On the Hardness of 4-coloring a 3-colorable Graph*

Venkatesan Guruswami† Sanjeev Khanna ‡

Abstract

We give a new proof showing that it is NP-hard to color a 3-colorable graph using just four colors. This result is already known [19], but our proof is novel as it does not rely on the PCP theorem, while the one in [19] does. This highlights a qualitative difference between the known hardness result for coloring 3-colorable graphs and the factor n^ϵ hardness for approximating the chromatic number of general graphs, as the latter result is known to imply (some form of) PCP theorem [3].

Another aspect in which our proof is different is that using the PCP theorem we can show that 4-coloring of 3-colorable graphs remains NP-hard even on bounded-degree graphs (this hardness result does not seem to follow from the earlier reduction of [19]). We point out that such graphs can always be colored using $O(1)$ colors by a simple greedy algorithm, while the best known algorithm for coloring (general) 3-colorable graphs requires $n^{\Omega(1)}$ colors. Our proof technique also shows that there is an $\varepsilon_0 > 0$ such that it is NP-hard to legally 4-color even a $(1-\varepsilon_0)$ fraction of the edges of a 3-colorable graph.

1 Introduction

The graph coloring problem is to assign colors to vertices of a graph G such that no two adjacent vertices receive the same color; such a coloring is referred to as a *legal* coloring of G. The minimum number of colors required to do a legal coloring is known as the *chromatic number* of G, and is denoted $\chi(G)$. Graph coloring is a fundamental and extensively studied problem, which besides its theoretical significance as a canonical NP-hard problem [17], also arises naturally in a variety of applications including register allocation and timetable/examination scheduling.

Coloring a graph G with the minimum number $\chi(G)$ of colors is NP-hard [17], so the focus shifts to efficiently coloring a graph with an approximately optimum number of colors. Garey and Johnson [10] proved that it is NP-hard to approximate the chromatic number within a factor of $(2-\epsilon)$ for any $\epsilon > 0$. The best known algorithm for general graphs appears in [14] and colors a graph using a number of colors that is within a factor of $O(n(\log\log n)^2/\log^3 n)$ of the optimum (here and elsewhere n refers to the number of vertices in the graph). There is strong evidence that one cannot do substantially better than this for general graphs, as the recent connection between Probabilistically Checkable Proofs (PCPs) and hardness of approximations [7, 2, 1], has led to strong hardness results for graph coloring also. The first such result was established by Lund and

*A preliminary version of this paper appeared in the *Proceedings of the 15th Annual IEEE Conference on Computational Complexity*, July 2000.

†Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139. Email: venkat@theory.lcs.mit.edu. Part of this work was done when the author was visiting Bell Labs, Murray Hill.

‡Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104. This work was done when the author was at Bell Labs, Murray Hill. Email : sanjeev@cis.upenn.edu.

Yannakakis [20] who proved that chromatic number is hard to approximate within n^ϵ for some constant $\epsilon > 0$. Feige and Kilian [8], using the powerful PCP constructions due to Håstad [15], prove that unless NP $\subseteq$ ZPP one cannot approximate the chromatic number within a factor of $n^{1-\epsilon}$ for any constant $\epsilon > 0$.

However, *none* of these inapproximability results apply to the case when the input graph is k-colorable for some small constant k. Indeed, better performance guarantees are known in this case. For instance, a polynomial time algorithm that colors 3-colorable graphs using $\tilde{O}(n^{3/14})$ colors is known [23, 5, 16, 6]. It is known that for every constant h there exists a large enough constant k such that coloring k-colorable graphs using kh colors is NP-hard [20, 19]; it is however not known if the order of quantifiers above can be reversed. Khanna, Linial and Safra [19] proved that it is NP-hard to color a 3-colorable graph using only 4 colors, and to this date no improvement to this hardness result has been obtained.

Our Results. Our main result in this paper is a new proof of the above result of [19], stated formally below:

Theorem 1 (Main Theorem) *It is NP-hard to color a* 3*-colorable graph with only four colors.*

The proof of Khanna *et. al.* [19] uses the result that MAX CLIQUE is NP-hard to approximate within a factor of two, a consequence of the PCP theorem [2, 1]. An important distinguishing aspect of our proof is that it *does not require the PCP theorem* and only relies on the NP-hardness of the MAX CLIQUE problem. The hardness for 3-colorable graphs is the most intricate of the results in [19], and has not been improved upon or simplified ever since. Our work represents the first progress on this important problem after the result of [19], and one which will hopefully spur further improvements. Not relying on PCP machinery implies that this hardness result could have been obtained almost three decades ago, long before the arrival of the PCP theorem. In contrast the hardness result (for approximating within n^ϵ for example) for general graph coloring implies some form of PCP [3]; our result therefore also highlights a qualitative difference between the hardness of general graph coloring and coloring 3-colorable graphs.

