

Some Tools for Approximate 3-Coloring

(Extended Abstract)

Avrim Blum*

MIT Laboratory for Computer Science
Cambridge, MA 02139

Abstract

We present several tools for use in approximation algorithms to color 3-chromatic graphs. We then use these techniques in an algorithm that colors any 3-chromatic graph with $O(n^{3/8+o(1)})$ colors (or more precisely $O(n^{3/8} \log^{5/2} n)$ colors) which improves the previous best bound of $O(n^{0.4+o(1)})$ colors.

We also illustrate these techniques by considering a problem in which the 3-chromatic graph is created not by a worst-case adversary, but by an adversary each of whose decisions (whether or not to include an edge) is reversed with some small probability, or noise rate p . This type of adversary is equivalent to the “semi-random” source of Santha and Vazirani [SV86]. We present an algorithm that will actually 3-color such a graph with high probability even for quite low noise rates ($p \geq n^{-1/2+\epsilon}$ for constant $\epsilon > 0$).

1 Introduction

In this paper we consider the following two problems on the (vertex) coloring of 3-chromatic graphs. The first is the standard worst-case problem: given a 3-chromatic graph, color it with as few colors as possible in polynomial time. For this problem, Wigderson [Wig83] gave a simple algorithm for coloring with $O(n^{1/2})$ colors, and this bound was improved to $O(n^{1/2}/\log^{1/2} n)$ by Berger and Rompel [BR88] and to $O(n^{2/5+o(1)})$ by Blum [Blu89]. We present an algorithm that improves these bounds and colors any 3-chromatic graph with $O(n^{3/8+o(1)})$ (or more precisely $O(n^{3/8} \log^{5/2} n)$) colors. The second problem we consider is that of coloring a graph which is created not by a worst-case adversary, but by an adversary each of whose decisions (whether or not to include an edge) is reversed with some small probability, or noise rate p . We present an algorithm that even for quite low

noise rates will actually 3-color such a graph with high probability.

The algorithms are based on strategies for making progress from regions of high density in a 3-chromatic graph. For the worst-case problem, the previous $O(n^{2/5+o(1)})$ -color algorithm of [Blu89] performed most poorly when the input graph consisted of a collection of high-density regions or “clumps” with a lower density of edges between clumps. Essentially, that algorithm looked at subsets of the “grandchildren” (the neighbors of neighbors) of nodes in an attempt to find a large independent set. So, for a given average node degree, if the start node were in a dense region, the subsets considered might not be very large as many paths of length two could converge on the same grandchild. The tools presented here allow one to take advantage of such dense regions when they are found and thus to improve the coloring bound.

Although coloring 3-chromatic graphs with few colors in the worst case appears difficult—the bounds known are all quite far from the lower bound of 3—the problem of coloring *random* 3-chromatic graphs is much easier, and in fact it is known that one can actually 3-color such graphs with high probability [Kuc77][Tur88][DF89]. For concreteness, let us consider the model which we will denote $\mathcal{G}(n, p, 3)$ where the n vertices are separated into 3 equally-sized color classes, and then for each pair u, v of vertices of different colors, the edge (u, v) is placed into the graph with probability p . (See [DF89] for the relationship of this to other models for random 3-chromatic graphs.) Turner [Tur88] presents an algorithm that he shows will 3-color $G(n, p, 3)$ with high probability for all $p \geq n^{-1/3+\epsilon}$. Dyer and Frieze [DF89] go further and show an algorithm that works with “high enough” probability that when amortized over all 3-chromatic graphs, spends polynomial time on average per graph.

While the above results for coloring random 3-chromatic graphs imply that most 3-chromatic graphs are easy to 3-color, the random graphs are of a very special type. For instance, with high probability all vertices have nearly exactly the same degree and all

*Supported by an NSF Graduate Fellowship, NSF grant CCR-8914428 and the Siemens Corporation. Author’s email address: avrim@theory.lcs.mit.edu

have nearly the same number of edges to each of the other two color classes. So, many graphs created in only a “somewhat random” manner may not be covered. On the other hand, the worst-case model may be overly pessimistic in many situations. So, to analyze the coloring of graphs in an intermediate range, we consider here a “semi-random” graph model that lies in between the random and worst-case models.

In this model, which we will denote $\mathcal{G}_S(n, p, 3)$, the graph is generated by a version of the *semi-random* source (also called a “slightly-random” source) of Santha and Vazirani [SV86] (also discussed in [Vaz85] [VV85] [CG85]). The graph is created as follows. First, an adversary splits the n vertices into three color classes: R (red), B (blue), and G (green). Then for each pair of vertices u, v where u and v belong to different color classes (running through such pairs in an order of its choosing), the adversary decides whether or not to include edge (u, v) in the graph. Once the adversary has made a choice for a particular edge (u, v) , the choice is then reversed with probability p . So, in keeping with the semi-random source of [SV86], later choices of the adversary may depend on the outcomes of earlier decisions. An alternative way to view this model is that in an order of its choosing, for each pair of vertices u, v belonging to different color classes, the adversary picks a bias p_{uv} between p and $1 - p$ of a coin which is flipped to determine whether edge (u, v) is placed in the graph. The bias p_{uv} may depend on the outcome of previous coin tosses. We will call p the *noise rate* of the source. We also consider a slightly modified version of the semi-random model, which we will denote $\mathcal{G}_{SB}(n, p, 3)$, where the sizes of R , B , and G are required all to be $\Omega(n)$. We will call this last model the *balanced* semi-random graph model

The semi-random model separates the algorithms for coloring random 3-chromatic graphs into two categories. Some of the algorithms for the random model [DF89][Kuc77] are highly dependent on facts such as the edge probabilities all being equal and are easily defeated by a semi-random source. Others, such as Turner’s algorithm [Tur88], adapt well to the semi-random model. In particular, Turner’s bound of $p \geq n^{-1/3+\epsilon}$ for the random case holds in the balanced semi-random model as well. We present first an algorithm that achieves the same bound as Turner’s but with significantly simpler analysis (and that holds in the slightly more general $\mathcal{G}_S(n, p, 3)$ model), and then one that improves these bounds by 3-coloring semi-random graphs $\mathcal{G}_{SB}(n, p, 3)$ with high probability for $p \geq n^{-1/2+\epsilon}$.

