Coloring Random and Semi-Random k-Colorable Graphs

Avrim Blum*
School of Computer Science
Carnegie Mellon University

Joel Spencer
Courant Institute
New York University

*Supported by an NSF Postdoctoral Fellowship. Part of this research was conducted while the author was at MIT and supported by an NSF Graduate Fellowship.

Please send all correspondence to:

Avrim Blum
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Tel: (412) 268-6452
Email: `avrim@theory.cs.cmu.edu`

Proposed running head: **Coloring Random and Semi-Random Graphs**

Abstract

The problem of coloring a graph with the minimum number of colors is well known to be NP-hard, even restricted to k-colorable graphs for constant $k \geq 3$. On the other hand, it is known that *random* k-colorable graphs are easy to k-color. The algorithms for coloring random k-colorable graphs require fairly high edge densities, however. In this paper we present algorithms that color randomly generated k-colorable graphs for much lower edge densities than previous approaches. In addition, to study a wider variety of graph distributions, we also present a model of graphs generated by the *semi-random* source of Santha and Vazirani that provides a smooth transition between the worst-case and random models. In this model, the graph is generated by a "noisy adversary" — an adversary whose decisions (whether or not to insert a particular edge) have some small (random) probability of being reversed. We show that even for quite low noise rates, semi-random k-colorable graphs can be optimally colored with high probability.

1 Introduction

A k-coloring of a graph is an assignment of one of k distinct colors to each vertex in the graph so that no two adjacent vertices are given the same color. The *chromatic number* of a graph is the smallest k such that the graph can be k-colored. Graph coloring problems have a long history in mathematics and computer science. They model a collection of scheduling problems such as examination scheduling and register allocation [2, 7, 8]. Unfortunately from the algorithmic point of view, coloring k-colorable graphs with k colors for $k \geq 3$ is NP-hard (for $k = 2$, 2-coloring is easy). On the other hand, knowing that the coloring problem is NP-hard does not make it disappear, and it also does not necessarily mean nothing useful can be done.

One general direction towards the goal of providing a useful algorithm is to relax the restriction that the algorithm work on all k-colorable graphs, and instead study *random* families of graphs. Turner [16], Kucera [11], and Dyer and Frieze [10] give polynomial-time algorithms that color random k-colorable graphs with k colors with high probability, for any constant k. So, *most* k-colorable graphs are easy to k-color. In fact, Dyer and Frieze go further and provide an algorithm that when amortized over all n-vertex k-colorable graphs, spends on average polynomial time per graph. Petford and Welsh [12] present experimental work using heuristics for coloring random 3-colorable graphs and claim success for a wide range of edge probabilities.

In this paper we extend the class of randomly-generated k-colorable graphs that can be provably optimally colored in polynomial time. In addition, we provide algorithms that color graphs created in a "somewhat random" manner based on the *semi-random source* model of Santha and Vazirani.

We first consider a standard model of a random k-colorable graph in which vertices are first randomly assigned to one of k color classes and then each edge between two vertices of different color is placed into the graph with probability p. For this model, we are able to find colorings for a wider range of edge probabilities ($p \geq n^{-1+\epsilon}$ for any constant $\epsilon > 0$) than was previously known.

While the known results on random graphs imply that *most* k-colorable graphs are easy to k-color, random k-colorable graphs tend to be of a very special type. For example, with high probability all vertices of a random k-colorable graph have nearly the same degree and all pairs of vertices of the same color class have nearly the same number of common neighbors. So, graphs created in only a "somewhat random" manner may not be colored well by algorithms for the average case. On the other hand, worst-case assumptions may be overly pessimistic in many situations. To explore an intermediate range, and a wider variety of graph distributions, we present a model of graphs created by the *semi-random* source of Santha and Vazirani [13]. In this model, the graph is generated by a "noisy adversary" — an adversary whose decisions (whether or not to insert a particular edge) have some small probability of being reversed. This model provides a smooth transition between the worst-case and random models. We show that even for quite low noise rates, these semi-random k-colorable graphs can be colored with high probability using just k colors. Because 3-chromatic graphs are the simplest and in a sense the most fundamental graphs for which optimal coloring is NP-hard, part of this paper will focus on the special case of coloring graphs of chromatic number 3.

A second standard approximation issue that we do *not* consider here is to examine worst-case graphs, but allow the number of colors used to be non-optimal. For work in this direction, see [20, 3, 4, 5]. Some of the work in this paper has previously appeared in extended abstract form [4] and a longer version of this paper appears in [5].

2 Notation and definitions

In this section we review some standard combinatorial and graph-theoretic definitions and notation that will be used throughout this paper.

Given a graph G, let $V(G)$ denote the vertices of G and $E(G)$ denote the edges of G. We will use $N(v)$ to denote the *neighborhood* of a vertex v. That is, for $G = (V, E)$, $N(v) = \{w \in V \mid (v, w) \in E\}$. We also define the neighborhood $N(S)$ of set S to be the union of the neighborhoods of the vertices in S. An *independent set* in a graph is a set of vertices no two of which are adjacent to each other. All graphs we consider are labeled graphs.

As mentioned in the introduction, the *chromatic number* of a graph is the least number of colors needed to color the graph so that no two adjacent vertices are given the same color. We say that an algorithm *t-colors* a graph if it colors the graph with at most t colors, and it *optimally* colors a graph if it colors with the fewest number of colors possible.

For the special case where G is a 3-colorable graph, we use red, blue, and green to denote the colors of vertices in G under some legal (but unknown) 3-coloring. We also use these terms to denote the sets of vertices belonging to each color class under that legal coloring.

For functions f and g we will write "$g(n) \gg f(n)$" to mean that $f(n) = o(g(n))$. Finally, we use the following general standard notation:

- $(m)_i = m(m-1)(m-2)\cdots(m-i+1)$.
- K_t is the clique on t vertices.
- For S a subset of vertices of graph G, the graph $H = G|_S$ is the subgraph of G induced by set S. That is, $V(H) = S$ and $E(H) = \{(i,j) \in E(G) \mid i,j \in S\}$.

When discussing a family of graphs, we will use the term "with high probability" to mean that the probability tends to 1 as the number of vertices tends to infinity.

3 Coloring random k-colorable graphs

The standard model for a random n-vertex graph is the model $\mathcal{G}(n,p)$ in which each possible edge is placed into the graph independently with probability p. This model has the property that the distribution $\mathcal{G}(n, 1/2)$ is the same as that obtained by selecting a labeled n-vertex graph uniformly at random from the set of all n-vertex graphs.

There are several natural models, however, for what one means by a *random k-colorable* graph. Dyer and Frieze examine several and prove relationships among them [10]. We focus here on one model that happens to be simplest to analyze, which we shall denote $\mathcal{G}(n,p,k)$. A graph is selected in $\mathcal{G}(n,p,k)$ according to the following procedure. First each labeled vertex is independently assigned to one of k color classes with equal probability $1/k$. Then, independently for each pair u, v of vertices in different color classes, the edge (u,v) is placed into the graph with probability p. We use the notation:

$$G \leftarrow \mathcal{G}(n,p,k)$$

to mean that G is selected according to the distribution defined by this model. Note: what is given to the algorithm is just a graph; the assignment of nodes to color classes is kept secret.

The $\mathcal{G}(n,p,k)$ model is a natural one for a random n-vertex k-colorable graph, though even for $p = 1/2$ it is not equivalent to selecting a graph uniformly at random from the set of all n-vertex k-colorable graphs. In particular, graphs that can be k-colored in multiple different ways are over-represented in $\mathcal{G}(n,1/2,k)$ since different assignments to the color classes may still lead to the same graph. (See Dyer and Frieze [10] for more on the relationship between the models.)

In this section, we consider the problem of k-coloring graphs in $\mathcal{G}(n,p,k)$ for as low an average edge density as possible. We present an algorithm to k-color such graphs with high probability for any constant k, and for $p \geq n^{\epsilon-1}$ for any fixed $\epsilon > 0$; that is, the procedure will work when the average degree is as low as n^{ϵ}. Before describing that algorithm, however, we point out first a quite easy method to k-color $G \leftarrow \mathcal{G}(n,p,k)$ for $p \geq n^{-1/2+\epsilon}$ $(\epsilon > 0)$.

This idea of the easier procedure is simply this: two vertices from the *same* color class in G will tend to share more neighbors in common than two vertices of different color. If two vertices are of the same color, then each of the $n-2$ other vertices has probability $(1-1/k)$ being assigned to a different color and thus has a probability $(1-1/k)p^2$ of being a neighbor to both. So, the two vertices share an expected $(n-2)(1-1/k)p^2$ neighbors in common. However, if two vertices are from different color classes, then each additional vertex has probability only $(1-2/k)$ of being assigned to a different color from both, so the expected number of shared neighbors is only $(n-2)(1-2/k)p^2$. For $p \geq n^{-1/2+\epsilon}$, these values are within $(1+o(1))$ of $n^{2\epsilon}(1-1/k)$ and $n^{2\epsilon}(1-2/k)$ respectively. Since for any given pair of colored vertices x, y, the indicator random variables X_v for the event that v is a neighbor to both x and y are mutually independent over all v, we may apply Chernoff bounds. In particular, if $X = \sum X_v$, and $\mu = \mathbf{E}[X]$ is the expected number of neighbors in common between x and y, Chernoff bounds state that for any $\delta > 0$,

$$\mathbf{Pr}[X < (1-\delta)\mu \ \mathbf{or}\ X > (1+\delta)\mu] \ < \ 2e^{-\delta^2\mu/3}$$

For $\mu = \Theta(n^{2\epsilon})$, this probability is so small that even when summed over all pairs of starting vertices x, y, the probability *any* pair shares a number of neighbors that differs by more than $\delta\mu$ from the expectation is $o(1)$.