As in essentially all previous reductions showing hardness of graph coloring, our reduction too starts from the hardness of INDEPENDENT SET (MAX CLIQUE): it transforms an instance G of INDEPENDENT SET to an instance H of graph coloring such that a large independent set in G translates into a small collection of (in our case three) independent sets in H which together cover all vertices in H. But in addition, our proof is based only on local gadgets and easily leads to the hardness of 4-coloring even bounded degree instances of 3-colorable graphs, albeit only by resorting to the PCP theorem:

Theorem 2 *There is a constant Δ such that given a graph 3-colorable graph with maximum degree at most Δ, it is NP-hard to color it using just 4 colors.*

Note that since such graphs can be colored using $O(1)$ colors (in fact $(\Delta + 1)$ colors) by a simple greedy algorithm, while the best algorithm for general 3-colorable graphs uses $n^{\Omega(1)}$ colors, this hardness result is stronger than that of Theorem 1. Another strengthening of Theorem 1 which the degree-bounded result enables us to deduce is the following:

Theorem 3 *There is a constant $\varepsilon_0 > 0$ such that it is NP-hard to, given a graph G, distinguish between the case when G is* 3*-colorable and when any* 4*-coloring of G miscolors at least an ε_0-fraction of its edges.*

Both of these results do not seem to follow from the proof technique of [19] and therefore appear to be new to our paper. Note that the latter claim also generalizes the result of Petrank [22] which shows that there is an $\varepsilon > 0$ such that it is NP-hard to legally color a $(1 - \varepsilon)$ fraction of the edges of a 3-colorable graph using only 3 colors.

We also note, using standard PCP techniques, a uniform hardness result that χ^h-coloring a graph G with $\chi(G) = \chi$ is NP-hard for some (fixed) constant $h > 1$, for chromatic numbers χ in the range $c \le \chi \le n^\delta$ for some constants $c, \delta > 0$. A similar result has earlier been shown by Fürer [9], but our proof uses the simpler and more transparent transformations in [19], and also does not use a randomized reduction (while the result in [9] is obtained under the stronger assumption of NP $\not\subseteq$ ZPP).

Inapproximability Results and PCPs. In light of our main result, it is natural to ask how far non PCP techniques can go in proving hardness results for coloring 3-colorable graphs. It turns out that an inapproximability factor of $\Omega(\log n)$ does imply some form of PCP. This is by a result of Blum [4] (see also [3]) which shows that if coloring a 3-colorable graph using $c \log n$ colors is hard for every constant c, then for every $\epsilon > 0$, it is hard to approximate MAX CLIQUE within a factor of $n^{1-\epsilon}$; and using the "reversal" of the FGLSS connection presented in [3], this implies the PCP theorem (in fact a very strong version of it, see [3] for details).[1] It, however, seems entirely possible that any $o(\log n)$ hardness bound can be proved for coloring 3-colorable graphs without resorting to PCP techniques.[2] We hope our work will spur further investigation into these questions that will eventually lead to improving the current results on the complexity of coloring 3-colorable graphs.

Expanding the scope of our investigation, it is natural to ask which inapproximability results really require PCPs. It is known for example that PCPs are inherent to obtaining (at least strong) hardness results for approximating MAX SAT, MAX CLIQUE, Chromatic Number and Vertex Cover. Recent work in [12] and [18] proves strong (in fact near-tight) inapproximability results for Disjoint Paths and Longest Path problems without requiring PCPs; prior results for these problems always began with the PCP theorem, and yet turned out to be weaker. Together with our result, these raise similar questions about the hardness results for certain other fundamental problems like Set Cover, Nearest Codeword Problem, Shortest Vector Problem, etc, which while currently relying on the PCP theorem, are not known to *provably require* PCPs. In each case it is interesting to see if a reverse connection to PCPs exists or if PCPs are only an artifact of the current proof techniques. Even more ambitiously, one can ask what aspect(s) of an optimization problem cause its inapproximability results to necessarily imply some non-trivial PCP constructions.

Notation. We use the standard notation to denote graph-theoretic parameters. For a graph G, we denote by $\chi(G)$, $\alpha(G)$, $\omega(G)$ and $\theta(G)$, the chromatic number of G, the size of a largest independent set in G, the size of a largest clique in G and the clique cover number of G (the minimum number of cliques to cover all the vertices of G) respectively. Clearly $\alpha(G) = \omega(\bar{G})$ and $\chi(G) = \theta(\bar{G})$ where $\bar{G}$ is the complement of the graph G.

Organization. We present the proof of our main theorem (Theorem 1) in Section 2. Section 3 describes the hardness result for bounded degree 3-colorable graphs and sketches the proof of Theorem 3. Section 4 describes the uniform χ^h hardness bound for coloring χ-chromatic graphs.

[1] Since the reduction from 3-coloring to finding large cliques is only a Turing reduction, strictly speaking we can only conclude that every language in NP Turing reduces to a language in a certain PCP class.

[2] Actually, such a hardness result *does* imply the existence of very good *covering PCPs*, a notion recently introduced in [13] for the purpose of studying minimization problems like Coloring. Constructing a "good" covering PCP without resorting to the PCP theorem appears very difficult, so such a PCP-free hardness result for coloring 3-colorable graphs might be hard to come-by.

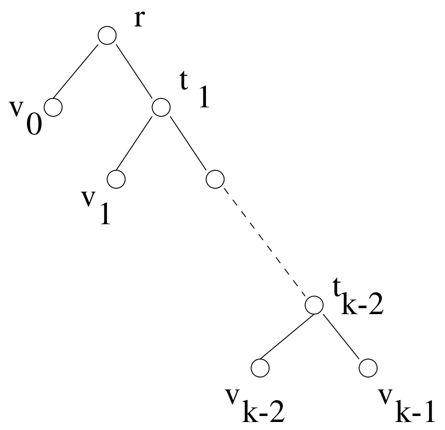


Figure 1: High-level structure of each T_i

2 Proof of the Main Theorem

We describe a reduction from the INDEPENDENT SET problem. Specifically, we start with instances of the following form: we are given a graph G along with a partition of the vertices of G into r cliques R_i, $1 \leq i \leq r$, each with exactly k vertices. Clearly, $\alpha(G) \leq r$. It is NP-hard to determine if $\alpha(G) = r$ on instances with this structure even when the partition into the R_i's is given as part of the input. This hardness even holds with $k = 3$ – the standard reduction for NP-hardness of INDEPENDENT SET in fact produces such instances [11]. Thus the proof of Theorem 1 only requires us to consider the case $k = 3$. However, we will present here a construction for any arbitrary k. This is because the starting point for Theorems 2 and 3 are INDEPENDENT SET instances that are generated by PCP constructions and k is a suitably large constant in this case.

