the minimal value for k such that there is a local coloring algorithm which reduces the number of colors from n to k for any graph with maximum vertex degree at most Δ. We are interested in local coloring algorithms such that k is strictly less then n when $n > \Delta+1$. For these it is possible to reduce the number of colors to $\Delta + 1$ with finitely many steps of reduction, where Δ is an upper bound on the maximum degree.

Cole and Vishkin were the first to note the power of this technique[4]. They devised an algorithm to 3-color a labeled cycle in $O(log^*n)$ time. Extending this work substantially, Goldberg *et. al.* [6] showed how one could 3-color trees in $O(\log^* n)$ time if each vertex "knows its parent". Both results, though stated for the stronger PRAM model, apply equally well to the distributed setting. Goldberg *et. al.* also showed how to find a $O(\Delta^2)$ coloring in time $O(\log^* n)$, for constant Δ. This bound was extended to work for non-constant Δ by Linial.

Linial was the first to explicitly consider the problem of local coloring. He proves a lower bound of $k(n, \Delta) \geq \log\log n$ even when $\Delta = 2$, i.e. the graph is a union of cycles. He also proves that one needs $\frac{1}{2}\log^* n$ steps to 3-color a cycle. He then proves that $k(n, \Delta) \leq O(\Delta^2 \log n)$. Iterating this algorithm and also using a construction due to Erdös, Frankl and Füredi (for the case that $n = \Delta^3$) for the last iteration, he observes that an $O(\Delta^2)$ coloring can be achieved in $O(\log^* n)$ steps.

We give an improved randomized algorithm to locally color a graph. Our algorithm uses at most $O(\Delta \cdot 2^{\Delta} \log\log n)$ colors. In section 5 we show that this bound is essentially tight for small Δs. Note that Linial's upper bound is better than ours when $\Delta > \Omega(\log\log n)$. When our local coloring algorithm is used iteratively after two iterations of Linial's algorithm, we can $O(\Delta^2)$ color a graph in $\frac{1}{2}\log^* n + O(1)$ steps. This shows that Linial's lower bound is precise up to a constant term (!).

An important question left open in Linial's paper was the case of large Δ. The best lower bound known was the obvious $\Delta+1$. Linial observed that a result of Erdös, Frankl and Füredi [5] implied that his method cannot be applied to reduce the number of colors to $\binom{\Delta+2}{2}$. But it was not known if one could use a better local coloring function to reduce the number of colors substantially. We answer this question in the negative. We prove a lower bound of $n - a \leq k(n, \frac{n}{a} + a)$. As a consequence, the best algorithm that iteratively uses local colorings in the distributed model of computation to obtain a $\Delta + 1$ coloring, is likely to terminate in no less than $\Omega(\Delta \log \Delta)$ steps as indicated by a heuristic argument in Section 8.

2 Preliminaries.

We begin with a formal definition of the problem. Our notation is standard. Let $[n]$ denote the set $\{1, \ldots, n\}$. For V a set $\binom{V}{k}$ would denote the set of all k-subsets of V. A set system is a pair (V, F), where $F \subseteq 2^V$. V is called the base set of the set system. In cases when V's identity is clear, we will also refer to just F as the set system. By the size of the set system we mean $|F|$, i.e. the number of sets in F. We follow Linial's terminology: a k-labeling of a graph $G = (V, E)$ is a 1:1 mapping $f : V \to [k]$. In case $k = |V|$ a k-labeling is called a *labeling.* We will say that a labeling is proper (in which case we refer to it as a coloring) if adjacent vertices do not get the same labels. Throughout this paper, labelings will always be proper. We describe most of our results starting with a proper k-labeling when $k = |V|$, but our results apply for the more general case when k is less.

Let $\mathcal{G} = \mathcal{G}(n, \Delta)$ be a family of graphs on n vertices such that the maximum vertex degree is Δ. Let V be the set of vertices. Let $N_G(v)$ denote the neighbourhood of vertex v in graph G, i.e. the set of vertices adjacent to it in G.