One thus finds that with high probability, *all* pairs of vertices selected in the same color class share $n^{2\epsilon}[1-1/k](1+o(1))$ neighbors and *all* pairs of vertices of different color share only $n^{2\epsilon}[1-2/k](1+o(1))$ neighbors in common. Thus, one can easily algorithmically separate the color classes.

3.1 Idea for an improved algorithm

Another way to view the above observation is that vertices of the same color will have more paths of length 2 between them than vertices of different colors. This idea can be extended to paths of a longer constant length l for improved bounds. If l is *even*, it turns out that the expected number of simple paths of length l between two vertices of the same color is higher than the expected number for vertices of different color. If l is *odd*, the reverse holds. The difficulty in analyzing the case $l > 2$, however, is that the events corresponding to the paths of length l between two vertices are no longer independent. Different paths of length 2 between two vertices x and y share no edges in common, but two paths of length 3 might share an edge: for example, consider the two paths (x, w, w', y) and (x, w', w, y). So, to prove that the number of paths will be with high probability close to the number expected, one needs a more sophisticated probabilistic analysis. Such analysis for a general class of problems of this sort was described in [14] in the context of the random graph model $\mathcal{G}(n,p)$. It turns out that the analysis can be used for the $\mathcal{G}(n,p,k)$ model as well.

3.2 Calculating expectations

Let $l \geq 2$ be some integer constant and let us fix two vertices x and y. By a "path" we will always mean a simple path; that is, one that never touches any vertex more than once. In this section, we calculate the *expected* number of paths of length l between x and y in $G \leftarrow \mathcal{G}(n,p,k)$ and show this expectation differs by a constant factor depending on whether or not x and y are in the same color class. (l is the number of edges in the path.)

Let $E_l(p)$ be the expected number of paths of length l between x and y in $G \leftarrow \mathcal{G}(n,p,k)$, and let $E_l^{\text{same}}(p)$ and $E_l^{\text{diff}}(p)$ be the expected number of such paths *given* that x and y are placed in the same or in different color classes respectively. Also, let $\lambda_l(p) = [E_l^{\text{same}}(p) - E_l^{\text{diff}}(p)]/E_l(p)$. When p is clear from context, we will just write E_l, E_l^{same}, E_l^{diff}, and λ_l for the above quantities. We show now that for constant k and l, the value $|\lambda_l|$ is at least a constant > 0.

We can calculate the expected number of paths between x and y by fixing some arbitrary sequence of distinct vertices (also distinct from x and y) $v_1, \ldots, v_{l-1}$ and calculating the probability of the event B_l that each pair $\langle x, v_1\rangle, \langle v_1, v_2\rangle, \ldots, \langle v_{l-2}, v_{l-1}\rangle, \langle v_{l-1}, y\rangle$ consists of vertices assigned to different color classes. Given that the event B_l occurs, the probability the path $\langle x, v_1, \ldots, v_{l-1}, y\rangle$ appears in G is simply p^l. Given that B_l does not occur, the probability is 0. Since there are $(n-2)(n-3)\cdots(n-l) = (n-2)_{l-1}$ possible sequences $v_1, \ldots, v_{l-1}$, the expectation E_l is simply $p^l(n-2)_{l-1}\mathbf{Pr}[B_l]$.

For any event X, let $\mathbf{Pr}^{\text{same}}[X]$ and $\mathbf{Pr}^{\text{diff}}[X]$ be the probability that X occurs *given* that x and y are in the same color class, or *given* that x and y are in different color classes, respectively. Thus, $E_l^{\text{same}} = p^l(n-2)_{l-1}\mathbf{Pr}^{\text{same}}[B_l]$ and $E_l^{\text{diff}} = p^l(n-2)_{l-1}\mathbf{Pr}^{\text{diff}}[B_l]$.

Since the $p^l(n-2)_{l-1}$ terms factor out of the expression for λ_l, we have:

$$\lambda_l = \lambda_l(p) \quad = \quad \frac{\mathbf{Pr}^{\text{same}}[B_l] - \mathbf{Pr}^{\text{diff}}[B_l]}{\mathbf{Pr}[B_l]} \tag{1}$$

So, to compute λ_l we need only examine the fixed sequence $v_1, \ldots, v_{l-1}$ and the event that all are chosen colors such that the path $\langle x, v_1, \ldots, v_{l-1}, y\rangle$ is a "potential path" in the graph. Note that the value $\mathbf{Pr}[B_l]$ is quite easy to calculate: each vertex in the path has a $(1-\frac{1}{k})$ probability of being given a different color than the preceding vertex. So, $\mathbf{Pr}[B_l] = (1-\frac{1}{k})^l$.

All that remains is to prove the following theorem.

Theorem 1 $\mathbf{Pr}^{\text{same}}[B_l] - \mathbf{Pr}^{\text{diff}}[B_l] = (-1)^l(\frac{1}{k})^{l-1}$.

Proof: Define the following events A_t and B_t for $t \leq l$. Notice that this definition of B_t coincides with the previous definition of B_l for $t = l$.

- For $t \leq l$, let A_t be the event that each pair $\langle x, v_1\rangle, \langle v_1, v_2\rangle, \ldots, \langle v_{t-2}, v_{t-1}\rangle$ consists of vertices assigned to different color classes.
- For $t \leq l$, let B_t be the event that A_t occurs *and* in addition vertex v_{t-1} is given a different color than y.

For convenience, let $A_t - B_t$ denote the event that A_t occurs and B_t does not. Since $B_t \Rightarrow A_t$, we have $\mathbf{Pr}[A_t - B_t] = \mathbf{Pr}[A_t] - \mathbf{Pr}[B_t]$. Also note that event A_t does not depend on whether x and y are chosen in the same or different color class.

The probability of event A_t is easy to calculate: we just need v_1 a different color from x, v_2 a different color from v_1, and so on up to v_{t-1}. Thus:

$$\mathbf{Pr}[A_t] = (1-1/k)^{t-1}. \tag{2}$$

For $t = 2$, event B_2 is the event that v_1 is a different color from both x and y, so $\mathbf{Pr}^{\text{same}}[B_2] = 1 - 1/k$ and $\mathbf{Pr}^{\text{diff}}[B_2] = 1 - 2/k$. For $t > 2$, event B_t occurs if either: (1) B_{t-1} occurs *and* v_{t-1} is of one of the $k-2$ colors not used in y or v_{t-2}, or (2) event $A_{t-1} - B_{t-1}$ occurs *and* v_{t-1} is of one of the $k-1$ colors not used in y or v_{t-2}. Thus,

$$\begin{aligned}\mathbf{Pr}^{\text{same}}[B_t] &= \mathbf{Pr}^{\text{same}}[B_{t-1}](1-2/k) + \Big(\mathbf{Pr}[A_{t-1}] - \mathbf{Pr}^{\text{same}}[B_{t-1}]\Big)(1-1/k) \\ &= (1-1/k)^{t-1} - \tfrac{1}{k}\mathbf{Pr}^{\text{same}}[B_{t-1}]. \end{aligned} \tag{3}$$

$$\text{Similarly,} \quad \mathbf{Pr}^{\text{diff}}[B_t] = (1-1/k)^{t-1} - \tfrac{1}{k}\mathbf{Pr}^{\text{diff}}[B_{t-1}]. \tag{4}$$

We can now solve for $\mathbf{Pr}^{\text{same}}[B_l]$ as follows.

$$\begin{aligned}\mathbf{Pr}^{\text{same}}[B_l] &= (1-1/k)^{l-1} - \tfrac{1}{k}\Big[(1-1/k)^{l-2} - \tfrac{1}{k}\mathbf{Pr}^{\text{same}}[B_{l-2}]\Big] \\ &\;\;\vdots \\ &= (1-1/k)^{l-1} - \tfrac{1}{k}(1-1/k)^{l-2} + \ldots + (-1)^{l-2}\left(\tfrac{1}{k}\right)^{l-2}\mathbf{Pr}^{\text{same}}[B_2]. \end{aligned} \tag{5}$$

Similarly,

$$\mathbf{Pr}^{\text{diff}}[B_l] = (1-1/k)^{l-1} - \tfrac{1}{k}(1-1/k)^{l-2} + \ldots + (-1)^{l-2}\left(\tfrac{1}{k}\right)^{l-2}\mathbf{Pr}^{\text{diff}}[B_2]. \tag{6}$$

Finally, from equations (5) and (6), we have:

$$\begin{aligned}\mathbf{Pr}^{\text{same}}[B_l] - \mathbf{Pr}^{\text{diff}}[B_l] &= (-1)^{l-2}(\tfrac{1}{k})^{l-2}\Big[\mathbf{Pr}^{\text{same}}[B_2] - \mathbf{Pr}^{\text{diff}}[B_2]\Big] \\ &= (-1)^{l}(\tfrac{1}{k})^{l-2}[(1-1/k) - (1-2/k)] \\ &= (-1)^{l}(\tfrac{1}{k})^{l-1} \qquad (7)\end{aligned}$$

as claimed. ■

By Theorem 1 and equation (1), we have $\lambda_l = [(-1)^l/k^{l-1}]/\mathbf{Pr}[B_l]$, so:

$$|\lambda_l| \geq 1/k^{l-1}. \qquad (8)$$

Thus, for l and k both constant, $|\lambda_l|$ is bounded away from 0 by some constant > 0 as desired. One final fact to note for this section is that E_l^{same} and E_l^{diff} are both $\Theta(p^l(n-2)_{l-1}) = \Theta(p^l n^{l-1})$. So, if $p \geq n^{-1+\epsilon}$ for some constant $\epsilon > 0$ then for sufficiently large constant l, both E_l^{same} and E_l^{diff} are large (eg., $\geq n^{1/3}$) and they differ by a constant factor.