Starting with such an instance G, we construct (in polynomial time) a graph H which will have the property that $\chi(H) = 3$ if $\alpha(G) = r$ and $\chi(H) \geq 5$ otherwise. This will clearly prove Theorem 1.

2.1 Overview of the Reduction

Let G be a graph with vertices partitioned into r cliques R_i, $1 \leq iler$, with exactly k vertices in each clique, i.e., let $R_i = \{v_{i,0}, \ldots, v_{i,k-1}\}$ for $1 \leq i \leq r$. The graph H comprises of r "tree-like" structures, say $T_1, \cdots, T_r$, one for each clique R_i of G, together with a specific interconnection pattern between the leaves of the different tree-structures based on the adjacency of vertices in G. There are two key properties satisfied by the construction of the T_i's:

- Any 4-coloring of a T_i can be interpreted as "selecting" a unique vertex $v_{i,p}$ in the clique R_i of graph G (Section 2.2).
- The edges between the T_i's are such that no 4-coloring is feasible if two vertices that are adjacent in G are selected from two different trees (Section 2.3).

In other words, any 4-coloring of H can be interpreted as selecting a vertex in each of the r cliques R_i of G such that the selected vertices induce an independent set of size r in G, ensuring that if $\alpha(G) < r$, then in fact $\chi(H) > 4$. The other part, viz. H is 3-colorable if $\alpha(G) = r$, will also be easily seen to hold for our reduction.

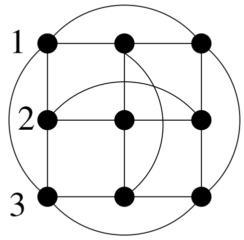


Figure 2: The basic template

2.2 The Structure of each T_i

Each T_i will have the structure of a binary tree with k leaves, $\{v_{i,j} : 0 \leq j < k\}$, one for each of the k vertices of G in the clique R_i (see Figure 1). It also has $(k-1)$ additional internal nodes $\{t_{i,j} : 0 \leq j < k-1\}$ with $t_{i,0}$ being the "root" r_i; by $t_{i,k-1}$ we mean the leaf node $v_{i,k-1}$. (The subscript i is omitted in Figure 1 for sake of readability. The exact "shape" of the tree T_i is not important; any binary tree with k leaves and with all internal nodes having exactly two children will suffice for our purposes.) Each individual node of T_i itself comprises of the template shown in Figure 2. This basic template, denoted H_{basic}, may be viewed as a 3×3 grid such that the vertices in each row and in each column of the grid induce a 3-clique. The vertices in the first column of any such template are referred to as *ground vertices* and are in fact *shared* across all such templates in all the tree-structures. Since the ground vertices form a clique, any legal coloring will assign three *distinct* colors to them; we refer to these colors as 1, 2 and 3.

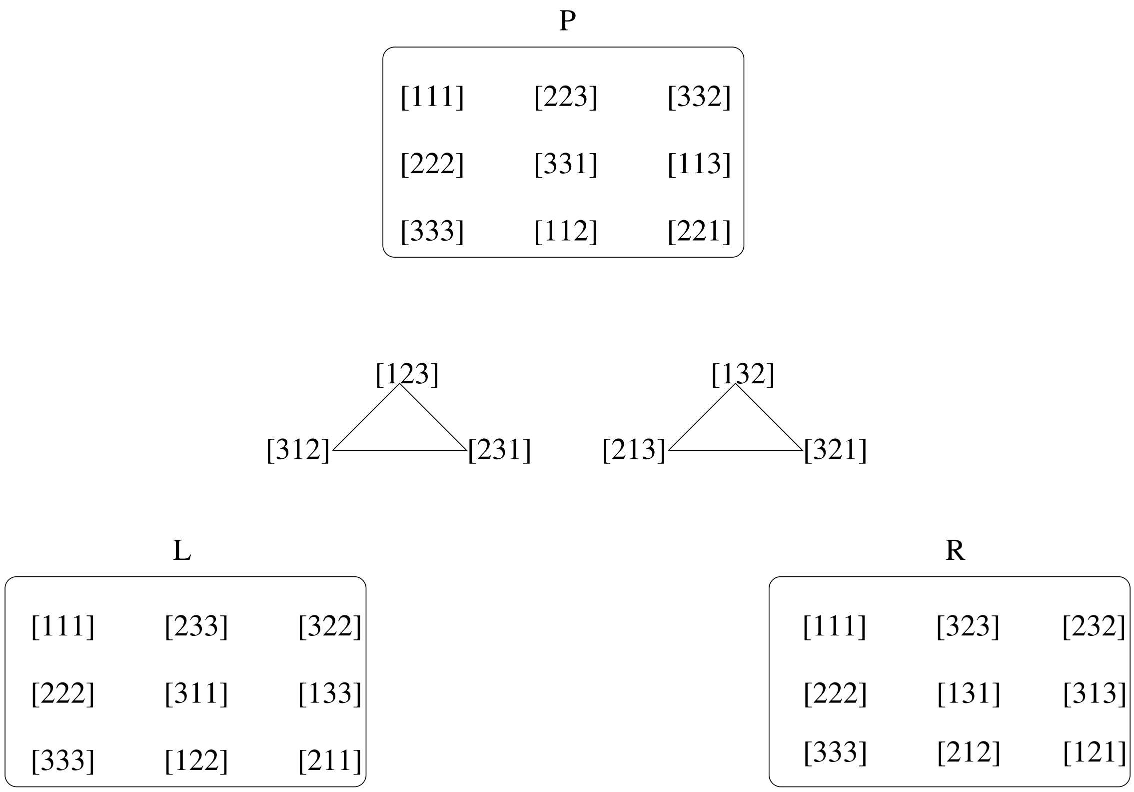
The connection pattern between the template at an internal node $t_{i,p}$ and its children templates $v_{i,p}$ and $t_{i,p+1}$ is best understood by the schematic depicted in Figure 3. (Nodes P, L and R will play the roles of $t_{i,p}$, $v_{i,p}$ and $t_{i,p+1}$ respectively.) In addition to the templates at these nodes, there are two 3-cliques that are connected to templates at $t_{i,p}, v_{i,p}$ and $t_{i,p+1}$ via appropriate edges. All nodes in the schematic are labeled as 3-tuples of the form $\langle xyz \rangle$ where $x, y, z \in \{1, 2, 3\}$. The edges (*not shown*) between the various vertices are given by the simple rule: two vertices are adjacent if and only if their labels *differ in all three coordinates*.