A local k-coloring algorithm for a family of graphs $\mathcal{G}$ is a function $\chi : V \times 2^V \to [k]$ such that $\forall G \in \mathcal{G}, \forall i, j \in V(G)$ if $\{i, j\} \in E(G)$ then $\chi(i, N_G(i)) \neq \chi(j, N_G(j))$. For the rest of the paper $\mathcal{G}(n, \Delta)$ will contain all n-vertex labeled graphs of maximum vertex degree Δ.

Here's a brief description of Linial's proof of the upper-bound. The following lemma (see [9, 5, 7]) is the heart of Linial's proof.

Lemma 1 *For integers $n > \Delta$ there is a family J of n subsets of $\{1, \ldots, 5\lceil\Delta^2 \log n\rceil\}$ such that if $F_0, \ldots, F_\Delta \in J$, then*

$$F_0 \not\subseteq \bigcup_1^{\Delta} F_i.$$

To find a local coloring, associate sets as in the lemma, one with each vertex. Now, given a vertex i with neighbours $j_1, \ldots, j_d$ where $d \leq \Delta$, color i with the minimum color in

$F_i \setminus \cup_{t=1}^{d} F_{j_t}$. The lemma guarantees that such a color exists.

This characterization of the local coloring problem in terms of finding the set systems of the above type is not exact. It can be shown that it is possible to find more efficient local colorings than the ones found this way.

3 Reformulating the problem in terms of set systems

We begin with an exact characterization of the problem in terms of a family of set systems with certain intersection conditions. We first give an informal definition of the equivalent characterization. A more formal definition follows. Consider the complete digraph on n vertices. Let the color-set (the set from which we choose the colors for local coloring) be $[k]$. Associate with each arc $i \to j$ a subset of the color set $[k]$. The sets must satisfy the following two conditions:

1. Consider the sets associated with the arcs leaving any fixed vertex v. Any Δ of these sets must have a non-empty intersection.

2. For any two vertices u and v, the set on the arc $u \to v$ and the set on the arc $v \to u$ must be disjoint.

Here's the more formal version: Find an assignment $\alpha : V \times V \to 2^{[k]}$, where k is some fixed parameter, such that

1. For every fixed $v \in V$ and every $u_1, \ldots, u_\Delta \in V \setminus \{v\}, \alpha(v, u_1) \cap \cdots \cap \alpha(v, u_\Delta) \neq \emptyset$. *[feasibility condition].*

2. For any $u, v \in V, u \neq v, \alpha(u, v) \cap \alpha(v, u) = \emptyset$ *[disjointness condition]*

Theorem 2 *The problem of finding an α for the above problem with parameter k is equivalent to the problem of finding a local k-coloring χ for $\mathcal{G}(n, \Delta)$.*

Proof. We first show, how to find χ given α. Let G be any graph from $\mathcal{G}(n, \Delta)$. For a vertex $v \in V(G)$ define $\chi(v, N_G(v))$ to be the smallest color in the following intersection: $\bigcap_{u \in N_G(v)} \alpha(v, u)$. By the feasibility condition on the αs' this set is non-empty and by the disjointness condition the color that adjacent vertices get are different.

We now have to show how to find α given χ. Suppose that we are given a local coloring function χ over $\mathcal{G}(n, \Delta)$. Define $\alpha(u, v)$ to be the set of colors that χ assigns to u in any graph $G \in \mathcal{G}$ such that $\{u, v\} \in E(G)$. Formally, $\alpha(u, v) = \{c : \exists G \in \mathcal{G}, \chi$ assigns color c to u in G, and $\{u, v\} \in E(G)\}$. We have to verify that the above α satisfies the two conditions.

The Feasibility Condition: Fix a vertex v. Let $u_1, \ldots, u_\Delta$ be Δ other vertices. Consider any graph G such that, in G, v is adjacent to $u_1, \ldots, u_\Delta$. Let c be the color that χ assigns to v in this graph. Clearly c occurs in the sets $\alpha(v, u_i)$ for all i.