3.3 The l-path algorithm

Note the following property of paths of constant length l between fixed vertices x and y. The number of edges in the path divided by the number of "non-rooted" vertices (that is, vertices not including x and y) is $l/(l-1)$. For any proper subgraph S of such a path, the quantity $|E(S)|/|V(S) - \{x,y\}|$ is strictly smaller. Because this ratio of edges to non-rooted vertices is strictly less for all proper subgraphs, we say that paths between x and y are "strictly-balanced".

Spencer [14] proves that for any such strictly balanced graph and any constants $\delta, c > 0$, if the expected number of copies of the graph in $G \leftarrow \mathcal{G}(n,p)$ is at least $K \log n$ for sufficiently large K, then the actual number of copies of the graph in $G \leftarrow \mathcal{G}(n,p)$ will be within $(1+\delta)$ of the expectation with probability $1 - o(n^{-c})$. We prove in this paper an analog for the model $\mathcal{G}(n,p,k)$ for the case of constant-length paths. For a given graph G, let $\mathrm{Num}_l(x,y)$ be the number of paths of length l between x and y in the graph.

Theorem 2 *For any constants $\delta, c > 0$, if l and p are such that $K \log n \leq E_l^{\text{same}}(p), E_l^{\text{diff}}(p)$ for sufficiently large K, then for $G \leftarrow \mathcal{G}(n,p,k)$:*

1. $\mathbf{Pr}^{\text{same}}\Big[(1-\delta)E_l^{\text{same}} < \mathrm{Num}_l(x,y) < (1+\delta)E_l^{\text{same}}\Big] \geq 1 - o(n^{-c})$,

2. $\mathbf{Pr}^{\text{diff}}\Big[(1-\delta)E_l^{\text{diff}} < \mathrm{Num}_l(x,y) < (1+\delta)E_l^{\text{diff}}\Big] \geq 1 - o(n^{-c})$.

Proof: The proof is somewhat long and is similar to an argument in [14], and so is deferred to Section 5. ■

So, if the expected number of paths is sufficiently large, then we can be assured that with probability $1 - o(n^{-2})$, the number of paths between x and y will be close to the expectation. We now present the algorithm l-**path**.

Algorithm l-path

> **Given:** *An n-vertex k-colorable graph G.*
>
> **Output:** *A k-coloring of G or else failure.*

1. *Let d_{avg} be the average degree in G and let $\hat{p} = \frac{d_{\text{avg}}}{n(1-1/k)}$.*
   (So, if $G \leftarrow \mathcal{G}(n,p,k)$ for $p = n^{-1+\epsilon}$ then with high probability, $\hat{p} = p[1+o(1)]$.)
2. *Pick the smallest l so that $\hat{p}^l n^{l-1} \geq n^{1/3}$ and let $\hat{\lambda} = 1/k^{l-1}$ (a lower bound on $|\lambda_l|$ by equation (8)).*
   Note: for n sufficiently large, $E_l^{\text{same}}(p)$ and $E_l^{\text{diff}}(p)$ will both be larger than $K \log n$ w.h.p. where K is such that Theorem 2 holds for $c = 2$ and $\delta = \hat{\lambda}/4$.
3. *Calculate $E_l^{\text{same}}(\hat{p})$ using $E_l^{\text{same}}(\hat{p}) = \hat{p}^l (n-2)_{l-1} \mathbf{Pr}^{\text{same}}[B_l]$ and equation (5).*
4. *For $i = 1$ to k do:*
   (a) *Pick an arbitrary uncolored vertex x and let S_i be the set containing x and all vertices with a number of simple paths of length l to x in the range:*
   $$[(1-\hat{\lambda}/3)E_l^{\text{same}}(\hat{p}), (1+\hat{\lambda}/3)E_l^{\text{same}}(\hat{p})].$$
   If the set S_i is not independent or S_i contains previously-colored vertices, then halt with failure.

   (b) *If S_i is independent, then assign color i to all vertices of S_i.*
5. *If in Step 4 we assigned one of k colors to each vertex in the graph, then halt with success. If we did not color each vertex, then halt with failure.*

Theorem 3 *Algorithm l-**path** k-colors graphs $G \leftarrow \mathcal{G}(n,p,k)$ with high probability for $p \geq n^{-1+\epsilon}$ for any constant $\epsilon > 0$.*

Proof: Let $C_1, \ldots, C_k$ be the sets of vertices in each color class in the creation of graph G. Let us say that Step 4 *succeeds* in iteration i if the set S_i created equals C_j for some $1 \leq j \leq k$.

In Step 1, as noted, with high probability $\hat{p} = p[1+o(1)]$, and let us for convenience assume now that this is the case. So, $E_l(\hat{p}) = [1+o(1)]E_l(p)$ and $E_l^{\text{same}}(\hat{p}) = [1+o(1)]E_l^{\text{same}}(p)$.

In Step 2, $E_l(p) \geq K \log n$ and Theorem 2 applies. Also, δ is chosen sufficiently small so that the range $[(1-\hat{\lambda}/3)E_l^{\text{same}}(\hat{p}), (1+\hat{\lambda}/3)E_l^{\text{same}}(\hat{p})]$ contains the range $[(1-\delta)E_l^{\text{same}}(p), (1+\delta)E_l^{\text{same}}(p)]$ and does not intersect $[(1-\delta)E_l^{\text{diff}}(p), (1+\delta)E_l^{\text{diff}}(p)]$. So, with probability $1 - n^2[o(n^{-2})] = 1 - o(1)$, for every pair of vertices x, y in the same color class, and for no pairs x, y in different color classes, the number of paths of length l between x and y is in the range used in Step 4. Thus, with high probability, Step 4 succeeds for each iteration i and Algorithm l-**path** k-colors the graph. ∎

The running time of the algorithm is dominated by the time needed in Step 4 to compute the number of simple paths of length l between a given vertex x and all the other vertices in the graph. If we assume we have the graph as an adjacency list, this computation can be done in time $O(n+d^l)$, where d is the maximum degree of the graph, as follows. We perform a depth-first search from x to distance l, marking vertices so that we do not visit any twice on a path. Each time we visit a vertex at depth l, we increment a counter associated with that vertex. Since $d \leq 2pn$ with high probability, we can rewrite the time bound as $O(n + (pn)^l)$. This quantity is just $O(n^2)$ since we chose l in Step 2 to be the least value such that $(\hat{p}n)^l \geq n^{1/3+1}$: that is, if $\hat{p} \geq n^{-1/3}$ then $n^{4/3} \leq (\hat{p}n)^2 \leq n^2$, and if $\hat{p} < n^{-1/3}$ then $n^{4/3} \leq (\hat{p}n)^l \leq n^2$ for some $l \geq 3$. So, with high probability, the running time of the algorithm is $O(kn^2)$.

4 Semi-random graphs

4.1 Basic definitions and statement of results

We define here two graph models both based on the *semi-random* source (also called a "slightly-random" source) of Santha and Vazirani [13] (see also [17, 18, 9]). In the first model, which we denote $\mathcal{G}_S(n, p, k)$, the graph is generated as follows. First, an adversary splits the n vertices into k color classes (for $k = 3$, we denote these classes by red, blue, and 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. Later choices of the adversary may depend on the outcomes of earlier decisions, as in the Santha-Vazirani source [13]. We call p the *noise rate* of the source, and this model the *semi-random* graph model.

The second model we consider is a slightly modified version of the above, differing in that we require the size of each color class to be $\Omega(n)$. We call this second model the *balanced* semi-random graph model and denote it by $\mathcal{G}_{SB}(n, p, k)$. Following the notation in Section 3, we write:

$$G \leftarrow \mathcal{G}_S(n, p, k) \quad \text{or} \quad G \leftarrow \mathcal{G}_{SB}(n, p, k)$$

to denote that G is selected according to the corresponding model for *some* unknown adversary. We denote the semi-random and balanced semi-random models for a *fixed* adversary $\mathcal{A}$ by $\mathcal{G}_S^{\mathcal{A}}(n, p, k)$ and $\mathcal{G}_{SB}^{\mathcal{A}}(n, p, k)$ respectively. Formally, we say that an algorithm t-colors $G \leftarrow \mathcal{G}_S(n, p, k)$ with high probability (or t-colors $G \leftarrow \mathcal{G}_{SB}(n, p, k)$ with high probability) if it does so for any fixed adversary strategy. (Note: as in the random case, the assignment of vertices to color classes is *not* told to the algorithm.)