2.2.1 Node Selection

A node of the tree is called *selected* if at least one of the three rows in its template has colors which reading from left to right form an even permutation of $\{1, 2, 3\}$ (i.e the first row has colors 1,2,3; the second one has 2,3,1; or the third one has 3,1,2). Similarly, we say that a row is *not selected* if at least one of the three rows in its template has colors which reading from left to right form an odd permutation. It is easy to see that in any legal 4-coloring a node can never be simultaneously labeled selected as well as not selected. Moreover, in any 4-coloring a node is always either selected or not selected.

2.2.2 Enforcing Selection of a Leaf Node

Our goal now is to enforce that for any legal 4-coloring of the tree-structure T_i, at least one leaf node is selected. Broadly speaking, our approach here will be to "hardwire" selection of the root node and then introduce gadgets to ensure that whenever a node is selected, one of its two children is selected as well. In other words, our construction propagates selection from the root to some



Two nodes are adjacent iff their labels differ in every coordinate.

Figure 3: The connection pattern between the templates at a node and its children

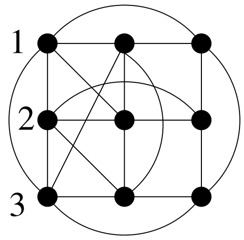


Figure 4: Enforcing selection at a root

leaf node. While one can imagine, at least for the case $k = 3$, that one can construct a "direct" 1-*out-of*-3 gadget which will ensure that one of three nodes is always selected, this "top-down" approach works for any value of k, and is also more modular and easier to present.

Root Selection: In each tree T_i, $1 \leq i \leq r$, we enforce selection of the root using the gadget shown in Figure 4. It is obtained by adding, for each $j \in \{1, 2, 3\}$, edges from the ground vertex colored j to the first vertex in row number ($j \bmod 3 + 1$) of the copy of H_{basic} at the root node r_i of T_i. This ensures that in any 4-coloring of H, there will be one row of (each) root which will be selected (and hence the root itself will be selected). Indeed, there must exist one row whose vertices are not colored using 4, say for concreteness the third row is such. But since we added an edge between ground vertex colored 2 to the first vertex in the third row, this vertex cannot be colored 2, and it follows that the third row of the root must be colored $(3, 1, 2)$, as desired.

Propagating the Selection: Next we show how selection of a node in the tree can be propagated to at least one of its children. This ensures that in each tree at least one leaf node must be selected. Consider again the schematic in Figure 3 and assign the following interpretation to the node labels:

- Colors in the first coordinate of each node correspond to the situation where $t_{i,j}$ is selected and it enforces selection at $v_{i,j}$.
- Colors in the second coordinate of each node correspond to the situation where $t_{i,j}$ is selected and it enforces selection at $t_{i,j+1}$.
- Colors in the third coordinate of each node correspond to the situation where $t_{i,j}$ is not selected and it does not enforce any selection at its children.