The Disjointness Condition: Consider two vertices u and v. We need to prove that $\alpha(u, v)$ is disjoint from $\alpha(v, u)$. Suppose for a contradiction that there is a color c in both of them. Then there is some neighbourhood $N_{G_1}(v)$ of v and a neighbourhood $N_{G_2}(u)$ of u for which χ assigns the color c to both vertices. Also, by the construction, $u \in N_{G_1}(v)$ and $v \in N_{G_2}(u)$. But then consider a graph G_3 where u has neighbourhood $N_{G_2}(u)$ and v has neighbourhood $N_{G_1}(v)$. In this graph then χ would give the color c to both u and v, despite the presence of the edge $\{u, v\}$; a contradiction. ∎

To demonstrate the power of our new characterization we observe that Linial's $\log \log n$ lower bound for $\Delta = 2$ now follows immediately. (We must point out though that Linial proved a stronger bound on an iterated version of local coloring.) To each vertex u associate the set system of the sets of colors adjacent to it, i.e., the set $\mathcal{H}_u = \{\alpha(u, v) \mid v \neq u\}$.

Claim 1 *If $u \neq v$ then $\mathcal{H}_u \neq \mathcal{H}_v$.*

Proof. Consider two vertices u and v. Every set associated with u (i.e. every set in $\mathcal{H}_u$) has a non empty intersection with $\alpha(u, v)$ while $\alpha(v, u)$, a member of $\mathcal{H}_v$, is disjoint from it. ∎

Since the number of subsets of $2^{[k]}$ is at most 2^{2^k} the result follows.

4 Upper Bound

We show the proof of an upper bound $O(\Delta \cdot 2^\Delta \log \log n)$ originated in [10]. We begin with a simple lemma.

Lemma 3 *It suffices to construct a family of set systems, $F_1, \ldots, F_n$, on the color set $[k]$ such that:*

1. *In each F_i, every collection of Δ sets have a non-empty intersection.*

2. *For every pair i and j, there is a set A in F_i and a set B in F_j such that $A \cap B = \emptyset$.*

We leave it as an easy exercise to show that this is an equivalent formulation of our problem. Here is the algorithm to construct the family of set systems. Assume for ease of description that n is a power of 2. Consider a binary tree of depth $\log n$ with n leaves. Label the leaves arbitrarily from 1 to n. With each vertex of the tree we will associate a subset of $[k]$. The set system F_i

will consist of all the sets associated with the vertices along the path from the root to the leaf i.

Suppose that u and v were sons of a common parent p. We will assure, during the course of the construction, that the sets associated with vertices u and v are disjoint. This takes care of requirement 2 since for set systems F_i and F_j, if w is the last common vertex in the paths from the root to the leaves i and j then the sets associated with the sons of w will provide witnesses for verifying the disjointness condition.

We use randomization to assign the sets to the vertices in the tree. For a level ℓ of the tree, flip a coin for each element of $[k]$. This splits $[k]$ into two disjoint sets H_ℓ (elements for which we got heads) and T_ℓ (elements for which we got tails). We associate either H_ℓ or T_ℓ with nodes at level ℓ such that if one son of a parent gets H_ℓ then the other gets T_ℓ. The coin tosses for different levels are independent.

We now prove that such a construction works with overwhelming probability when $k = O(\Delta \cdot 2^\Delta \log\log n)$.

A bad event corresponds to some Δ sets not intersecting for some set system F_i. The number of ways in which a bad event happens is at most $2^\Delta\binom{\log n}{\Delta}$. This is because all possible bad events can be chosen by first choosing Δ levels of the tree and then choosing one of the 2^Δ possible intersections from sets at these levels. The probability of a bad event happening, once the Δ sets are chosen is bounded in the lemma below.