A third model, which we call the *colorgame* model, is perhaps more conceptually elegant. We begin with n vertices split into k color classes, each of size precisely n/k. A random graph G_1 is created by joining each pair of vertices from different color classes with independent probability p. Then an adversary may place additional edges between nodes of different color classes giving a graph G_2 — with no restriction on how many such edges are added. The algorithm (without knowing the original color classes of course) must now find a k-coloring of G_2. Note that the adversary here may make its decision after all coin tosses have been performed. All the algorithms presented in this section for the semi-random models work also for the colorgame version.

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 any coloring algorithm, it turns out that it does not. While some of the algorithms for the random model [10, 11] are highly dependent on facts such as the edge probabilities all being equal, others such as Turner's No-Choice algorithm [16] adapt well to the semi-random model. In fact, Turner's bound of $p \geq n^{-1/k+\epsilon}$ for k-coloring $G \leftarrow \mathcal{G}(n,p,k)$ holds in the balanced semi-random model as well.

We present first an algorithm that achieves the same bound as Turner's algorithm for $k = 3$ but with significantly simpler analysis, and that holds in the slightly more general $\mathcal{G}_S(n,p,3)$ model. We then present an algorithm with better bounds for the balanced model: this algorithm 3-colors graphs in $\mathcal{G}_{SB}(n,p,3)$ with high probability for $p \geq n^{-0.6+\epsilon}$, and more generally k-colors graphs in $\mathcal{G}_{SB}(n,p,k)$ for $p \geq n^{\left[\frac{-2k}{(k+1)k-2}\right]+\epsilon}$. The improved algorithm requires a more involved analysis, and the use of the Janson inequality for estimating probabilities of "almost" independent events. In Section 4.7 we present some relationships between the coloring problems in the balanced and unbalanced semi-random models.

For convenience, we make the following definition.

Definition 1 *Let $G \leftarrow \mathcal{G}_S(n,p,k)$ or $G \leftarrow \mathcal{G}_{SB}(n,p,k)$. The pair (u,v) is a* **potential edge** *in G if u and v belong to different color classes in the adversary's color scheme.*

For a subgraph H of G or a subset U of $V(G)$, we will use $colors(H)$ and $colors(U)$ to denote the set of color classes of G that are represented in the subgraph or subset.

4.2 A first algorithm

We now describe a simple algorithm for the model $\mathcal{G}_S(n,p,3)$ of a 3-colorable graph generated by a semi-random source. The idea is the following. If in the adversary's color scheme $u \in$ blue and $v \in$ green, then the shared neighborhood $S = N(u) \cap N(v)$ contains only red vertices. Thus, $N(S) \subseteq$ blue $\cup$ green. For $p \geq n^{-1/3+\epsilon}$, we show that with high probability $N(S)$ actually equals the entire set blue $\cup$ green. So, given u and v, one can split G into a 2-colorable portion $N(S)$ and an independent portion $V - N(S)$ and thus 3-color the graph.

Algorithm Two-Stage

Given: *A graph $G = (V,E)$.*

Output: *Either a 3-coloring of G or failure.*

1. *First try to 2-color G. If that works, halt with success. Otherwise, do the following:*
2. *For each pair of vertices u, v (think of u as a candidate green node and v as a candidate blue node),*
   (a) *Let $S = N(u) \cap N(v)$.*

(b) *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.

3. *If Step 2 did not succeed for any pair u, v, then halt with failure.*

Theorem 4 *Algorithm* **Two-Stage** *3-colors $G \leftarrow \mathcal{G}_S(n, p, 3)$ 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 and v 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 $T = N(S) \subseteq \mathsf{blue} \cup \mathsf{green}$. We now prove that for $p \geq n^{-1/3+\epsilon}$, with high probability $T = \mathsf{blue} \cup \mathsf{green}$.

The performance of the algorithm can clearly only improve if the adversary puts more edges into G, so we may assume that the adversary always elects *not* to place edges into the graph. In that case, every vertex in red independently has a probability p^2 of belonging to S. So, using Chernoff bounds, $|S| \geq \frac{1}{2}|\mathsf{red}|p^2 = \Omega(np^2)$ with high probability. Now, each vertex $z \in \mathsf{blue} \cup \mathsf{green}$ 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 $w \in \mathsf{red}$, 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:

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

That is, with high probability *all* vertices $z \in \mathsf{blue} \cup \mathsf{green}$ belong to T. Thus, with high probability, $T = \mathsf{blue} \cup \mathsf{green}$ and $V - T = \mathsf{red}$ and so for some pair u, v considered, algorithm **Two-Stage** succeeds. ■

Note that if the sizes of the color classes are roughly balanced, we can speed up Algorithm **Two-Stage** considerably by choosing the vertices u and v at *random*. For instance, if the sizes of the color classes are all within constant factors, then we have a constant probability of selecting two "good" vertices each time.

Algorithm **Two-Stage** fails when p falls below $n^{-1/3}$ because then the vertices $u \in \mathsf{green}$ and $v \in \mathsf{blue}$ may not share enough neighbors for $N(S)$ to equal $\mathsf{blue} \cup \mathsf{green}$. However, for p below $n^{-1/3}$, set S might still contain many vertices, and applying additional iterations of the neighbor-taking process can then boost its size if the sizes of the blue, green, and red vertex sets are roughly balanced. In fact, this can be shown to color graphs $G \leftarrow \mathcal{G}_{SB}(n, p, 3)$ with high probability for $p \geq n^{-1/2+\epsilon}$ and $\epsilon > 0$ constant, in polynomial time (see [5]).

4.3 A better algorithm

We now describe a different style of algorithm that improves upon the above bound in the balanced case, and 3-colors graphs in $\mathcal{G}_{SB}(n, p, 3)$ with high probability for $p \geq n^{-0.6+\epsilon}$ and more

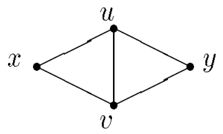

Figure 1: **A link between x and y.**

generally k-colors graphs from $\mathcal{G}_{SB}(n,p,k)$ for $p \geq n^{\left[\frac{-2k}{(k+1)k-2}\right]+\epsilon}$. The algorithm, while quite simple, requires a more involved probabilistic analysis than the previous one. In particular, we will need to use the Janson inequality [6] to bound probabilities of "nearly" independent events based on pairwise dependencies.

The algorithm is based on the following simple observation. If in a 3-colorable graph G there are two vertices x and y both adjacent to a pair of vertices u and v that are adjacent to each other, then x and y must be the same color in any legal 3-coloring. We call the subgraph induced by $\{x, u, v, y\}$ a *link* between x and y. (See Figure 1).

At first glance, it would seem the above observation does not help, since for fixed vertices x and y, the probability there exists a link between x and y even if G is chosen from $\mathcal{G}(n,p,3)$ is at most $O(n^2p^5)$. (There are $O(n^2)$ possible pairs $\langle u, v\rangle$ and for each pair the probability all necessary edges exist is p^5.) Thus, the probability there is a link between x and y is much less than 1 even for $p = o(n^{-0.4})$. The key fact to note, however, is that we do not need a link between every pair of, say, red vertices x and y. All we need is that for each such pair there is a *sequence* of links between x and some x', between x' and some x'', and so forth, until eventually at some point we reach y. We will call such a sequence a "chain".

Another way to think of this observation is that given a graph G we can create a new graph H as follows. The vertex set $V(H)$ equals $V(G)$, and if x and y are connected by a link in G, we put an edge between x and y into H. So, while edges in G exist only between vertices of different color, edges in H exist only between vertices of the *same* color (in G). The "key observation" is then just that in order to easily select the set of red vertices in G, we do not need red to be a *clique* in H, just a connected set. So, the algorithm for $k = 3$ is just to create the graph H and output the components of H as the color classes of G; if there are more than 3 components of H, then the algorithm halts with failure.

For more general constant k, we just replace the notion of a "link" by that of a "t-link" defined as follows.

Definition 2 *A* **t-link** *for some constant t is a $(t+2)$-vertex graph consisting of two vertices x and y called the* endpoints *both fully connected to a t-clique. (See Figure 2).*

Equivalently, a t-link between x and y is a $(t+2)$-clique with the edge (x, y) removed.

Note that if two vertices in a k-colorable graph are endpoints of a $(k-1)$-link, then they must be the same color in any legal k-coloring. We can thus get the following natural generalization of the algorithm described above to graphs of constant chromatic number $k \geq 3$.

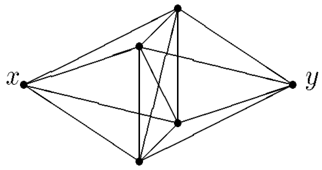

Figure 2: **A 4-link between x and y.**

Algorithm Link

Given: *$G = (V, E)$, and a constant $k \geq 3$.*

Output: *A k-coloring of G or else failure.*

1. *Create graph $H = (V, F)$, where*

$$F = \{(x, y) \mid \exists \text{ a } (k-1)\text{-}link \text{ in } G \text{ between } x \text{ and } y\}.$$

2. *Find all connected components in H. If there are exactly k components, then halt with success, producing those components as the color classes of G.*

   Otherwise, if there are more than k components, then halt with failure.