It is easy to verify that for any $l \in \{1, 2, 3\}$, if we assign colors 1, 2 and 3 to the nodes as specified by their lth coordinate, it forms a feasible coloring. Moreover, for any choice of a leaf node to be selected in T_i, coloring the nodes along the unique root-leaf path as selected (i.e coloring the three rows of the corresponding templates as $\{1, 2, 3\}$, $\{2, 3, 1\}$ and $\{3, 1, 2\}$), and the remaining nodes in T_i as not selected (i.e coloring the three rows of the corresponding templates as $\{1, 3, 2\}$, $\{2, 1, 3\}$ and $\{3, 2, 1\}$), yields a legal 3-coloring of T_i. The following is thus evident for our construction:

Lemma 2.1 *For each i, $1 \leq i \leq r$, and $\forall j$, $0 \leq j < k$, there is a 3-coloring of the vertices in the tree-structure T_i such that the leaf corresponding to $v_{i,j}$ is the only selected leaf in T_i.*

We can now establish the following key lemma:

Lemma 2.2 *In any 4-coloring of a tree T_i, whenever an internal node is selected, one of its two children must be selected.*

Proof. Consider again the schematic of Figure 3, with P being the parent whose selection we want to argue implies the selection of one of its children L and R. We consider two cases:

<u>Case 1</u>: Both vertices in one of the pairs $\{\langle 112\rangle, \langle 113\rangle\}$, $\{\langle 221\rangle, \langle 223\rangle\}$, and $\{\langle 331\rangle, \langle 332\rangle\}$ receive color 4 in the 4-coloring of H.

Suppose it is the pair $\{\langle 331\rangle, \langle 332\rangle\}$ that receives color 4. Since P is selected, the third row of P must be colored $(3, 1, 2)$ in this case. We now claim that one of L and R will in fact be selected with their third row being colored $(3, 1, 2)$. Indeed, none of the vertices $\langle 122\rangle$, $\langle 211\rangle$, $\langle 212\rangle$ and $\langle 121\rangle$ (the third row non-ground vertices of L and R) get the color 4 as they are all adjacent to one of $\langle 331\rangle$ or $\langle 332\rangle$. Thus if neither of L, R is selected, $\langle 122\rangle, \langle 212\rangle$ get colored 2 and $\langle 211\rangle, \langle 121\rangle$ get colored 1. Now it is easy to see that each of the vertices $\langle 123\rangle$, $\langle 231\rangle$ and $\langle 312\rangle$ have color 1 as well as color 2 neighbors (for instance $\langle 123\rangle$ is adjacent to $\langle 211\rangle$ and $\langle 212\rangle$), and this implies that all these three must be colored either 3 or 4. But this is impossible as these three vertices form a clique. Thus, one of L, R must be selected.

Exactly similar arguments will hold when both of the vertices $\langle 112\rangle, \langle 113\rangle$ receive color 4 or if both the vertices $\langle 221\rangle, \langle 223\rangle$ receive color 4. So it remains to consider the following case.

<u>Case 2</u>: At most one of the vertices in each of the pairs $\{\langle 112\rangle, \langle 113\rangle\}$, $\{\langle 221\rangle, \langle 223\rangle\}$, and $\{\langle 331\rangle, \langle 332\rangle\}$ receives color 4 in the 4-coloring of H.

In this case we first claim:

Claim 1 *At least one of the vertices $\langle 112\rangle, \langle 113\rangle$ gets colored 1, one of $\langle 221\rangle, \langle 223\rangle$ gets colored 2, and one of $\langle 331\rangle, \langle 332\rangle$ gets colored 3.*

To see this, note that P is selected, so we may assume without loss of generality, that the third row of P is colored $(3, 1, 2)$. Thus the above claim is trivially verified for the colors 1 and 2 (since $\langle 112\rangle$ is colored 1, and $\langle 221\rangle$ is colored 2). Now if neither $\langle 331\rangle$ nor $\langle 332\rangle$ is colored 3, then in fact they must both be colored 4 (since, for instance, $\langle 332\rangle$ cannot be colored either 1 or 2 since it is adjacent to $\langle 111\rangle$ and $\langle 221\rangle$ respectively). But this contradicts the hypothesis of this case, and therefore our claim holds. □

We are now ready to finish the proof for Case 2 also. Suppose P is selected, but neither of L, R is selected. We will call a row of a node *pure* if none of its vertices are colored 4. Clearly at least one of the rows of both L and R is pure. Since the entire gadget is totally symmetric, assume for definiteness that the third row of L is pure, so that it is colored $(3, 2, 1)$ (recall that L is *not* selected, so it cannot be colored $(3, 1, 2)$). Now if the third row of R is pure, then it will also be colored $(3, 2, 1)$, and we will get a contradiction *exactly* as we obtained in the analysis of Case 1. So one of the first or second rows of R is pure, say without loss of generality again, that the first row of R is pure so that it is colored $(1, 3, 2)$. The upshot of all this is that the vertices $\langle 122\rangle, \langle 211\rangle, \langle 323\rangle, \langle 322\rangle$ receive colors $2, 1, 3, 2$ respectively.

Now consider the vertex $\langle 231\rangle$. It is adjacent (among other vertices) to $\langle 122\rangle$ (which is colored 2), $\langle 323\rangle$ (which is colored 3), and to both $\langle 112\rangle, \langle 113\rangle$ one of which is colored 1 by Claim 1. It follows therefore that $\langle 231\rangle$ is colored 4. An exactly similar argument shows that $\langle 123\rangle$ must also be colored 4 — indeed $\langle 123\rangle$ is adjacent to $\langle 211\rangle$ (colored 1), to $\langle 322\rangle$ (colored 2), and to both $\langle 331\rangle, \langle 332\rangle$ one of which is colored 3 by Claim 1. But now $\langle 231\rangle$ and $\langle 123\rangle$ are both colored 4 and they are adjacent, a contradiction. This completes the analysis for Case 2 as well, and the proof is now complete. □