The semi-random model is also useful for illustrating some of the strategies presented for the worst-case

problem. These strategies are used to make progress from dense regions, and in the semi-random graphs, the noise rate p determines a minimum density which applies uniformly to all parts of the graph. Each possible edge of the semi-random graph (that is, an edge between vertices of different colors in the adversary’s color scheme) appears in the graph with probability *at least* p no matter what the adversary’s strategy, although the adversary may make some regions of the graph even more dense if it so chooses. Because of the random noise of the source, the semi-random model is simpler than the worst-case model for describing the coloring strategy.

2 Notation and definitions

Given a 3-chromatic graph $\mathcal{G} = (V, E)$ and sets of vertices S and T ,

- Let $N(S) = \{u \in V \mid (u, v) \in E \text{ for some } v \in S\}$ be the *neighborhood* of S , and let $N(v) = N(\{v\})$ for vertices v .
- Let $d(v) = |N(v)|$ be the *degree* of vertex v , and let $d_T(v) = |N(v) \cap T|$ be the *degree into* T of v .
- Let $D(S) = \sum_{v \in S} d(v)$ be the *degree* of S , and let $D_T(S) = \sum_{v \in S} d_T(v)$ be the *degree into* T of S .

Notice that $D_T(S) = D_S(T)$.

- We will use R , B , and G to denote the sets of red, blue, and green vertices of \mathcal{G} respectively under some (unknown) 3-coloring of \mathcal{G} .

For functions f and g we will say $g(n) = \tilde{O}(f(n))$ to mean that $g(n)$ is bounded above by $f(n) \log^c n$ for some constant $c > 0$, and similarly use $\tilde{\Omega}(f(n))$ to mean bounded below by $f(n) \log^c n$.

3 Semi-random graphs

We now consider the models $\mathcal{G}_S(n, p, 3)$ and $\mathcal{G}_{SB}(n, p, 3)$ of a 3-chromatic graph generated by a semi-random source. Although for small constant noise rates p , say $p = 0.01$, it appears at first that the adversary has a good deal of power to defeat a coloring algorithm, it turns out that it does not. As previously mentioned, Turner’s algorithm [Tur88] will actually 3-color such a graph with high probability for any $p \geq n^{-1/3+\epsilon}$ for constant $\epsilon > 0$.

We present first a different algorithm that achieves the same bound, but works for the unbalanced case $\mathcal{G}_S(n, p, 3)$ as well, and has a much simpler analysis. We then show an extension of this strategy to an algorithm that will 3-color $\mathcal{G}_{SB}(n, p, 3)$ with high probability for p as low as $n^{-1/2+\epsilon}$ for constant $\epsilon > 0$.

Algorithm Two-Step

On input $\mathcal{G} = (V, E)$, first try to 2-color the graph. If that works, halt with success. Otherwise, do the following:

For each pair of vertices u, v (think of u as a candidate green node and v as a candidate blue node),

1. Let $S = N(u) \cap N(v)$.
2. Let $T = N(S)$.
If T is 2-colorable and $V - T$ is an independent set, then color T blue and green, color $V - T$ red and halt. Otherwise go to the top of the loop with a different pair u, v .

Theorem 1 (weak version) *Algorithm Two-Step will 3-color $\mathcal{G}_S(n, p, 3)$ with high probability (over the coin tosses of the semi-random source) for $p \geq n^{-1/3+\epsilon}$ and constant $\epsilon > 0$.*

Proof: For convenience, let “red” be the color with the most vertices in the adversary’s 3-coloring. If there are either no blue or no green vertices, then we will 2-color the graph at the start. Otherwise, let u be a green vertex of \mathcal{G} and v be a blue vertex (in the adversary’s 3-coloring). Then, the set $S = N(u) \cap N(v)$ contains only red vertices and so set $T = N(S)$ is blue and green. In fact with high probability, for $p \geq n^{-1/3+\epsilon}$, set T contains *all* the blue and green vertices for the following reason. If we view the semi-random source as choosing biases $p_{uv} \in [p, 1-p]$, then the sizes of sets S and T are minimized when each p_{uv} equals p . In that case, every vertex in R independently has a probability p^2 of belonging to S . So, using Chernoff bounds, $|S| \geq \frac{1}{2}|R|p^2 = \Omega(np^2)$ with high probability. Now, each vertex $z \in B \cup G$ such that $z \notin \{u, v\}$ has a probability $(1-p)^{|S|}$ of *not* belonging to T . The reason is that for $z \notin \{u, v\}$, for each $w \in R$, the events $A_{z,w}$ that edge (z, w) appears in the graph, occur with probability p and are independent of each other and of the choice of S . So, we have:

$$\Pr[z \notin T] \leq e^{-p|S|} = e^{-\Omega(np^3)} = e^{-\Omega(n^{3\epsilon})} = o(1/n).$$

That is, with high probability *all* vertices $z \in B \cup G$ will belong to T . Thus, with high probability, $T = B \cup G$ and $V - T = R$ and so for some pair u, v considered, algorithm Two-Step will succeed. ■

Algorithm Two-Step fails when p falls below $n^{-1/3}$ because then the vertices $u \in G$ and $v \in B$ may not share enough neighbors for $N(S)$ to equal $B \cup G$. However, for p below $n^{-1/3}$, set S might still contain many vertices, and applying additional iterations of the neighbor-taking process can be used to boost its size, especially when the sizes of the blue, green, and red vertex sets are roughly balanced. The following procedure extends algorithm Two-Step with additional neighbor-taking stages.