It is clear from the description of this algorithm that the performance can only improve if the adversary decides to add additional edges into the graph. Therefore, it is sufficient to prove success for the case where each edge between vertices from different color classes (in the adversary's color scheme) is placed into the graph with probability exactly p.

4.4 Motivation

Let us consider the case of $k = 3$. As mentioned above, each connected component in graph H produced by Algorithm **Link** consists of vertices that *must* be the same color under any legal 3-coloring of G. The following sections contain a proof that when $p \geq n^{-0.6+\epsilon}$, with high probability there will be only 3 such components in H. Let us first, however, give a motivational argument, supposing that each edge between two vertices of the same color were placed independently with the same probability into H.

Let $p = n^{-0.6+\epsilon}$. Given two vertices x and y of the same color (say red) in G, the expected number of links between x and y in $G = \Omega(n^2p^5) = \Omega(n^{-1+5\epsilon})$. For $\epsilon < 1/5$, the *probability* there exists a link between x and y, and thus the probability that x and y are are neighbors in H is $\Omega(n^{-1+5\epsilon})$ as well. Thus, if we consider the subgraph in H induced by the set red, the average degree of each vertex is $\Omega(n^{5\epsilon})$. It is well known that in the random graph model $\mathcal{G}(n, p)$, once the average degree exceeds $K \log n$ for sufficiently large constant K, the graph is connected with high probability. So, if the edges in the red subgraph of H were placed *randomly*, the red set would be a single connected component almost surely since $n^{5\epsilon} \gg K \log n$.

4.5 Janson's inequality

Janson's inequality [6] is used in the following setting. Consider a universe U of points and a collection of subsets $X_1, \ldots, X_m$ of U. We now create a new subset S of U by placing each $j \in U$ into S independently with probability p_j. Let A_i be the event that $X_i \subseteq S$. Janson's inequality bounds the probability that no set X_i is contained within S: that is, the probability that no event A_i occurs.

Define:

$$M = \prod_{i=1}^{m} \mathbf{Pr}[\bar{A}_i].$$

If the X_i were all disjoint, then the events A_i would all be independent and so M would be the probability that no A_i occurs. If the X_i are *not* disjoint, then the events A_i are *not* independent. However, Janson's inequality allows us to bound the probability no X_i is contained in S by looking only at *pairwise* dependencies. In particular, Janson's inequality states that:

$$M \leq \mathbf{Pr}\left[\bigwedge_i \bar{A}_i\right] \leq M e^{\frac{1}{1-\lambda}\frac{\Delta}{2}} \tag{9}$$

where λ is an upper bound on $\mathbf{Pr}[A_i]$ and we define:

$$\Delta = \sum_{\substack{\text{ordered pairs } (i \neq j) \\ X_i \cap X_j \neq \phi}} \mathbf{Pr}[A_i \textbf{ and } A_j].$$

Notice that if $\lambda \leq 1/2$ and $\Delta = o(1)$, then by equation (9), $\mathbf{Pr}[\text{no } X_i \text{ is contained in } S] = M(1 + o(1))$. That is, under these two conditions, the probability is within $1 + o(1)$ of what the probability would be had the A_i been independent.

In the study of random graphs Janson's inequality is often used to show that some structure exists with high probability. For example, suppose one wishes to prove that the graph $G \leftarrow \mathcal{G}(n,p)$ contains a triangle with high probability for $p \gg 1/n$. For such a setting, we let U be the set of all edges of the n-clique K_n (thought of as possible edges in G) and have one X_i for each set of three edges corresponding to a triangle. Janson's inequality then provides an upper bound on the probability that *no* triangle is contained in G. Here we will use Janson's inequality to prove that in the balanced semi-random model, for sufficiently large noise rate p, for any $x, y \in \mathsf{red}$ there will be a chain between x and y with probability at least $1 - o(n^{-2})$.

The following definitions are taken (roughly) from [14]. [1]

Definition 3 *Let H be a graph in which some subset R of its vertices are specified to be "roots" where there are no edges between vertices in R. We will call the pair (R, H) a* **rooted graph**, *or simply say that H is a rooted graph when R is implicit. Define* edges(H) *to be the total number of edges in H and* nonroots(H) *to be the number of vertices in H excluding roots. Define the density of H to be* dens(H) = edges$(H)/$nonroots(H).

[1] The term "image" used here is essentially the same as an "extension" in [14], except that paper counts maps while we count images of maps.

We will always consider rooted graphs to be graphs on a constant number of vertices, and examine the number of copies of such graphs in larger n-vertex graphs.

Definition 4 *A rooted graph (R, H) is* **strictly balanced** *if for some constant $\epsilon' > 0$, for every proper subgraph (R, H') (that is, $R \subseteq H' \subset H$), we have $dens(H') \leq dens(H) - \epsilon'$.*

Definition 5 *Suppose (R, H) is a rooted graph and $G = (V, E)$ is a graph with $V \supseteq R$. An* **image of** *H* **over** *R in G is a subgraph of G isomorphic to H by a map which is the identity function on the vertex set R. When R is clear from context, we will drop the phrase "over R".*

So, for example, if H is a triangle with a root vertex x, then the images of H over $\{x\}$ in G are all triangles in G containing vertex x.

Definition 6 *Suppose (R, H) is some strictly balanced rooted graph and V is a set of n vertices containing R. Let $X_1, \ldots, X_m$ denote all distinct images of H in the clique K_n on V. That is, the X_i are all possible images of H fixing root set R in an n-vertex graph. For some model $\mathcal{M}$ (such as $\mathcal{G}(n,p)$), we define:*

$$\Delta(H, \mathcal{M}) = \sum_{\substack{\text{ordered pairs } i \neq j \\ E(X_i) \cap E(X_j) \neq \phi}} \mathbf{Pr}[X_i \subseteq G \text{ and } X_j \subseteq G \mid G \leftarrow \mathcal{M}].$$

Spencer [15, 14] proves the following theorem for random graphs $\mathcal{G}(n,p)$.

Theorem 5 ([14]) *Let (R, H) be a strictly balanced rooted graph on a constant number of vertices with $\mathbf{v} = \text{nonroots}(H)$ and $\mathbf{e} = \text{edges}(H)$. Then there exists $\epsilon^* > 0$ so that if $p \leq n^{-\mathbf{v}/\mathbf{e}+\epsilon^*}$, then $\Delta(H, \mathcal{G}(n,p)) = o(1)$.*

We can use Theorem 5 to prove the following.

Theorem 6 *Let $\mathcal{G}_{SB}^{\mathcal{A}}(n,p,k)$ be the semi-random model with an adversary that first selects the colors of the vertices according to some strategy and then always elects* not *to place edges into the graph. Let (R, H) be a strictly balanced rooted graph on a constant number of vertices with $\mathbf{v} = \text{nonroots}(H)$ and $\mathbf{e} = \text{edges}(H)$. Then there exists $\epsilon^* > 0$ so that if $p \leq n^{-\mathbf{v}/\mathbf{e}+\epsilon^*}$, then $\Delta(H, \mathcal{G}_{SB}^{\mathcal{A}}(n,p,k)) = o(1)$.*

Proof: Let $X_1, \ldots, X_m$ denote the images of H in the clique K_n and let A_i be the event that $X_i \subseteq G$. Each edge (x, y) is placed into G with probability *at most* p (either probability 0 if x and y are in the same color class or else probability p if they are in different color classes). Thus, for any pair of events A_i, A_j, we have:

$$\mathbf{Pr}[A_i \wedge A_j \mid G \leftarrow \mathcal{G}_{SB}^{\mathcal{A}}(n,p,k)] \leq \mathbf{Pr}[A_i \wedge A_j \mid G \leftarrow \mathcal{G}(n,p)].$$

So, Theorem 6 follows immediately from Theorem 5.

For sake of completeness, however, we provide a direct proof here as well following the argument of Spencer [14].

We prove the theorem by considering separately for each fixed value of $s \in \{1, \ldots, \mathbf{v}\}$, the pairs X_i, X_j that share s vertices in common in addition to the roots. Note that if $s = 0$, then the graphs X_i and X_j share no edges and so are not counted in the summation in Definition 6. The number of pairs X_i and X_j sharing s vertices in common in addition to the roots is $O(n^{2\mathbf{v}-s})$ since there are $2\mathbf{v} - s$ possible vertices and only a constant number of permutations.

Let $\epsilon' > 0$ be a value such that every proper subgraph of H containing the roots has density at most $dens(H) - \epsilon'$ (see the definition of "strictly balanced"). If $s = \mathbf{v}$, then since X_i and X_j are distinct, there must be at least $\mathbf{e} + 1$ edges in $X_i \cup X_j$. For any fixed edge, the probability that edge belongs to G is at most p (it could be smaller, e.g. 0, if the two endpoints are in the same color class in the graph). So, the contribution to the summation from such pairs X_i and X_j is at most: $O(n^{\mathbf{v}}p^{\mathbf{e}+1}) = O(n^{\mathbf{v}+(\mathbf{e}+1)(-\mathbf{v}/\mathbf{e}+\epsilon^*)}) = O(n^{-\mathbf{v}/\mathbf{e}+(\mathbf{e}+1)\epsilon^*}) = o(1)$, for ϵ^* sufficiently small.