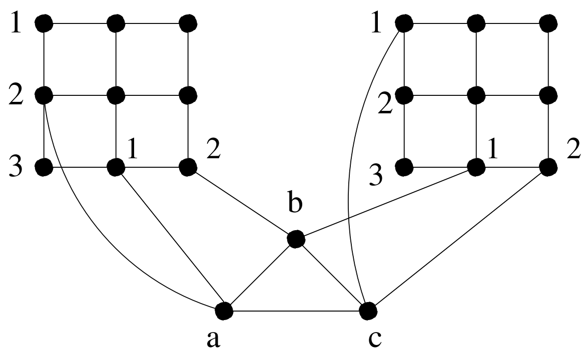


Figure 5: The Leaf-level Gadget: "Same Row Kind"

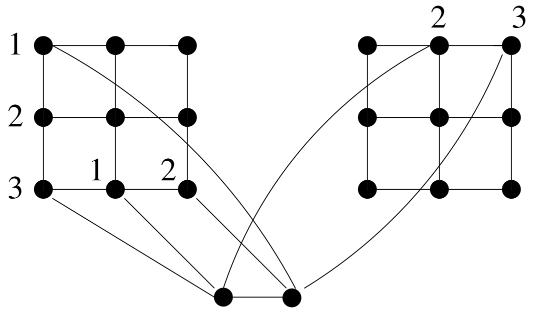


Figure 6: The Leaf-level Gadget: "Different Rows Kind"

2.3 The Structure Across the Trees

We now specify how the nodes across different T_i's are connected. For every pair of leaf nodes $v_{i,p} \in T_i$ and $v_{j,q} \in T_j$ such that $v_{i,p}$ and $v_{j,q}$ are adjacent in G, we insert a gadget (actually a combination of more than one gadget) that prevents both of these leaf nodes from being selected simultaneously in any legal 4-coloring of H. Observe that this would immediately imply that if H is 4-colorable, then there must be an independent set of size at least r in G. This follows from Lemma 2.2 which shows that in any 4-coloring of H, every tree has at least one selected leaf, and the fact that no two vertices of G corresponding to selected leaves can be adjacent in G.

The leaf-level gadget consists of two parts, as shown in Figures 5 and 6. Given two nodes, each a copy of the basic template H_{basic}, we use two kinds of gadgets. The one of the first kind, shown in Figure 5, prevents both nodes being selected because of the *same* row (for example because the third row of both nodes is colored $(3, 1, 2)$) – we use three such gadgets, one for each row. It is easy to check that the gadget in Figure 5 is 3-colorable as long as at least one of the two third rows are colored $(3, 2, 1)$, but is not even 4-colorable if both the third rows are colored $(3, 1, 2)$.

The second kind of leaf-level gadget, shown in Figure 6, ensures that the two nodes are not both selected because of *different* rows, and is even simpler than the first. Once again it is completely straightforward to check that the gadget works as desired; for instance for the gadget shown, there exists a valid 3-coloring as long either the third row of the left hand side node is $(3, 2, 1)$ or the first row of the right hand side node is $(1, 3, 2)$ (i.e at least one is not selected), but there is no valid 4-coloring if these rows are colored $(3, 1, 2)$ and $(1, 2, 3)$ (i.e if both are selected).

The preceding discussion has thus established the following:

Lemma 2.3 *If the graph H constructed as above is 4-colorable, then $\alpha(G) = r$.*

Lemma 2.4 *If $\alpha(G) = r$, then H is 3-colorable.*

Proof. Let $K = \{v_{i,p_i} : 1 \leq i \leq r\}$ be an independent set of size r in G, where $0 \leq p_i < k$ for each i. By Lemma 2.1, we can legally color all the vertices of the tree structures T_i using only three colors such that for each tree T_i, the leaf corresponding to v_{i,p_i} is the only one that is selected. It remains only to color the vertices used in the leaf-level gadgets. By the argument above we can color the vertices of any leaf-level gadget using just three colors provided at least one of the two leaf nodes it "connects" is not selected. But this condition is met for every leaf-level gadget in our case, since K is an independent set, and therefore there is no leaf-level gadget between any two of our selected leaf nodes. The entire graph H is thus 3-colorable. $\square$

Theorem 1 now follows from Lemmas 2.4 and 2.3 since the construction of H can be clearly accomplished in polynomial time.

We point out here that the graph H constructed in the reduction above is always 6-colorable. Indeed one can legally color all nodes in the tree-structures using three colors, and legally color all nodes in the leaf-level gadgets using three different colors, for a total of six colors.

3 Hardness for degree-bounded 3-colorable graphs

We now show that the result of Theorem 1 holds even if the input graph G has degree bounded by some constant Δ, thus establishing Theorem 2. Unlike Theorem 1, however, we do not see how to prove the result below without using the PCP Theorem. Specifically we use Proposition 3.1 below which follows from the PCP theorem and MAX SNP-hardness of MAX 3-SAT instances where each variable appears in at most a constant number of, say 5, clauses [21].