Lemma 4 *Suppose we pick Δ subsets of $[k]$ at random by flipping a coin and putting an element in a set if the coin toss was a head. Then the probability that these sets are disjoint is at most $e^{-k/2^\Delta}$.*

Proof. The probability that a fixed element is not in the intersection is exactly $(1-\frac{1}{2^\Delta})$. Since the coin-tosses for the different elements are independent the probability that none of the elements are in the intersection is at most $(1-\frac{1}{2^\Delta})^k \le e^{-k/2^\Delta}$. ∎

Hence the probability that the set systems we have chosen is bad is at most $e^{-k/2^\Delta}2^\Delta\binom{\log n}{\Delta}$. We see that choosing k as prescribed makes this probability small and hence the method works with high probability. ∎

We make two comments on the upper bound.

1. Strictly speaking the algorithm is not probabilistic, but rather the method with wich an appropriate set system can be found.

2. For $\Delta = 2$ an explicit construction is possible. Choose k, even, such that $\binom{k}{k/2} > 2\log n$. Now for the ith level, pick sets A_i and $[k]\setminus A_i$ such that $|A_i| = k/2$ and A_i has not been used at previous levels. This construction works because any two sets of size $k/2$ intersect as long as one is not the complement of the other.

5 Lower bounds

Thus far we have shown upper bounds. The two known upper bounds $5\Delta^2\log n$ and $\Delta 2^\Delta \log\log n$ will be denoted by $B_1(n,\Delta)$ and by $B_2(n,\Delta)$ respectively. We remark that these bounds are based on very different construction ideas [9], [10]. When $\Delta < \log\log n$ then $B_2(n,\Delta)$ is better, otherwise $B_2(n,\Delta)$.

In this section we show that no significant improvement on these bounds can be made. When Δ approaches n however, then more subtle improvements on the bounds become interesting. Section 7 gives very tight bounds for the case of large Δs.

Let us denote the number of colors of an optimal construction for n vertices and degree Δ by $k(n,\Delta)$. The main result of this section is:

Theorem 5

$$k(n,\Delta) \le \min(B_1(n,\Delta), B_2(n,\Delta), n) \le k(n,\Delta)(\log k(n,\Delta))^3. \quad (1)$$

Proof: We have to prove the r.h.s. inequality of (1). Define $d_{thres} = \log\log n - \log\log\log n$. The function $k(n,\Delta)$ is monotone increasing in both of its variables. We use different estimates for different ranges of Δ. If $\Delta \le d_{thres}$ then we use Theorem 6. If $7d_{thres} \ge \Delta \ge d_{thres}$ then the monotone increasing property of $k(n,\Delta)$ gives a $\log n$ lower bound. If $\sqrt{n} \ge \Delta \ge d_{thres}$ then we use Theorem 7. If $\Delta \ge \sqrt{n}$ then again, monotonicity gives us an $n/\log(n)$ lower bound. It is an easy calculation that inequality 1 holds for the above four different ranges of Δ. ∎

Theorem 6 *If $\Delta < d_{thres}$ then:*

$$k(n,\Delta) \ge 2^{\Delta-4}\log\log n.$$

Theorem 7 *If $\sqrt{n} \ge \Delta \ge 7d_{thres}$ then*

$$k(n,\Delta) \ge \frac{\Delta^2}{(\log\Delta + \log\log\Delta)^2\log\Delta}\log n.$$

Proof of Theorems 6 and 7: Assume that we have a construction $\alpha(i,j)$ $(1 \le i,j \le n)$ which satisfies the conditions in Section 3 on a color set $[k]$. Let β be a parameter, which is $1/2$ if Δ is below $d_{thres} = \log\log n - \log\log\log n$ and $\frac{(2\log\Delta + \log\log n)}{\Delta}$ otherwise. We partition the vertex set into two classes:

Class 1: A vertex v belongs to the first class if there is a non empty set H_v such that for every vertex $x \ne v$:

$|H_v \setminus \alpha(v,x)| < \beta|H_v|$. For instance, if a single color c occurred in every set $\alpha(v,x)$ then one could take $H_v = \{c\}$.

Class 2: Every other vertex.