Algorithm k-Step

On input $\mathcal{G} = (V, E)$, and integer k :

For each pair of vertices u, v ,

1. Let $S_G^1 = \{u\}$, $S_B^1 = \{v\}$, and $S_R^1 = N(S_G^1) \cap N(S_B^1)$.
2. Let $S_G^2 = N(S_B^1) \cap N(S_R^1)$, $S_B^2 = N(S_G^1) \cap N(S_R^1)$, and $S_R^2 = N(S_G^2) \cap N(S_B^2)$.
3. Let $S_G^3 = N(S_B^2) \cap N(S_R^2)$, $S_B^3 = N(S_G^2) \cap N(S_R^2)$, and $S_R^3 = N(S_G^3) \cap N(S_B^3)$.
- ⋮
- k . Let $T = N(S_R^{k-1})$.

If T is 2-colorable and $V - T$ is an independent set, then color T blue and green, color $V - T$ red and halt. Otherwise go to the top of the loop with a different pair u, v .

Theorem 1 (strong version) *Algorithm k-Step will 3-color $\mathcal{G}_{SB}(n, p, 3)$ with high probability for $p \geq n^{-1/2+\epsilon}$, $\epsilon > 0$ constant, and $k > \log_3(1/\epsilon)$.*

Proof sketch: Again, if u is green and v is blue then for all i , $S_G^i \subseteq G$, $S_B^i \subseteq B$, and $S_R^i \subseteq R$, and $T \subseteq B \cup G$. Also, the sizes of the sets S_G^i, S_B^i, S_R^i and T are minimized by the semi-random source that chooses each p_{uv} to equal p . The general argument now is just repeated application of bounds for large deviations, being somewhat careful about independence. For this proof sketch, we will focus on the case where $k = 3$ and show that algorithm k-Step will 3-color $\mathcal{G}_S(n, p, 3)$ for $p = n^{-5/11+\epsilon}$ with high probability. We will then briefly sketch the proof idea for larger values of k .

We can imagine that the coin deciding the presence of an edge is not flipped until we actually examine that edge. So, we first examine all edges (u, w) and (v, w) for $w \in R$ and find that almost surely $|S_R^1| = \Theta(|R|p^2) = \Theta(np^2)$. Next, for each $z \in G$, we examine the edges (z, w) for $w \in S_R^1$ and the edge (z, v) . For $z \neq u$, these are all previously unexamined edges. So,

for $z \in G - \{u\}$ we have:

$$\begin{aligned} \text{Prob}[z \in S_G^2] &= p(1 - (1 - p)^{|S_R^1|}) \\ &= p^2 |S_R^1| (1 + o(1)). \\ &\quad (\text{using } p|S_R^1| = o(1)) \end{aligned}$$

This implies that with high probability, $|S_G^2| = \Theta(|G|np^4) = \Theta(n^2p^4)$ and similarly we have $|S_B^2| = \Theta(n^2p^4)$. Now, for each $z \in R - S_R^1$ we examine the edges (z, w) and (z, w') for $w \in S_G^2 - S_G^1$ and $w' \in S_B^2 - S_B^1$. Again, these are all previously unexamined edges, so the same argument as above shows that the probability z belongs to S_R^2 is proportional to $p^2 |S_G^2 - S_G^1| |S_B^2 - S_B^1|$. Thus with high probability, $|S_R^2 - S_R^1| = \Theta(n^5 p^{10})$. Finally, we have $T = N(S_R^2)$. Notice that set T contains $S_G^2 \cup S_B^2$ and that for each vertex $z \in (B \cup G) - (S_G^2 \cup S_B^2)$, we have not yet examined the edges (z, w) for $w \in S_R^2 - S_R^1$. So, for each such vertex z ,

$$\begin{aligned} \Pr[z \notin T] &\leq (1 - p)^{|S_R^2 - S_R^1|} \\ &= (1 - p)^{\Theta(n^5 p^{10})} \\ &\leq e^{-\Theta(n^5 p^{11})} \\ &= o(1/n), \quad \text{for } p = n^{-5/11+\epsilon}. \end{aligned}$$

So, with high probability, $T = B \cup G$.

More generally, so long as $p|S_G^i|$, $p|S_B^i|$, and $p|S_R^i|$ are all $o(1)$, then if $|S_R^i| = \Theta(n^x p^{2x})$ we will have with high probability that $|S_R^{i+1}| = \Theta(n^y p^{2y})$ for $y = 3x + 2$. Since we begin with $|S_R^1| = \Theta(n^{2\epsilon})$ and at each step the size of S_R^i more than triples, we can continue with the sets S_G^i , S_B^i , and S_R^i having size $o(1/p)$ for at most $\log_3(1/\epsilon)$ iterations. Once the condition on the set sizes no longer holds, say at step i , then after the next iteration set S_R^{i+1} will be large enough so that its neighborhood will equal $B \cup G$ with high probability. ■

4 Worst-case model: preliminaries

We now focus on the worst-case problem. Let $\mathcal{G} = (V, E)$ be a 3-chromatic graph on n vertices. For the strategy presented here, it will be useful to have a notion of “making progress” towards an $O(n^\alpha)$ -coloring of \mathcal{G} , where we will assume for the rest of this paper that $\alpha < 1/2$. Three important ways of making progress are defined as follows.

Progress type 1: Given a 3-chromatic graph on m vertices, find an independent or 2-colorable set of size $\Omega(m^{1-\alpha})$.

Progress type 2: Given a 3-chromatic graph on m vertices, find an independent set S of size r such that $|N(S)| = O(rm^\alpha)$.