Proposition 3.1 *For every constant $t > 1$ there exist constants q, Δ such that given a graph G whose vertices can be partitioned into r cliques each containing exactly q vertices, and in which each vertex has degree at most Δ, it is NP-hard to distinguish between the cases $\alpha(G) = r$ and $\alpha(G) < r/t$.*

Proof of Theorem 2: We employ (essentially) the same reduction as in the proof of Theorem 1, except that we now start from a hard instance of INDEPENDENT SET as in Proposition 3.1 with a "gap" (in independent set size) of $t = 24$. The graph H thus constructed will satisfy $\chi(H) = 3$ if $\alpha(G) = r$ while $\chi(H) \geq 5$ if $\alpha(G) < r$. By the nature of the reduction presented in Section 2, and the fact that the maximum degree of G is at most Δ, it is easy to see that all vertices in H have very small degree except the three *ground vertices* which are shared across all the r tree-like structures in H (that correspond to the r cliques in G). We get around this by simply using a distinct set of three ground vertices in each of the r tree-structures to give a new degree-bounded graph H'. By a pigeonhole argument, since there are only 24 different colorings of a (labeled) 3-clique using 4 colors, there are at least $r/24$ of the tree-structures whose ground vertices in rows $1, 2, 3$ are colored using the same three colors c_1, c_2, c_3; we just label these colors as $1, 2, 3$ respectively. Now applying the argument used in the proof of Lemma 2.3 to the subgraph of G induced by the vertices in the $r/24$ cliques corresponding to these tree-structures, we conclude that if H' is 4-colorable, then $\alpha(G) \geq r/24$. Of course in the case when $\alpha(G) = r$, the same coloring used to establish Lemma 2.4

with all copies of the ground vertices being colored as $1, 2, 3$ properly implies that H' is 3-colorable. Combining this reduction with Proposition 3.1, therefore, gives us our claimed result. □

It turns out that the above argument also suffices to establish Theorem 3.

Proof of Theorem 3 *(Sketch)*: Use the same reduction to get a graph H as in the above proof, except now start from a hard instance of INDEPENDENT SET with a "gap" of $t = 48$. If n, m are respectively the number of vertices and edges in H, then we have $n = O(r)$, and since H is degree-bounded, $m = O(n)$. Thus $m = O(r) \leq c_0 r$ for some absolute constant c_0. Now define $\varepsilon_0 = 1/4c_0$. If a 4-coloring of H miscolors at most $\varepsilon_0 m$ edges, then since $\varepsilon_0 m \leq \frac{r}{4}$, there are at least $r/2$ tree-like structures such that they, and the leaf-level gadgets associated with them, are all legally colored using only 4 colors. Arguing as in the proof of Theorem 2, we can now conclude $\alpha(G) \geq r/48$. Thus when $\alpha(G) < r/48$, every 4-coloring of H legally colors at most $(1 - \varepsilon_0)$ fraction of the edges. □

4 Hardness of the form χ^δ for coloring

In this section we sketch the proof of a uniform inapproximability result of χ^h for coloring in terms of the chromatic number χ for a wide range of values of χ (for a constant $h > 1$). Using more complicated techniques, the same hardness result was established in [9] under the assumption NP $\not\subseteq$ ZPP. Our proof, based on transformations in [19], is simpler and also shows hardness under the weaker assumption NP $\neq$ P. By using more recent PCP constructions in the proof, we can prove the claimed hardness with any $h < 6/5$. Our focus here is not on the quantitative aspects, but to illustrate the fact that a uniform χ^h hardness can be shown to hold over a large range of values of χ.

Theorem 4 *There exist constants c, $\delta > 0$ and $h > 1$ such that for chromatic numbers in the range $c \leq \chi(G) \leq n^\delta$, it is NP-hard to $\chi(G)^h$ color a graph G on n vertices with chromatic number $\chi(G)$.*

Proof. The proof of this result follows by combining existing results in [3] and [19]. We assume familiarity with the terminology of PCPs like free bit complexity [3] and the construction of the FGLSS graph from a proof system [7].

We start with the PCP theorem [2, 1] which gives a PCP construction for NP that uses a logarithmic number of random bits, f free bits for some constant f, has perfect completeness and soundness 1/2, i.e we use NP $\subseteq \mathrm{FPCP}_{1,1/2}[\log n, f]$. This, together with [3][cf. Proposition 11.4] implies that, for every $\epsilon > 0$ and every admissible function $t : \mathcal{Z}^+ \to \mathcal{Z}^+$,

$$\mathrm{NP} \subseteq \mathrm{FPCP}_{1,2^{-t}}[O(\log n) + (2+\epsilon) \cdot t, (1+\epsilon) \cdot tf]. \tag{1}$$

Using the FGLSS transformation from proof systems to layered graphs with gaps in clique size [7], Equation (1) above (with the choice $\epsilon = 1$) implies that in a graph G layered into $R = n^{O(1)} 2^{3t}$ rows each being an independent set of size $q = 2^{2tf}$, it is NP-hard to distinguish whether $\omega(G) = R$ or if $\omega(G) \leq s \cdot R = 2^{-t} R$.

We now further map G to a graph H as in the reduction of Section 3 of [19] with parameter $k = 2^t = q^{1/2f}$. For sake of completeness we briefly sketch this mapping. The transformation from G to H is best described through an intermediate $(k \cdot R)$-partite graph G'. For each row i of G, there is a block B_i of k rows in G' such that the j^{th} row in B_i is simply the i^{th} row of G shifted by $(j-1)$ columns to the right in a wraparound manner. Thus each vertex of G has k copies in G'. For each edge (u, v) in G (here we also treat (u, u) to be an edge of G for every vertex u), we insert an edge between every copy of u and v in G'. The graph H is now obtained from G' by applying an appropriate injection $T : [q] \to [q']$ (with $q' = q^5$, see [19] for details) to the vertices of each row of G'. For every edge (u, v) in G', we have an edge $(T(u), T(v))$ in H, and the edge set is extended by including all the wraparound *rotations* of these edges. Hence the graph H has $N = q'kR$ vertices in all, organized as R *blocks* corresponding to the R rows in G, with k rows in each block and each row having exactly $q' = q^5$ vertices.