Now consider $s < \mathbf{v}$ and fix a pair X_i and X_j sharing s vertices in common in addition to roots. Let S be the subgraph $X_i \cap X_j$; that is, $V(S) = V(X_i) \cap V(X_j)$ and $E(S) = E(X_i) \cap E(X_j)$. Since (R, H) is strictly balanced, we know that $|E(S)|/s \leq \mathbf{e}/\mathbf{v} - \epsilon'$ for some $\epsilon' > 0$. So, $|E(X_i) \cup E(X_j)| = 2\mathbf{e} - |E(S)| \geq 2\mathbf{e} - s\mathbf{e}/\mathbf{v} + s\epsilon'$ and thus the probability that both X_i and X_j are subgraphs of G is at most $p^{2\mathbf{e}-s\mathbf{e}/\mathbf{v}+s\epsilon'}$.

Finally, summing over all $O(n^{2\mathbf{v}-s})$ pairs X_i and X_j sharing s vertices in common besides the endpoints, the contribution to Δ is at most:

$$\begin{aligned} O(n^{2\mathbf{v}-s}p^{2\mathbf{e}-s\mathbf{e}/\mathbf{v}+s\epsilon'}) &= O(n^{2\mathbf{v}-s}(n^{-\mathbf{v}/\mathbf{e}+\epsilon^*})^{2\mathbf{e}-s\mathbf{e}/\mathbf{v}+s\epsilon'}) \\ &= O(n^{-s\epsilon'\mathbf{v}/\mathbf{e}+(2\mathbf{e}-s\mathbf{e}/\mathbf{v}+s\epsilon')\epsilon^*}) \\ &= o(1) \qquad \text{(for } \epsilon^* \text{ sufficiently small).} \end{aligned}$$

Thus, the contribution to Δ from each value of $s \in \{1, \ldots, \mathbf{v}\}$ is $o(1)$, and since there are only a constant number of different choices for s, we have $\Delta(H, \mathcal{G}_{SB}^{\mathcal{A}}(n, p, k)) = o(1)$. ∎

4.6 The main theorem

We now prove the following theorem.

Theorem 7 *For constant $k \geq 3$, algorithm* **Link** *k-colors $G \leftarrow \mathcal{G}_{SB}(n, p, k)$ for $p \geq n^{\left[\frac{-2k}{(k+1)k-2}\right]+\epsilon}$, ($\epsilon > 0$) with high probability.*

For example, for $k = 3$, the algorithm succeeds for $p \geq n^{-3/5+\epsilon}$ and for $k = 4$ it succeeds for $p \geq n^{-4/9+\epsilon}$.

In order to prove Theorem 7, we consider $(k-1)$-chains of some constant length r. We then prove that $(k-1)$-chains are strictly-balanced, so Theorem 6 applies. Finally, using Janson's inequality, we will get that for any fixed x and y of the same color class, there is some t-chain between x and y with probability $1 - o(n^{-2})$.

Let us first formally define a t-chain.

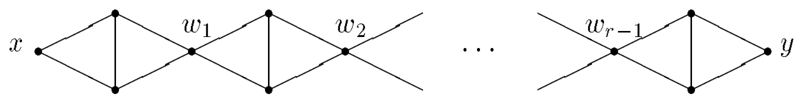

Figure 3: **A 2-chain of length r between x and y.**

Definition 7 *A t-**chain** of length r is a sequence of r t-links connected at their endpoints. For a t-chain C with fixed endpoints x and y, we will treat the chain as a rooted graph, with x and y as the roots. (See figure 3.)*

Fact 8 *If C is a t-chain of length r, then $|V(C)| = r(t+1)+1$, $nonroots(C) = r(t+1)-1$, and $|E(C)| = edges(C) = r\left[\binom{t+2}{2} - 1\right] = \frac{r}{2}[(t+1)(t+2)-2]$. So, $\frac{|E(C)|}{|V(C)|-1} = \frac{1}{2}[t+2-\frac{2}{t+1}] = \frac{t}{2}+\frac{t}{t+1}$.*

Note that if C_r is a $(k-1)$-chain of length r, then the term $\left[\frac{-2k}{(k+1)k-2}\right]$ in the statement of Theorem 7 equals $\lim_{r\to\infty} \frac{-nonroots(C_r)}{edges(C_r)}$. So, for sufficiently large r, the edge probability p is of the form needed in Theorem 6 to prove that $\Delta = o(1)$.

Our first step is to prove that t-chains are strictly-balanced. To do this, we need the following lemma. (The reader may wish to skip the somewhat tedious proofs of lemmas 8 and 9.)

Lemma 8 *Let S be a subgraph of a t-chain C of length r. Then:*

$$|E(S)| \le \left[\frac{t}{2} + \frac{t}{t+1}\right](|V(S)|-1).$$

Proof:

Let L be a t-link and let $g(t) = \frac{t}{2} + \frac{t}{t+1}$. Define $dens_1(H) = \frac{|E(H)|}{|V(H)|-1}$, so:

$$dens_1(L) = dens_1(C) = g(t). \tag{10}$$

Claim: If $S \subseteq L$, then $dens_1(S) \le g(t)$.

Proof: We may assume S is vertex-induced. Thus, S is either a $(t+2-c)$-clique or else S is a $(t-c)$-link for some $c \ge 1$. In the latter case, the claim follows from equation (10) since $g(t)$ is an increasing function of t. In the former case, we have $dens_1(S) = \frac{t+2-c}{2} = \frac{t}{2}+1-\frac{c}{2} < g(t)$ for $c \ge 1$. $\square$

Now, we prove the lemma that if $S \subseteq C$ then $dens_1(S) \le g(t)$ by induction on the length of C. The base case is proved in the above claim, so we may assume the lemma holds for any t-chain of length $r-1$. Let C' be the t-chain of length $r-1$ consisting of the first $r-1$ links in C and let S' be S restricted to C'. So, $dens_1(S') \le g(t)$. Let L be the last link in C and let S_L be S restricted to L. So, $|E(S)| = |E(S')| + |E(S_L)|$ and $|V(S)| \ge |V(S')| + |V(S_L)| - 1$ (note: S' and S_L may share one vertex in common where L joins C'). Thus, $dens_1(S) \le \frac{|E(S')|+|E(S_L)|}{|V(S')|+|V(S_L)|-2} = \frac{|E(S')|+|E(S_L)|}{[|V(S')|-1]+[|V(S_L)|-1]} \le g(t)$ by induction and the Claim. ■

We can now use Lemma 8 to prove that chains are strictly-balanced, and so allow easy application of Janson's inequality.

Lemma 9 *Let S be a subgraph of a t-chain C of some constant length r such that $V(S)$ contains the endpoints x and y but $V(S) \neq V(C)$. Then, for some constant $\epsilon' = \epsilon'(r) > 0$,*

$$dens(S) \leq dens(C) - \epsilon'.$$

That is, t-chains are strictly-balanced.

Proof: For C a t-chain of length r, we have $edges(C) = \frac{r}{2}[(t+1)(t+2)-2] = \frac{r}{2}[t^2+3t]$. So, we may upper bound the density as follows:

$$\begin{aligned} dens(C) &= \frac{\frac{r}{2}[t^2+3t]}{r(t+1)-1} \\ &= \frac{t^2+3t}{2t+2-2/r} \\ &\leq \frac{t+3}{2}. \end{aligned}$$

Again, we may assume that S is a vertex-induced subgraph.

First, suppose S consists of just one connected component. So, S contains at least one non-endpoint for each t-link L in C, and contains all link endpoints. Let us focus on some fixed t-link L in C. If S is missing m vertices from that link, then it must be missing *at least* $(t+1)+t+(t-1)+\ldots+(t-m+2)$ edges from that link as well. So, the ratio of edges missing to vertices missing is at least $\frac{(t+1)+(t-m+2)}{2} \geq \frac{t+4}{2}$, for $m \leq t-1$ which is the largest value of m possible. Thus, if there are in total $\hat{m}$ vertices in C missing from S, then

$$\begin{aligned} \frac{edges(S)}{nonroots(S)} &\leq \frac{edges(C) - \hat{m}(t+4)/2}{nonroots(C) - \hat{m}} \\ &\leq \frac{edges(C)}{nonroots(C)} - \epsilon' \qquad \text{for some } \epsilon' > 0, \end{aligned}$$

because $\frac{t+4}{2} \geq \frac{1}{2} + dens(C)$.