Progress type 3: Given a 3-chromatic graph, find two vertices that can be guaranteed to be the same color under any legal 3-coloring of the graph.

Progress type 1 “makes progress” towards an n^α -coloring essentially because we can color the set found with two colors, throw away the colored vertices, pick two new colors to work with and continue. The idea for progress type 2 is that we can use it to find many different independent sets each of which is independent of all the others, thereby giving us progress of type 1. In addition, progress type 3 always helps us towards any approximate coloring. More formally,

Claim 1 *If some algorithm \mathcal{A} is guaranteed on any 3-chromatic graph of m vertices to make progress of types 1, 2 or 3 towards an $O(m^\alpha)$ -coloring, then there exists an algorithm \mathcal{B} that will color graph \mathcal{G} with $O(n^\alpha)$ colors.*

Progress type 1 and a weaker variant of 2 were used in [Wig83] and type 3 was used in [Blu89]. In fact, we can state Wigderson’s algorithm for coloring with $O(n^{1/2})$ colors by using progress types 1 and 2 as follows. If a vertex v has a neighborhood of size $\Omega(n^{1/2})$ then we make progress of type 1; otherwise, $|N(v)| = O(n^{1/2})$ so we make progress type 2.

Proof of Claim 1: First, if algorithm \mathcal{A} ever makes progress of type 3 on a subgraph of \mathcal{G} , then since the two vertices found must be the same color under any 3-coloring of the subgraph, they also must be the same color under any 3-coloring of \mathcal{G} . So, we can just merge the two vertices found and start again from the top, removing one vertex and using no colors. So, we may now assume that \mathcal{A} only makes progress of types 1 or 2. Next, if we can always find an independent or 2-colorable set of size $\frac{1}{c}m^{1-\alpha}$ (for some constant c) in a 3-chromatic graph of m vertices, then we will achieve an $O(n^\alpha)$ -coloring of \mathcal{G} by finding the set, coloring it with two colors, throwing those vertices out of the graph, and repeating. The number of colors used, $C(m)$, will satisfy $C(m) \leq 2 + C(m - \frac{1}{c}m^{1-\alpha})$ which implies $C(m) < 2c(m/2)^\alpha + C(m/2)$ and thus $C(n) < \frac{2c}{2^\alpha - 1}n^\alpha$. So, we just need some algorithm \mathcal{B}' that on any 3-chromatic graph of m vertices finds an independent set of size $(1/c)m^{1-\alpha}$. \mathcal{B}' works as follows.

On input (V, E) , where $m = |V|$,

1. Initialize set U to ϕ and V' to V .

2. While $|V'| \geq m/2$ do:
 - (a) Let (V', E') be the subgraph induced by the vertices in V' and run algorithm \mathcal{A} on (V', E') .
 - (b) If \mathcal{A} returns with progress of type 1, then since $|V'| \geq m/2$, we have an independent set of size $\Omega((m/2)^{1-\alpha}) = \Omega(m^{1-\alpha})$, so halt and output that set.
 - (c) If \mathcal{A} returns with progress of type 2, let S denote the set returned by \mathcal{A} , and update: $U \leftarrow U \cup S$ and $V' \leftarrow V' - S - N(S)$. Notice that we maintain an invariant that there are no neighbors of U in V' .
3. We now have $|V'| < m/2$. U is an independent set since there are no neighbors of U in any of the sets S which comprise it. In addition, each time we add r vertices to U , we remove at most $r + c'rm^\alpha$ vertices from V' for some constant c' by the definition of progress type 2. So, $|U| + c'|U|m^\alpha \geq m/2$ which implies $|U| = \Omega(m^{1-\alpha})$, so halt and output U . \blacksquare

Two simple corollaries of claim 1 are as follows.

Corollary 1 *Given an independent set S of $\Omega(n^{1-2\alpha})$ vertices, we can either make progress towards an $O(n^\alpha)$ -coloring or else guarantee that the vertices of S are not all the same color under any legal 3-coloring of \mathcal{G} .*

Proof: If the vertices of S were the same color, then $N(S)$ would be 2-colorable. So, if $N(S)$ is not 2-colorable, we output with guarantee. Otherwise, if $|N(S)| \geq n^{1-\alpha}$, we make progress of type 1, and if $|N(S)| < n^{1-\alpha}$, we output S as progress type 2. \blacksquare

Notice that this implies that for the case $\alpha \geq 1/3$, we may assume that each vertex in \mathcal{G} has as neighbors at least one representative of each of the other two colors under any legal 3-coloring of \mathcal{G} .

Corollary 2 (generalizes a statement in [Blu89]) *If there exist vertices v and w that share $\Omega(n^{1-2\alpha})$ neighbors, then we can make progress towards an $O(n^\alpha)$ -coloring of \mathcal{G} .*

Proof: Feed $N(v) \cap N(w)$ to the algorithm of Corollary 1. If that algorithm does not make progress, then we know that v and w must have been the same color under any legal 3-coloring of \mathcal{G} , so we make progress of type 3. \blacksquare

5 A useful lemma

We now present a strengthening of Corollary 1, described in Lemma 1 below, that provides us with a method for forcing the graph \mathcal{G} to behave in a certain “nice” way. For any vertex v of \mathcal{G} , for any subset S we choose of $N(v)$ of size at least $n^{1-2\alpha} \log^2 n$, the lemma allows us to force that S contain $\tilde{\Omega}|S|$ vertices of each of the two available colors (that is, the colors that v does not have), or else make progress in coloring \mathcal{G} .