For the sake of analysis, we distinguish two cases depending on whether the number of vertices in class 1 is larger or smaller than the number of vertices in class 2. Within each case we will also analyse the $\Delta \leq d_{thres}$ and the $\Delta \geq 7d_{thres}$ cases separately.

Case 1: The number of vertices that belong to the first class is at least $n/2$.

Consider first, the case when $\Delta < d_{thres}$. We claim that in this case $H_u \neq H_v$ for $u \neq v$. This is because more than 1/2 of the elements in the set H_u is contained in $\alpha(u,v)$ and more than 1/2 of the elements in the set H_v is contained in $\alpha(v,u)$. Since $\alpha(u,v) \cap \alpha(v,u) = \emptyset$, $H_u \neq H_v$. Hence the base set $[k]$ has size at least $\log(n/2)$ and the estimate in Theorem 6 follows.

Consider now, the case when $\Delta \geq 7d_{thres}$. Classify these $n/2$ or more H_v's according to their size. (Loosely– consider even equal H_v's associated with different vertices to be different.) For some i ($1 \leq i \leq \log n$) the number of H_is that have size between 2^i and 2^{i+1} is at least $n/\log n$. Concentrate only on these H_vs. If H_v and H_w are two such sets, then $|H_v \cap H_w|$ is at most $3\beta|H_v|$. This is because $|H_v \cap H_w| \leq |H_v \setminus \alpha(u,v)| + |H_u \setminus \alpha(v,u)| + |\alpha(u,v) \cap \alpha(v,u)| \leq \beta(|H_v| + |H_u|) \leq 3\beta|H_v|$. That a system of $m = O(n/\log n)$ sets with the above condition on their pairwise intersections should be supported on a basis of size at least $\min(m, \Omega(\frac{1}{-\log \beta(\beta)^2} \log m)$ will be the topic of the next section. The estimate is clearly sufficient to vindicate Theorem 7.

Case 2: The number of vertices that belong to the second class is at least $n/2$.

First, for every vertex v in class 2, using a procedure that we describe below, we construct a set $X(v)$. $X(v)$ will be the intersection of $\Delta - 2$ sets of the form $\alpha(v,x)$. Define the set $X_1(v)$ to be any of the sets $\alpha(v,x)$. We will construct sets $X_i(v)$, deriving set $X_{i+1}(v)$ from set $X_i(v)$, as given below. $X(v)$ will be the set $X_{\Delta-2}(v)$. Suppose that we have determined the sets $X_1(v), \ldots, X_k(v) = X_k$. We now choose an $x_{k+1} = x_{k+1}(v)$ such that $|X_k \setminus \alpha(v, x_{k+1})| \geq \beta|X_k|$. Note that such an x_{k+1} exists; otherwise v would be a class 1 vertex with $H_v = X_k(v)$. We then set $X_{k+1} := X_k \cap \alpha(v, x_{k+1})$. Set $X(v)$ is non-empty because of the feasibility condition. Note that $|X_{k+1}(v)| = |X_k(v) \cap \alpha(v, x_{k+1})| = |X_k(v)| - |X_k(v) \setminus \alpha(v, x_{k+1})| \leq (1-\beta)|X_k(v)|$. Hence

$$|X(v)| \leq |X_1(v)|(1-\beta)^{\Delta-3}. \tag{2}$$

The above inequality immediately gives Theorem 7 when $\Delta \geq 7d_{thres}$ using the obvious $|X_v| \geq 1$. (Recall that in this case $\beta = \frac{(2\log\Delta + \log\log n)}{\Delta}$.)