This transformation can be shown to satisfy the following two properties (see [19]). First, if $\omega(G) = R$, then $\theta(H) = \chi(\bar{H}) = q'$. Second, we have $\omega(H) \leq k\omega(G) + r$, and hence if $\omega(G) \leq sR$, $\omega(H) \leq ksR + R$, so that

$$\begin{aligned} \chi(\bar{H}) = \theta(H) &\geq \frac{N}{\omega(H)} \geq \frac{q'kR}{ksR + R} \\ &\geq \frac{q'q^{1/2f}}{s2^t + 1} = \frac{q'q^{1/2f}}{2} \\ &> q'^{1+1/10f}/4 > q'^h \end{aligned}$$

for any $h < 1 + 1/10f$ (if q' is greater than a sufficiently large constant). Now $q' = 2^{10tf} = 2^{\Theta(t)}$ and the total number of vertices in $\bar{H}$ is $N = n^{\Theta(1)} \cdot 2^{\Theta(t)} = n^{\Theta(1)}$ (assuming $t = O(\log n)$); and it is NP-hard to distinguish between the cases $\chi(\bar{H}) = q'$ and $\chi(\bar{H}) > q'^h$. By choosing t in the range of $O(1)$ to $O(\log n)$ we get the claimed range in q', the chromatic number of H, in terms of the number of vertices N. □

Acknowledgments

We would like to thank Madhu Sudan for several useful discussions.

References

[1] S. ARORA, C. LUND, R. MOTWANI, M. SUDAN, AND M. SZEGEDY. Proof verification and hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. Preliminary version in *Proc. of FOCS'92*.

[2] S. ARORA AND S. SAFRA. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. Preliminary version in *Proc. of FOCS'92*.

[3] M. BELLARE, O. GOLDREICH, AND M. SUDAN. Free bits, PCP's and non-approximability – towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998. Preliminary version in *Proc. of FOCS'95*.

[4] A. BLUM. *Algorithms for Approximate Graph Coloring*. Ph.D. thesis, Laboratory for Computer Science, MIT, Cambridge, MA, 1991.

[5] A. BLUM. New approximation algorithms for graph coloring. *Journal of the ACM*, 41:470-516, 1994.

[6] A. BLUM AND D. R. KARGER. An $\tilde{O}(n^{3/14})$-coloring algorithm for 3-colorable graphs. *Information Processing Letters*, 61(1):49-53, 1997.

[7] U. FEIGE, S. GOLDWASSER, L. LOVÁSZ, S. SAFRA, AND M. SZEGEDY. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268-292, 1996. Preliminary version in *Proc. of FOCS'91*.

[8] U. FEIGE AND J. KILIAN. Zero-knowledge and the chromatic number. In *Proceedings of the 11th Annual Conference on Computational Complexity*, 1996.

[9] M. FÜRER. Improved hardness results for approximating the chromatic number. *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science*, pp. 414-421, 1995.

[10] M. R. GAREY AND D. S. JOHNSON. The complexity of near-optimal graph coloring. *Journal of the ACM*, 23(1976), pp. 43-49.

[11] M. R. GAREY AND D. S. JOHNSON *Computers and Intractability — A guide to the theory of NP-completeness*. W. H. Freeman, 1979.

[12] V. GURUSWAMI, S. KHANNA, R. RAJARAMAN, B. SHEPHERD AND M. YANNAKAKIS. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. *Proc. of the 31st Annual ACM Symposium on Theory of Computing*, 1999, pp. 19-28.

[13] V. GURUSWAMI, J. HÅSTAD AND M. SUDAN. Hardness of Approximate Hypergraph Coloring. In *Proceedings of the 41st FOCS*, 2000, to appear.

[14] M. M. HALLDÓRSSON. A still better performance guarantee for approximate graph coloring. *Information Processing Letters*, 45:19-23, 1993.

[15] J. HÅSTAD. Clique is hard to approximate within $n^{1-\epsilon}$. *ECCC Technical Report TR97-038*. (Preliminary version in *Proc. of FOCS '96*).

[16] D. R. KARGER, R. MOTWANI AND M. SUDAN. Approximate graph coloring using semidefinite programming. *Journal of the ACM*, 45(2):246-265, 1998.

[17] R. M. KARP. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pp. 85-103, Plenum Press, 1972.

[18] S. KHANNA. Longest directed path is $n^{1-\epsilon}$-hard. Manuscript, 1999.

[19] S. KHANNA, N. LINIAL AND S. SAFRA. On the hardness of approximating the chromatic number. In *Proceedings of the 2nd Israel Symposium on Theory and Computing Systems, ISTCS*, pp. 250-260, IEEE Computer Society Press, 1993.

[20] C. LUND AND M. YANNAKAKIS. On the hardness of approximating minimization problems. *Journal of the ACM*, 41:960-981, 1994.

[21] C. H. PAPADIMITRIOU AND M. YANNAKAKIS. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43 (1991), pp. 425-440.

[22] E. PETRANK. The hardness of approximation: Gap Location. *Computational Complexity*, 4(2):133-157, 1994.

[23] A. WIGDERSON. Improving the performance guarantee for approximate graph coloring. *Journal of the ACM*, 30:729-735, 1983.