If S consists of more than one connected component, then the endpoints of C cannot be in the same component since we have assumed that S is vertex-induced. We can thus partition S into two disjoint subgraphs: S_{start} and S_{rest} where S_{start} contains root x and S_{rest} is everything else. So, $nonroots(S) = |V(S_{start})| + |V(S_{rest})| - 2$. Let $g(t) = \frac{t}{2} + \frac{t}{t+1}$. Applying Lemma 8, we get:

$$\begin{aligned} edges(S) &= |E(S_{start})| + |E(S_{rest})| \\ &\leq g(t)(|V(S_{start})| - 1) + g(t)(|V(S_{rest})| - 1) \\ &= g(t) nonroots(S). \end{aligned}$$

So, $\frac{edges(S)}{nonroots(S)} \leq \frac{|E(C)|}{|V(C)|-1} = \frac{edges(C)}{nonroots(C)+1} \leq \frac{edges(C)}{nonroots(C)} - \epsilon'$ for $\epsilon' > 0$. ∎

Proof of Theorem 7: As mentioned earlier, we may assume that the adversary $\mathcal{A}$ always elects *not* to place edges into the graph (otherwise, there can only be *more* t-chains). Let $C = C(r)$ be a $(k-1)$-chain of length r between two fixed vertices x and y. So:

$$\begin{aligned} -nonroots(C)/edges(C) &= -(kr-1)/\left(\tfrac{r}{2}[k(k+1)-2]\right) \qquad \text{(see Fact 8)} \\ &= \frac{-2k}{k(k+1)-2} + \frac{2}{r[k(k+1)-2]}. \end{aligned}$$

Thus, for sufficiently large r, for some $\hat{\epsilon} > 0$, we have $p \geq n^{-nonroots(C)/edges(C)+\hat{\epsilon}}$. By Lemma 9 we know C is strictly balanced, so let $\epsilon^* > 0$ be the constant of Theorem 6 such that for $p \leq n^{-nonroots(C)/edges(C)+\epsilon^*}$, we have $\Delta = \Delta(C, \mathcal{G}^{\mathcal{A}}_{SB}(n,p,k)) = o(1)$. Because additional edges cannot decrease the probability of success, we may for the purposes of analysis assume that $\hat{\epsilon} \leq \epsilon^*$. That is,

$$p \;=\; n^{-nonroots(C)/edges(C)+\hat{\epsilon}} \quad \text{for some } r \in \mathbf{Z},\ 0 < \hat{\epsilon} < \epsilon^*. \tag{11}$$

We now examine all potential $(k-1)$-chains C_i of length r between x and y. That is, all images over $\{x,y\}$ in K_n of C, such that the image contains no edges between two vertices in the same color class in the adversary's k-coloring. Since in the balanced semi-random model there are $\Theta(n)$ vertices in each color class and since r is constant, the number of potential $(k-1)$-chains is $m = \Theta(n^{nonroots(C)})$. Because each edge in some such C_i is placed into G with probability p, for any given C_i we have

$$\mathbf{Pr}[C_i \subseteq G] \;=\; p^{edges(C)} \;=\; n^{-nonroots(C)+edges(C)\hat{\epsilon}}.$$

So:

$$\begin{aligned}
M &= \prod_{i=1}^{m} \mathbf{Pr}[C_i \not\subseteq G] \\
&= (1 - n^{-nonroots(C)+edges(C)\hat{\epsilon}})^{\Theta(n^{nonroots(C)})} \\
&\leq e^{-n^{(-nonroots(C)+edges(C)\hat{\epsilon})}\Theta(n^{nonroots(C)})} \\
&= e^{-\Theta(n^{edges(C)\hat{\epsilon}})} \\
&= o(n^{-2}).
\end{aligned}$$

Since $\lambda = \mathbf{Pr}[C_i \subseteq G] = p^{edges(C)} = o(1)$ and $\Delta = o(1)$, we have by Janson's inequality that: $\mathbf{Pr}[x$ and y are *not* connected by a $(k-1)$-chain of length r in $G] = Me^{\frac{1}{1-\lambda}\frac{\Delta}{2}} = M(1+o(1)) = o(1/n^2)$. So, with high probability, all pairs of vertices from the same color class are connected by some such chain and Algorithm **Link** succeeds. ∎

4.7 Relating the balanced and unbalanced models

For graphs of chromatic number 3, we had fairly good performance bounds even in the unbalanced model. However, for graphs of higher chromatic number, the algorithms required the number of vertices in each color class to be roughly balanced. The reason that the unbalanced case is harder is that if a color class is very small, then the noise rate *p as a function of the number of vertices* in that class is dramatically lower. So, if $(k-1)$ color classes each are small, the algorithm is essentially required to solve a problem for a much lower noise rate on the $(k-1)$-chromatic graph defined by those colors. In particular, one gets the following theorem.

Theorem 10 *If BPP $\not\supseteq$ NP, then for $k \geq 4$ there is no polynomial-time algorithm that for some constant $\epsilon > 0$ will k-color graphs in $\mathcal{G}_S(n,p,k)$ with high probability, for $p = n^{-\epsilon}$.*

Proof: Suppose otherwise; that is, there exists an algorithm $\mathcal{B}$ for k-coloring graphs in $\mathcal{G}_S(n,p,k)$ for $p = n^{-\epsilon}$ for some constant $\epsilon > 0$ where $k \geq 4$. We show how to use $\mathcal{B}$ to optimally color an *arbitrary* $(k-1)$-colorable graph in probabilistic polynomial time. Note that for $k \geq 4$, the problem of optimally coloring $(k-1)$-colorable graphs is NP-hard.

Let $G = (V, E)$ be a $(k-1)$-colorable graph on n-vertices. We create a new N-vertex graph $H = (V \cup V', F)$ where V' is a set of vertices of size $n^{3/\epsilon}$ disjoint from V, and $N = n + n^{3/\epsilon}$ as follows. For each pair $u, v \in V$, if $(u,v) \in E$ then let (u,v) be an edge in F as well. Also, independently for each pair $v \in V$ and $v' \in V'$, let (v, v') be an edge of F with probability $1-p$ for $p = N^{-\epsilon}$. Note that there are no edges in F between vertices in V'. Now, feed graph H to algorithm $\mathcal{B}$. If $\mathcal{B}$ k-colors H, then with high probability it must assign at most $k-1$ colors to V and therefore $(k-1)$-color G. The reason is that otherwise there are k vertices in V all given different colors by $\mathcal{B}$, and with probability $(1-p)^k > 1 - kp = 1 - o(1)$, any given vertex in V' is connected to all k of them (and in fact with *extremely* high probability, there will be *some* such vertex in V'). This forces $(k+1)$ colors to be used in H.

The main point of the proof is that an adversary with noise rate $p = N^{-\epsilon}$ can create a k-colorable graph on N vertices in a distribution indistinguishable from that used to create H. In particular, as above, the adversary separates the N vertices into one set V of n vertices and $k-1$ colors, and one set V' of $N-n$ vertices and one color. It then attempts to places edges between vertices in V exactly where they appear in the graph G and to put in all edges between V and V'. Since n is so small (less that $N^{\epsilon/3}$), there are at most $N^{2\epsilon/3}$ potential edges in the set V. So for $p = N^{-\epsilon}$, with probability at least $1 - N^{-\epsilon/3}$ the adversary will be able to place exactly the edges it wishes between vertices in V without any noise at all. Thus, since we assumed that $\mathcal{B}$ can k-color graphs created by such an adversary with high probability, then with high probability $\mathcal{B}$ will k-color the graph H generated from G by our probabilistic polynomial-time procedure as well. ∎

5 Proof of Theorem 2

In this section we prove Theorem 2 which states that for constant l, if $G \leftarrow \mathcal{G}(n,p,k)$ then $\mathrm{Num}_l(x,y)$ is with high probability near its expectation so long as that expectation is sufficiently large. ($\mathrm{Num}_l(x,y)$ is the number of simple paths of length l between x and y.) This proof follows the argument of [14] where it is shown that the number of images of an arbitrary strictly balanced rooted graph (see Definitions 3, 4, and 5) is with high probability near its expectation, so long as the expectation is sufficiently large, in the $\mathcal{G}(n,p)$ model. A special case of that result is the following.

Theorem 11 ([14]) *Let $\delta, c > 0$ and $l > 1$, and let G be selected from $\mathcal{G}(n,p)$. Then $\exists K > 0$ so that if $\mu = E[\mathrm{Num}_l(x,y)] \geq K \log n$, then:*

$$\mathbf{Pr}[(1-\delta)\mu \;\leq\; \mathrm{Num}_l(x,y) \;\leq\; (1+\delta)\mu] = 1 - o(n^{-c}).$$

In order to prove Theorem 2, we need an analog for the model $\mathcal{G}(n,p,k)$. The result is easiest to prove for the special (but main) case where there exists some sufficiently small ϵ so that the expected value of $\mathrm{Num}_l(x,y)$ is at most n^ϵ. We consider that case first.

For convenience, let $\mathcal{G}^0(n,p,k) = \mathcal{G}^{\mathrm{same}}(n,p,k)$ and $\mathcal{G}^1(n,p,k) = \mathcal{G}^{\mathrm{diff}}(n,p,k)$ be the model $\mathcal{G}(n,p,k)$ but where x and y are required to be in the same or different color classes respectively. We first describe how to modify the proof of Theorem 11 to prove the following result.