We now turn to the analysis of the $\Delta < d_{thres}$ case. Our goal here is to extend (substantially!) the "$\log\log n$" lower bound proof for the $\Delta = 2$ case. So, mimicing the main idea of the $\Delta = 2$ case, we associate, with each vertex v belonging to the second class, a set system $\mathcal{S}_v$, such that the set systems associated with different vertices are different. For each such a v we construct a set system $\mathcal{S}_v$ as follows. The set system $\mathcal{S}_v$ will consist of all the sets $X(v) \cap \alpha(v,x)$, where x runs through all the vertices different from v. We now claim that if v and w are vertices in Class 2 then the systems $\mathcal{S}_v$ and $\mathcal{S}_w$ are distinct. Indeed, $\alpha(v,w)$ has non-empty intersection with any element of $\mathcal{S}_v$ (which are of the form $X_v \cap \alpha(v,x)$), whereas its intersection with $X_w \cap \alpha(w,v)$, an element of the other set system is empty. We will use the following simple combinatorial lemma:

Lemma 8 *If $(X_1, \mathcal{S}_1), (X_2, \mathcal{S}_2), \ldots, (X_m, \mathcal{S}_m)$ are set systems such that the $\mathcal{S}_i$'s are distinct then either $\max_i |X_i| \geq \lg\lg m/2$ or $|\cup_i X_i| \geq \lg m/2$.*

Proof. Assume that $\max_i |X_i| < \lg\lg m/2$. This implies that there are at least $m/2^{2^{\lg\lg m/2}}$ distinct X_i's. Which means that the size of their union is at least $\lg(m/2^{2^{\lg\lg m/2}}) > \lg m/2$. ∎

Applying the lemma to our case, we see that either the largest of X_v has size at least $\frac{\log\log(n/2)}{2}$, or else, the number of colors is at least $\frac{\log(n/2)}{2}$. In the first case we use Inequality 2 to get Theorem 6, whereas in the second case Theorem 6 follows immediately. ∎

6 Constant weight codes

Here we deal with the problem raised in the previous section. The problem is that given an integer m and a ratio $0 < \gamma < 1$ determine the lower bound on the size of X for a set sysem $(X, \mathcal{S})$ with the following properties:

- $|\mathcal{S}| = m$.
- There is an integer w such that for each $H \in \mathcal{S}$ $w \leq |H| \leq 2w$.
- If $H_1, H_2 \in \mathcal{S}$ then $|H_1 \cap H_2| \leq \gamma|H_1|$.

We simplify the problem by leaving out elements from each $H \in \mathcal{S}$ such that their size becomes w uniformly. This may increase γ by a factor of at most 2. We then recast this problem in the 'coding theory' framework by considering the characteristic vectors of these sets. Let us denote by $A(k,d,w)$ the maximal possible number of binary vectors of length k, Hamming distance at least d apart, and constant weight w. This problem is studied extensively in [3]. Our task now translates to: Give a lower bound on k under the assumption that $A(k,d,w) \geq m$ and $d = 2\lceil w(1-\gamma)\rceil$ for some fixed γ. (Note that the Hamming distance between the characteristic vectors of two sets H_1 and H_2 is exactly $|H_1 \triangle H_2| = |H_1| + |H_2| - 2|H_1 \cap H_2|$.)

To this end we use a fairly trivial upper bound on A:

$$A(k, 2\delta, w) \leq \frac{k(k-1)\cdots(k-w+\delta)}{w(w-1)\cdots\delta}. \qquad (3)$$

First fix k and optimize the above expression over all w under the condition that $\delta = \lceil w(1-\gamma)\rceil$. If $\delta = w$ then the optimal value for w is 1. In this case we can lower bound k by m. Otherwise the condition $w > \lceil w(1-\gamma)\rceil$ implies that $w \geq \gamma^{-1}$. (Note that w is an integer.) One can show that in this case the expression increases in a monotone fashion until w reaches k/e. We show that $w \leq 4k\gamma$. We have $\gamma^{-1} \leq n/\log n \leq m$. Consider now $1/(2\gamma)$ sets, each of size w with pairwise intersection at most γw. If we leave out the union for all the other sets from any one of these sets, then we are still left with at least $w - (w\gamma)/(2\gamma) = w/2$ elements. So each set contains at least $w/2$ elements that do not appear in any of the other sets. This immediately gives that $k \geq \gamma^{-1}w/4$. Thus we can estimate the optimal value for w by $w = O(\gamma k)$. But then from Inequality 3:

$$m \leq \prod_{i=0}^{\lfloor \gamma w\rfloor} \frac{k-i}{w-i} = \left(\frac{k}{\gamma k}\right)^{O(\gamma^2 k)}$$

Hence $k \geq min(m, O(\frac{1}{(-log\gamma)\gamma^2}\log m))$.