Lemma 1 *Given a set S of $\Omega(n^{1-2\alpha} \log^2 n)$ vertices, we can either make progress or else guarantee that under no legal 3-coloring of \mathcal{G} does one color comprise more than $(1 - \frac{1}{4 \log n})$ of the vertices of S .*

Proof: For convenience, let *red* be the color such that $|R \cap S| = \max(|R \cap S|, |B \cap S|, |G \cap S|)$.

First, find an independent set $S' \subseteq S$ containing most of the vertices of S . We can do this in a greedy fashion. Let $S' = S$. While S' is not an independent set, repeat the following: pick vertices $a, b \in S'$ with the edge $(a, b) \in E$ and let $S' \leftarrow S' - \{a, b\}$. If we have removed more than $\frac{|S|}{2 \log n}$ vertices from S' , then we know that no single color could have colored more than $(1 - \frac{1}{4 \log n})$ of the vertices of S , so halt with guarantee. Otherwise, we have $|S'| > (1 - \frac{1}{2 \log n})|S| = \Omega(n^{1-2\alpha} \log^2 n)$.

If the neighborhood of S' is small: $|N(S')| = O(n^{1-\alpha} \log^2 n)$, then we have made progress of type 2, so halt with “progress made”. Otherwise, let $T = N(S')$, so we have $|T| \geq n^{1-\alpha} \log^2 n$. Repeat the following.

1. For each vertex v in T , arbitrarily mark one of the edges from v into S' . Let E' be the set of marked edges. For set $A \subseteq S'$, define $N'(A)$ to be the set of w in T such that $(v, w) \in E'$ for some $v \in A$, and define $N'(v)$ to be $N'(\{v\})$.
2. Partition S' into at most $(1 + \log n)$ sets of vertices S_1, \dots, S_m , such that in each set, if we consider only the edges in E' , the minimum degree is at least half of the maximum degree. That is, for all j ,
$$\min_{v \in S_j} |N'(v)| \geq \frac{1}{2} \max_{v \in S_j} |N'(v)|.$$
3. Pick i such that $N'(S_i)$ is maximized; so $|N'(S_i)| = \Omega(1/\log n)|T|$. Notice that if more than $(1 - \frac{1}{2 \log n})$ of the vertices of S_i are red, then at most $\frac{1}{\log n}$ of the vertices in $N'(S_i)$ can be red, since at worst the non-red vertices in S_i have twice as many neighbors in T (considering only the edges in E') as the red vertices do.

4. If more that $(1 - \frac{1}{2 \log n})$ of the vertices of S_i are red, then $N'(S_i)$ has an independent set of at least $\frac{1}{2}(1 - \frac{1}{\log n})$ its vertices, namely either $N'(S_i) \cap B$ or $N'(S_i) \cap G$, whichever is larger. This implies it has a vertex cover of at most $\frac{1}{2}(1 + \frac{1}{\log n})$ of its vertices. We can now apply a vertex-cover algorithm of Bar-Yehuda and Even[BYE83] and Monien and Speckenmeyer[MS85] which finds a vertex cover of size at most $(2 - \frac{\log \log n}{\log n})$ times the size of the minimum vertex cover, to yield an independent set of size at least:

$$\begin{aligned} & \left[1 - \left(2 - \frac{\log \log n}{2 \log n}\right) \frac{1}{2} \left(1 + \frac{1}{\log n}\right)\right] |N'(S_i)| \\ &= \Omega\left(\frac{1}{\log n}\right) |N'(S_i)| \quad \text{for } \log \log n > 4 \\ &= \Omega(|T|/\log^2 n). \end{aligned}$$

This independent set has size $\Omega(n^{1-\alpha})$ so we have made progress of type 1 and can halt with “progress made”.

5. If we did not make progress in step 4, then we *know* that fewer than $(1 - \frac{1}{2 \log n})$ of the vertices of S_i are red; in other words, at least $\frac{1}{2 \log n}$ of the vertices in S_i are blue or green.

So, let $S' \leftarrow S' - S_i$. If S' has not been reduced to less than $1/3$ its original size, then go back to 1. Otherwise break out of the loop and go to the next part of the algorithm.

If we reduced S' to less than a third of its original size, it must be that we did so by removing sets from S' each of which had at least $\frac{1}{2 \log n}$ of its vertices blue or green. That is, we must have removed more than:

$$\frac{2}{3} \frac{1}{2 \log n} \left[\left(1 - \frac{1}{\log n}\right) |S| \right] \geq \frac{1}{4 \log n} |S|$$

blue and green vertices from S . So, halt with guarantee. \blacksquare

6 Making progress from dense regions

We will now use Lemma 1 and the strategy applied to semi-random graphs to show how to take advantage of certain types of dense regions in a 3-chromatic graph in the worst-case model. We will consider the case of two sets of vertices S and T where S is 2-colored under some legal 3-coloring of \mathcal{G} and the number of edges between S and T is large compared with the sizes of the two sets.

Theorem 2 *Given sets of vertices S and T such that*

1. S is 2-colored under some legal 3-coloring of \mathcal{G} ,
2. the ratio $D_T(S)/|S| > n^{1-2\alpha} \log^2 n$, and
3. $[D_T(S)]^3 = \Omega\left(\left[|S| + \max_{v \in S} \{d_T(v)\}\right] \times \left[|S||T|^2 n^{1-2\alpha} \log n + |T||S|^2 n^{2-4\alpha}\right]\right)$,

then we can make progress towards an $O(n^\alpha)$ -coloring.

Before proving this theorem, let us first make sense of the condition on $[D_T(S)]^3$ by considering some examples. Suppose we wish to color with $n^{3/8}$ colors, S has size $n^{3/8}$, and each vertex v in S has degree $n^{3/8}$ into T . Then, $\frac{D_T(S)}{|S|} = n^{3/8}$ which is greater than $n^{1/4} \log^2 n$ (condition 2). The main condition reduces to:

$$n^{18/8} \geq cn^{3/8} \left[|T|^2 n^{5/8} \log n + |T| n^{10/8} \right].$$

Ignoring logarithmic factors, we make progress if $|T| = \tilde{O}(n^{5/8})$. On the other hand, if T has more than $n^{5/8}$ vertices and we can find a large independent set inside T , then we may also make progress of type 1. This will be the basic idea for the $O(n^{3/8} \log^{5/2} n)$ -coloring described later.