We mention that the upper bound construction of Linial can be built using constant weight codes with parameters $\gamma = 1/\Delta$ and $A(k,d,w) \geq n$.

7 More precise upper and lower bounds for large values of Δ

Consider the case when Δ is very large compared to n. The range of our interest is when Δ is at least $2\sqrt{n}$. It is easy to show that when $\Delta = n-1$, $k = k(n,\Delta) = n-1$. It is not hard either to show that if $\Delta = n-2$, then $k = n-1$. For $n-2 \geq \Delta \geq 2\sqrt{n}$ the following lemma yields good lower bounds:

Lemma 9 *If $\Delta \geq n/a + a$ then k is at least $n-a$.*

Proof: Partition the vertices into two classes. Class 1 contains those vertices u for which there is an elment p_u that is common in all $\alpha(u,v)$, where v ranges through all the vertices $v \neq u$. Every other vertex belongs to class 2. We argue that there are at most $a-1$ vertices that belong to class 2. Indeed, assume that $u_1,\ldots,u_a$ are all in class 2. For $1 \leq i \leq a$ define $H_i = \bigcap_{1\leq i\neq j\leq a}\alpha(u_i,u_j)$. For $1 \leq i \leq a$ the sets H_i are disjoint. Thus there is some i such that $|H_i| \leq n/a$. We now define sets $Y_0, Y_1, Y_2, \ldots$ such that Y_j is the intersection of exactly $j+a$ sets of the form $\alpha(u_i,x)$. Define $Y_0 = H_i$. Once Y_l is defined, define Y_{l+1} as $Y_l \cap \alpha(u_i,v)$, for such a v that $|Y_{l+1}| < |Y_l|$. Such a v exists because u_i is in class 2. We stop when Y_l becomes the empty set. The number of steps (i.e. l) is at most $|Y_0| = |H_i| \leq n/a$. Note that Y_l is the intersection of at most $\frac{n}{a} + a$ of $\alpha(u_i,v)$s. But this violates the feasibility condition since $\Delta \geq \frac{n}{a} + a$. ∎

On the other hand:

Theorem 10 *There is a construction for $\Delta \leq \lfloor \frac{n}{a} - 3\rfloor$ that uses $n-a$ colors.*

Proof: Set the color set to be $[k] = [n-a]$. To the first a vertices $u_1,\ldots,u_a$ we assign disjoint sets $T_1,\ldots,T_a$, each of size $\lfloor n/a - 1\rfloor$. The corresponding set systems $(T_i, \mathcal{H}_{u_i})$ contain exactly those subsets of their basis set that have size $\lfloor n/a - 2\rfloor$. For the remaining vertices $u_a,\ldots,u_n$ the set systems are disjoint singletons: $(H_{u_i} = \{i-a\}, a < i \leq n)$. One can see that the conditions of Lemma 4 in Section 4 hold.

8 Iterative use of local coloring

R. Cole and Uzi Vishkin in [4] were the first to call attention to a type of algorithm that could break symmetry in a distributed environment without using randomization. They refer to the paradigm that these algorithms follow as *accelerated cascades.* To break the symmetry, each processor uses its own label as well the label of its neighbors. In each phase the labels are recomputed and the label set shrinks. However, throughout the computation neighbouring vertices never receive identical labels. This gives an illusion, locally to each processor, that the labeling is proper, i.e. each node has a different label assigned to it. If, the size of the label set shrinks exponentially in each phase, then the number of phases is $\log^* n + O(1)$. This allows them to 3-color a cycle in $\log^* n + O(1)$ phases. Our upper bound construction for the case $\Delta = 2$ yields an algorithm that 3-colors a cycle in $(1/2)\log^* n + O(1)$ phases which, by Linial's lower bound, is tight to within the $O(1)$ term.

The iteration is made possible by the following lemma:

Lemma 11 *Let G be graph with a proper coloring $\chi : V(G) \rightarrow [1,\ldots,k_1]$. Then we can obtain another coloring χ' using the set systems $\alpha(i,j)$ $(1 \leq i,j \leq k_1)$ with* *properties described in Section 3, to create χ' by the following procedure: For a vertex v denote $i = \chi(v)$ and $j_1 = \chi(u_1)$,..., $j_\Delta = \chi(u_\Delta)$. Then $\chi'(v)$ is defined as the smallest element of $\bigcap_{s=1}^{\Delta} \alpha(i, j_s)$. Coloring χ' will be proper.*

To see that the lemma holds we observe that χ' will be proper because of the "disjointness condition" (see section 3). When a local coloring procedure is iterated, is each step the number of colors is reduced. Typically one step reduces a color set of size n to one of size $k(n, \Delta)$. There is a possibility however, that after a few steps of iteration we arrive at a very special type of coloring that can be very efficiently reduced in the steps thereafter. Assuming that this does not happen, the results of the previous section give the following theorem:

Theorem 12 (heuristic) *Let $1 \leq b < a \leq \Delta/2$. To decrease the number of colors from $a\Delta$ to $b\Delta$ it takes $\Theta(\Delta \log(a/b))$ steps. In particular, to decrease the number of colors from $\Delta^2/2$ to Δ requires $\Theta(\Delta \log \Delta)$ steps.*

9 Open Problems

We list the following open problems:

- Find an explicit version of the upper bound construction given in this paper.
- We have a set-system characterization when one does local coloring after seeing only the immediate neighbourhood of a vertex. Is there such a characterization of the problem when one looks at the label on a vertex v and also at the subgraph induced by the set of vertices that are at a distance ℓ from v to decide on the color of v. Such a characterization could then, perhaps, be used to find the number of steps required to find a $\Delta + 1$ coloring in the distributed model of computing–a major open question in that field.

Acknowledgements. We are grateful to Andras Farago for bringing this problem to our attention, and, specially for telling us about the application of local coloring to mobile packet radio networks. A special thanks to Laci Babai for his constant encouragement and guidance. Thanks are due to Bela Bollobas for many insightful discussions on this subject.

References

[1] N. Alon, A. Bar-Noy, N. Linial and D. Peleg, A Lower Bound for Radio Broadcast *Journal of Computer and System Sciences*, 43: pages 290-298, 1991

[2] N. Alon, A. Bar-Noy, N. Linial and D. Peleg, Single round simulation on Radio Networks *Journal of Algorithms* 13: pages 188-210, 1992.

[3] A.E. Brouwer, J. B. Shearer, N.J. Sloane, W. D. Smith, A new table of constant weight codes, *IEEE Transactions on Information Theory* vol. 36 (nov 1990), 1334–1380.

[4] R. Cole and U. Vishkin. Deterministic coin tossing and accelerating cascades: micro and macro techniques for designing parallel algorithms. In *STOC*, pages 206–219, 1986.

[5] P. Erdös, P. Frankl, and Z. Füredi. Families of finite sets in which no set is covered by the union of r others. *Israel J. Math.*, 51:79–89, 1985.

[6] A. Goldberg, S. Plotkin, and G. Shannon. Parallel symmetry-breaking in sparse graphs. In *STOC*, pages 315–324, 1987.

[7] D. J. Kleitman, J. Shearer, and D. Sturtevant. Intersections of k-element sets. *Combinatorica*, 1:381–384, 1981.

[8] N. Linial. Distributive graph algorithms–global solutions from local data. In *FOCS*, pages 331–335, 1987.

[9] N. Linial. Locality in distributed graph algorithms. *SIAM J. COMPUT*, 21:193–201, 1992.

[10] Sundar Vishwanathan. Locality Based Graph Coloring. *Preprint*