As another example, if we wished to color with $n^{0.35}$ colors, S had size $n^{0.35}$ and each vertex in S had degree $n^{0.35}$ into T , the main condition reduces to $n^{2.1} \geq cn^{0.35} \left[|T|^2 n^{0.65} \log n + |T| n^{1.3} \right]$. In this case, we only make progress if $|T| = \tilde{O}(n^{0.45})$ (here the $|T| n^{1.3}$ term is dominant).

Proof of Theorem 2: For convenience, fix some (unknown) 3-coloring of \mathcal{G} and let “blue” and “green” be the two colors that appear in S . Let $d_{avg} = D_T(S)/|S|$ be the average degree into T of vertices in S and let $n_T = |T|$ and $n_S = |S|$. We will want to keep track of those vertices of T that have a reasonably large degree into S , so let T' be the set of those vertices $v \in T$ such that $d_S(v) \geq \frac{1}{2} \frac{D_S(T)}{n_T}$. So, we have $D_S(T') \geq \frac{1}{2} D_S(T) = \frac{1}{2} D_T(S)$. We will also want to look at those vertices in S that have reasonable degree into T' , so let S' be the set of $v \in S$ such that $d_{T'}(v) \geq \frac{1}{2} \frac{D_{T'}(S)}{n_S}$. Thus, we have $D_{T'}(S') \geq \frac{1}{2} D_{T'}(S) = \frac{1}{2} D_S(T') \geq \frac{1}{4} D_T(S)$. So, each vertex $v \in S'$ has $d_{T'}(v) \geq \frac{1}{4} d_{avg}$.

Since we are given that $d_{avg}/2 > \frac{1}{2} n^{1-2\alpha} \log^2 n$, by Lemma 1 we can guarantee that each vertex $v \in S'$ have at least a fraction $\frac{1}{4 \log n}$ of its edges into T entering into non-red vertices. So, for some non-red color, which we will call “green” without loss of generality, at least $D_T(S')/(4 \log n)$ edges from S' enter into vertices of T given that color. Thus, some green vertex $g \in T$ has degree at least $D_T(S')/(4n_T \log n)$ into S' .

Figure 1.

Let $X = N(g) \cap S'$ and $n_X = |X|$. So, we have:

$$\begin{aligned} n_X &\geq \frac{1}{4} D_T(S') / (n_T \log n) \\ &\geq \frac{1}{16} D_T(S) / (n_T \log n). \end{aligned} \quad (1)$$

Let Y be the set of neighbors of X in T' . We want to show that Y must be large. By Corollary 2 we may assume that no two nodes of X share more than $n^{1-2\alpha}$ neighbors in T' . If we label the vertices of X as: x_1, x_2, \dots, x_{n_X} , then this implies that vertex x_i shares at most $(i-1)n^{1-2\alpha}$ neighbors with any of x_1, \dots, x_{i-1} . Consider those x_i such that this amount of sharing is at most $\frac{1}{2} d_{T'}(x_i)$. For these x_i , the number of neighbors in T' of x_i not shared with any of x_1, \dots, x_{i-1} is at least $\frac{1}{2}(d_{avg}/4)$ since $d_{T'}(x_i) \geq d_{avg}/4$. We will include all x_i for which $(i-1)n^{1-2\alpha} \leq \frac{1}{2}(d_{avg}/4)$, so we will be considering at least $\min\{n_X, 1 + \frac{1}{2}(d_{avg}/4)/n^{1-2\alpha}\}$ vertices x_i . This implies:

$$\begin{aligned} |Y| &\geq \\ &\min\left\{n_X \left[\frac{1}{2}(d_{avg}/4)\right], \left[\frac{1}{2}(d_{avg}/4)\right] \left[\frac{1}{2}(d_{avg}/4)/n^{1-2\alpha}\right]\right\} \\ &= \Omega\left(\min\left\{\frac{D_T(S)}{n_S} n_X, \frac{D_T^2(S)}{n_S^2} \frac{1}{n^{1-2\alpha}}\right\}\right). \end{aligned} \quad (2)$$

Since all vertices in X were neighbors in S of some green vertex and since S is colored green and blue, it must be that all the vertices in X are blue. So, all vertices of Y are red and green. In addition, since $Y \subseteq T'$, we know we can lower-bound the number of edges from Y to S by:

$$\begin{aligned} D_S(Y) &= \Omega(|Y| D_S(T) / n_T) \\ &\subseteq \Omega\left(\min\left\{\frac{D_T^2(S)}{n_S n_T \log n} \frac{D_T(S)}{n_T}, \frac{D_T^2(S)}{n_S^2 n^{1-2\alpha}} \frac{D_T(S)}{n_T}\right\}\right) \\ &\quad (\text{using equations 1 and 2}) \\ &\subseteq \Omega\left(\left[n_S + \max_{v \in S}\{d_T(v)\}\right] n^{1-2\alpha}\right) \\ &\quad (\text{using the given bound on } D_T(S)) \end{aligned} \quad (3)$$

It now must be the case that one of the following two possibilities occurs. First, if there is some green vertex

$g' \in S$ that is hit by more than $\frac{1}{2} D_S(Y) / n_S$ edges from Y , then according to equation 3 it is hit by $\Omega(n^{1-2\alpha})$ edges, so it must be that $N(g') \cap Y$ is a set of at least $n^{1-2\alpha}$ red vertices. ($N(g')$ is blue and red and Y is red and green so the intersection is red. See Figure 1). Thus, we can use Corollary 1 to make progress on it. Otherwise, the collection Z of vertices in S hit by more than $\frac{1}{2} D_S(Y) / n_S$ edges from Y , is all blue. The size of Z is at least the number of edges from Y into S , minus $n_S(\frac{1}{2} D_S(Y) / n_S)$, divided by the maximum number of edges that may hit a single vertex. That is,

$$\begin{aligned} |Z| &\geq [D_S(Y) - \frac{1}{2} D_S(Y)] / \max_{v \in S}\{d_T(v)\} \\ &\geq \frac{1}{2} D_S(Y) / \max_{v \in S}\{d_T(v)\}. \end{aligned}$$

Using equation 3 we get:

$$|Z| = \Omega(n^{1-2\alpha}).$$

So, we can use Corollary 1 to make progress on Z . The final algorithm for making progress given our sets S and T is as follows:

1. Run the algorithm of Lemma 1 on $N(v) \cap T$ for all $v \in S$. If any runs make progress, then halt. Otherwise, we know there are many edges from S into red, blue, and green vertices under any legal 3-coloring of \mathcal{G} .
2. For every pair of vertices $u, v \in S$, if $|N(v) \cap N(u)| = \Omega(n^{1-2\alpha})$, then use the algorithm of Corollary 2 to make progress.
3. For each vertex $v \in T$,
 - (a) let $Y = N(N(v) \cap S) \cap T$. Let $Z = \{w \in S : d_Y(w) \geq n^{1-2\alpha}\}$. (Note that though we could, we do not actually need to use the sets S' and T' ; they were just convenient for the analysis.)
 - (b) Run the algorithm of Corollary 1 on Z .
 - (c) For each $w \in Z$, run the algorithm of Corollary 1 on $Y \cap N(w)$.

The above proof guarantees that this algorithm makes progress. ■

7 An $\tilde{O}(n^{3/8})$ -coloring

We can combine Theorem 2 with the $\tilde{O}(n^{0.4})$ -coloring algorithm of [Blu89] to yield an improved approximation algorithm that colors any 3-chromatic graph with $O(n^{3/8} \log^{5/2} n)$ colors. First, we need to define some additional notation.

- Let $\delta = \frac{1}{5 \log n}$.
- Let $I_j = \{v \mid d(v) \in [(1 + \delta)^j, (1 + \delta)^{j+1}]\}$ for $j = 0, 1, 2, \dots$. That is, in each set I_j , the ratio of degrees of any two vertices is at most $(1 + \delta)$. The number of sets I_j is at most $\log_{1+\delta} n = O(\log^2 n)$.
- Let $N_i(S) = \{v \in N(S) \mid d_S(v) \in [(1 + \delta)^i, (1 + \delta)^{i+1}]\}$ for $i = 0, 1, 2, \dots$. In other words, $N_i(S)$ for $0 \leq i \leq \log_{1+\delta} n$ is the subset of vertices in $N(S)$ that are endpoints of at least $(1 + \delta)^i$ and at most $(1 + \delta)^{i+1}$ edges from S .

Theorem 1 of [Blu89] states that under some legal 3-coloring of \mathcal{G} , there exists a red vertex v such that for some set I_j , the set $S = N(v) \cap I_j$ has the property that

$$\begin{aligned} |S| &\geq \delta^2 d / \log_{1+\delta} n & (4) \\ \text{and } D_R(S) &\geq \frac{1}{2}(1 - 3\delta)D(S), & (5) \end{aligned}$$

where d is the minimum degree in \mathcal{G} . That is, one of a relatively few subsets of the neighbors of v is both large and has nearly half the edges that leave it entering into red vertices. The proof of theorem 2 of [Blu89] shows that for some i , the set $T = N_i(S)$ has the property that

$$\begin{aligned} D_{T \cap R}(S) &\geq \delta D_R(S) / \log_{1+\delta} n & (6) \\ \text{and } |T \cap R| &\geq \frac{1}{2}(1 - 5\delta)|T|. & (7) \end{aligned}$$

We now put these facts together with Theorem 2 of this paper to yield an $O(n^{3/8} \log^{5/2})$ -coloring.

First, if $|T| \geq n^{5/8} / \log^{3/2} n$, then as in [Blu89], we can use the vertex-cover approximation algorithm of Bar-Yehuda et al. [BYE83][MS85] to find an independent set of size $\Omega(|T| / \log n) \subseteq \Omega(n^{5/8} / \log^{5/2} n)$, since T has an independent set (the red vertices) of $\frac{1}{2}(1 - \frac{1}{\log n})$ of its vertices. So, we have made progress of type 1. Otherwise, $|T| \leq n^{5/8} / \log^{3/2} n$, and we show that this implies that S and T satisfy the conditions of theorem 2 of this paper for making progress towards the desired coloring bound.

We may assume the minimum degree d is at least $n^{3/8} \log^{5/2} n$ or else we immediately make progress type 2. So, we can lower bound the size of the set S using equation 4 by

$$\begin{aligned} |S| &\geq \delta^2 d / \log_{1+\delta} n \\ &= \Omega(d / \log^4 n) \\ &\subseteq \Omega(n^{3/8} / \log^{3/2} n). \end{aligned} \quad (8)$$

In addition, equation 5 shows that $D_R(S) = \Omega(d|S|)$, so equation 6 implies:

$$D_T(S) \geq D_{T \cap R}(S)$$

$$\begin{aligned} &= \Omega(\delta d |S| / \log_{1+\delta} n) \\ &\subseteq \Omega(|S| n^{3/8} / \log^{1/2} n) \end{aligned} \quad (9)$$

or equivalently,

$$|S| = O(D_T(S)(\log^{1/2} n) / n^{3/8}). \quad (10)$$

Finally, since $S \subseteq I_j$, all vertices of S have nearly the same degree (though not necessarily the same degree into T), so

$$\begin{aligned} \max_{v \in S} \{d_T(v)\} &= O\left(\frac{D_T(S)}{|S|}\right) \\ &\subseteq O\left(\frac{D_T(S)}{|S|} \log^3 n\right) \\ &\quad (\text{using equations 5 and 6}) \\ &\subseteq O\left(D_T(S)(\log^{9/2} n) / n^{3/8}\right) \\ &\quad (\text{using equation 8}). \end{aligned} \quad (11)$$

Now, let us restate Theorem 2, plugging in $n^\alpha = n^{3/8} \log^{5/2} n$.

Corollary 3 *Given sets S and T such that*

1. S is 2-colored under some legal 3-coloring of \mathcal{G} ,
2. the ratio $D_T(S) / |S| > n^{1/4} / \log^3 n$, and
3. $[D_T(S)]^3 = \Omega\left(\left[|S| + \max_{v \in S} \{d_T(v)\}\right] \times \left[|S||T|^2 n^{1/4} / \log^4 n + |T||S|^2 n^{1/2} / \log^{10} n\right]\right)$,

then we can make progress towards an $O(n^{3/5} \log^{5/2} n)$ -coloring.

Claim 2 *Given sets S and T satisfying equations 8 through 11 with $|T| \leq n^{5/8} / \log^{3/2} n$, the algorithm of Theorem 2 makes progress towards an $O(n^{3/5} \log^{5/2} n)$ -coloring.*

Proof: First, by equation 9, $D_T(S) / |S|$ is much larger than $n^{1/4} / \log^3 n$. We now consider the restriction on $[D_T(S)]^3$. We can upper-bound the first term $[|S| + \max_{v \in S} \{d_T(v)\}]$ using equations 10 and 11 by $O(D_T(S)(\log^{9/2} n) / n^{3/8})$. We can upper-bound the second term using equation 10 and the given bound on $|T|$ by:

$$\begin{aligned} |S||T|^2 n^{1/4} / \log^4 n + |T||S|^2 n^{1/2} / \log^{10} n &= \\ O(D_T(S) n^{9/8} / (\log^{13/2} n) + [D_T(S)]^2 n^{3/8} / \log^{21/2} n). \end{aligned}$$

So, multiplying the two terms together, we just need show:

$$\begin{aligned} [D_T(S)]^3 &= \\ \Omega\left([D_T(S)]^2 n^{3/4} / \log^2 n + [D_T(S)]^3 / \log^6 n\right). \end{aligned}$$

Clearly $[D_T(S)]^3 > [D_T(S)]^3 / \log^6 n$, so we just need that $D_T(S) = \Omega(n^{3/4} / \log^2 n)$, but this is implied by equations 8 and 9. \blacksquare

8 Conclusions

This paper provides tools for use in approximate coloring algorithms for forcing a 3-chromatic graph to act in certain well-behaved ways. Theorem 2, which provides a bound on the density of certain kinds of “clumps” in a graph, can be used together with previous results to achieve an $\tilde{O}(n^{3/8})$ -coloring of any 3-chromatic graph. We also show how to 3-color *semi-random* 3-colorable graphs with high probability when the noise rate p is at least $n^{-1/2+\epsilon}$.

The techniques described in this paper might be useful in improved approximation algorithms, but it appears that something drastically different would be needed to color 3-chromatic graphs with fewer than $\tilde{O}(n^{1/3})$ colors in the worst case. At that point, Lemma 1 no longer forces each vertex to have similar numbers of neighbors of each of the two available colors (up to logarithmic factors) and so Theorem 2 no longer provides a useful density bound.

I would like to thank Umesh Vazirani and Joel Spencer for many helpful discussions.

References

- [Blu89] A. Blum. An $\tilde{O}(n^{0.4})$ -approximation algorithm for 3-coloring (and improved approximation algorithms for k -coloring). In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, pages 535–542, Seattle, May 1989.
- [BR88] B. Berger and J. Rompel. A better performance guarantee for approximate graph coloring. *Algorithmica*, 1988.
- [BYE83] R. Bar-Yehuda and S. Even. A $2 - \frac{\log \log n}{2 \log n}$ performance ratio for the weighted vertex cover problem. Technical Report Technical Report #260, Technion Haifa, January 1983.
- [CG85] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. In *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science*, pages 429–442, Portland, 1985.
- [DF89] M. E. Dyer and A. M. Frieze. The solution of some random NP-Hard problems in polynomial expected time. *Journal of Algorithms*, 10:451–489, 1989.
- [Kuc77] L. Kucera. Expected behavior of graph colouring algorithms. In *Lecture Notes in Computer Science No. 56*, pages 447–451. Springer-Verlag, 1977.
- [MS85] B. Monien and E. Speckenmeyer. Ramsey numbers and an approximation algorithm for the vertex cover problem. *Acta Informatica*, 22:115–123, 1985.
- [SV86] M. Santha and U. V. Vazirani. Generating quasi-random sequences from semi-random sources. *JCSS*, 33:75–87, 1986.
- [Tur88] J. S. Turner. Almost all k -colorable graphs are easy to color. *Journal of Algorithms*, 9:63–82, 1988.
- [Vaz85] U. V. Vazirani. Towards a strong communication complexity theory, or generating quasi-random sequences from two communicating slightly-random sources. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 366–378, Providence, 1985.
- [VV85] U. Vazirani and V. Vazirani. Random polynomial time is equal to slightly-random polynomial time. In *Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science*, pages 417–428, Portland, October 1985.
- [Wig83] A. Wigderson. Improving the performance guarantee for approximate graph coloring. *JACM*, 30(4):729–735, 1983.