Theorem 12 *Let $\delta, c > 0$ and $l > 1$, and let G be selected from $\mathcal{G}^j(n,p,k)$. Then, there exist $K, \epsilon > 0$ so that if $\mu = E[\mathrm{Num}_l(x,y)] \in [K \log n, n^\epsilon]$, then:*

$$\mathbf{Pr}[(1-\delta)\mu \;\leq\; \mathrm{Num}_l(x,y) \;\leq\; (1+\delta)\mu] \;=\; 1 - o(n^{-c}).$$

We then extend Theorem 12 to the result we want:

Theorem 13 (equivalent to Theorem 2) *Let $\delta, c > 0$ and $l > 1$, and let G be selected from $\mathcal{G}^j(n,p,k)$. Then, there exists K so that if $\mu = E[\mathrm{Num}_l(x,y)] \geq K \log n$, then:*

$$\mathbf{Pr}[(1-\delta)\mu \;\leq\; \mathrm{Num}_l(x,y) \;\leq\; (1+\delta)\mu] \;=\; 1 - o(n^{-c}).$$

Proof of Theorem 12: There are two random elements involved in $\mathcal{G}(n,p,k)$: first, each vertex is assigned a color at random, and then each "potential edge" is placed into the graph with probability p. For simplicity, we will separate these two events. In the following discussion, we will assume the colors have already been assigned to vertices, and that there are at least $\frac{n}{k}(1-o(1))$ vertices in each color class. This last condition occurs with probability exponentially close to 1 and so does not affect the statement of the theorem: the condition only affects the expectations by an exponentially small amount since the expectation given a particular assignment of colors is always $O(p^l n^{l-1})$ and is $\Theta(p^l n^{l-1})$ given the condition. Let $\mathcal{M}$ be the distribution given that such an assignment of vertices to colors has occurred, and let μ^* be the expectation of $\mathrm{Num}_l(x,y)$ given that $G \leftarrow \mathcal{M}$. So, $\mu^* = \mu(1+o(1))$.

Now, define a *potential path* to be a sequence of vertices $\langle x, v_1, v_2, \ldots, v_{l-1}, y\rangle$ with the property that each element in the sequence is in a different color class from the previous one. Also, let $X_1, \ldots, X_m$ be all the potential paths and let A_i be the event that $X_i \subseteq G$. For convenience, let $\mathbf{v} = l-1$ and $\mathbf{e} = l$ (the number of edges in the path); these are both constants. Note that the number of potential paths m is $\Theta(n^{\mathbf{v}})$ and for any given potential path X_i, $\mathbf{Pr}[A_i] = p^{\mathbf{e}}$.

The proof in [14] of Theorem 11 for the case where the expectation is between $K' \log n$ and $n^{\epsilon'}$ for sufficiently large K' and sufficiently small ϵ', proceeds in three stages. First, is shown a theorem stated here as Theorem 5. We have already proven the analog in Theorem 6. Second, is shown that for $G' \leftarrow \mathcal{G}(n,p)$, with probability $1 - o(n^{-c})$, the size of every maximal family $\mathcal{F}$ of *edge disjoint* X_i in G' is within $(1+\delta)$ of μ. Finally it is shown that for any fixed maximal family $\mathcal{F}$, with probability $1 - o(n^{-c})$ there are only a constant number of $X_j \notin \mathcal{F}$ in G' that intersect some path $X_i \in \mathcal{F}$ in an edge. Since every path must either belong to $\mathcal{F}$ or else intersect some $X_i \in \mathcal{F}$ (as $\mathcal{F}$ is a *maximal* family of disjoint paths), the last 2 parts of the argument imply that the number of paths of length l is within $(1+\delta)$ of the expectation with probability $1 - o(n^{-c})$.

Note that for X any subgraph of K_n at all,

$$\mathbf{Pr}[X \subseteq G \mid G \leftarrow \mathcal{M}] \;\; \leq \;\; \mathbf{Pr}[X \subseteq G' \mid G' \leftarrow \mathcal{G}(n,p)].$$

The reason is simply that each edge is placed into $G \leftarrow \mathcal{M}$ with probability *at most* p (either probability p or probability 0 depending on the colors of the endpoints), while in $\mathcal{G}(n,p)$ each edge is placed into the graph with probability exactly p. So, if we pick ϵ sufficiently small so that the $\mathcal{G}(n,p)$ argument holds, then the third part carries over directly and we need not prove it again here. We focus now on the second part. The analysis here is taken directly from the proof of [14].

Let us calculate some basic quantities. First, we can loosely upper bound the number of families $\mathcal{F} \subseteq \{X_1, \ldots, X_m\}$ of t pairwise edge-disjoint potential paths X_i, by $\binom{m}{t}$. Also, for any fixed such family $\mathcal{F}$, the probability that the X_i in $\mathcal{F}$ are all in G is $(p^{\mathbf{e}})^t$ since the $X_i \in \mathcal{F}$ are all disjoint so the corresponding events A_i are mutually independent.

For a given family $\mathcal{F}$ of t pairwise disjoint X_i, we now upper bound the probability that *no* path X_j that is edge-disjoint from all $X_i \in \mathcal{F}$ exists in G; that is, the probability that $\mathcal{F}$ is a maximal set given that all paths in $\mathcal{F}$ exist in G. Let $X_{i_1}, \ldots, X_{i_r}$ be all the potential paths disjoint from $\mathcal{F}$. By Theorem 6, for ϵ sufficiently small, $\sum_{i \sim j} \mathbf{Pr}[A_i \wedge A_j] = o(1)$ where $i \sim j$ if $i \neq j$ and X_i and X_j share an edge. So, certainly the summation restricted to just the $i, j \in \{i_1, \ldots, i_r\}$ equals $o(1)$ as well. Thus, by Janson's inequality, (noting that the "λ" term is $o(1)$) we have:

$$\begin{aligned} \mathbf{Pr}\Big[\bigwedge_{j=1}^{r} \bar{A}_{i_j}\Big] &= [1+o(1)] \prod_{j=1}^{r} \mathbf{Pr}\left[\bar{A}_{i_j}\right] \\ &= [1+o(1)](1-p^{\mathbf{e}})^r. \end{aligned}$$

Given the above facts, we can upper bound the probability P_t that there exists any maximal family $\mathcal{F}$ of t pairwise disjoint X_i's within G by the quantity:

$$\alpha_t \; = \; [1+o(1)]\tbinom{m}{t}(p^{\mathbf{e}})^t(1-p^{\mathbf{e}})^r.$$

Consider now two cases. First, suppose $t \geq n^{2\epsilon}$. We may upper bound $\binom{m}{t}$ by $\frac{m^t}{t!} \;\leq\; \left[\frac{3m}{t}\right]^t$. (Using $3 > 2.718...$ to avoid confusion with $\mathbf{e}$.) So, since $mp^{\mathbf{e}} = \mu^* = O(n^\epsilon)$ we have:

$$\alpha_t \;\; \leq \;\; \left[\frac{3m}{t}\right]^t (p^{\mathbf{e}})^t \;\; = \;\; O\Big(\frac{n^\epsilon}{n^{2\epsilon}}\Big)^{n^{2\epsilon}} \;\; = \;\; o(n^{-(c+1)}).$$

Thus, the probability there exists within G a maximal family $\mathcal{F}$ of *any* size $t \geq n^{2\epsilon}$ is at most $o(n^{-c})$.

The second case is $t \leq n^{2\epsilon}$. Here we assume that ϵ is sufficiently small (at most 1/4) so that $t \leq n^{1/2}$. We now lower-bound r (the number of paths that do not intersect $\mathcal{F}$) which we can do by upper-bounding the number of paths that *do* intersect $\mathcal{F}$. A path that intersects $\mathcal{F}$ must have one of its vertices besides x and y on the $t\mathbf{v}$ vertices used by paths in $\mathcal{F}$, so the total number of such paths is $O(t \cdot n^{\mathbf{v}-1})$ (using that $\mathbf{v}$ is a constant). Since $t \leq n^{1/2}$, this is at most

$O(n^{\mathbf{v}-1/2})$ and so $r \geq m - O(n^{\mathbf{v}-1/2})$. Thus,

$$
\begin{aligned}
(1-p^{\mathbf{e}})^r &\leq (1-p^{\mathbf{e}})^m/(1-p^{\mathbf{e}})^{O(n^{\mathbf{v}-1/2})} \\
&= (1-p^{\mathbf{e}})^m/(1-O(p^{\mathbf{e}}n^{\mathbf{v}-1/2})) \\
&\leq (1-p^{\mathbf{e}})^m/(1-O(n^{\epsilon}n^{-1/2})) \qquad (\text{since } \mu^* < n^{\epsilon}) \\
&= (1-p^{\mathbf{e}})^m[1+o(1)].
\end{aligned}
$$

So, we can upper bound the probability P_t by:

$$P_t \;\leq\; [1+o(1)]\binom{m}{t}(p^{\mathbf{e}})^t(1-p^{\mathbf{e}})^m \;\leq\; [1+o(1)]\binom{m}{t}(p^{\mathbf{e}})^t(1-p^{\mathbf{e}})^{m-t}.$$

Thus, $P_t \leq (1+o(1))\mathbf{Pr}[Y=t]$ where Y has the binomial distribution $B(m, p^{\mathbf{e}})$. We know for such a distribution, for any $\delta > 0$ we have $\mathbf{Pr}[|Y-\mu^*| > \delta\mu^*] = o(n^{-c})$, so long as $\mu^* > K\log n$ for sufficiently large K. Thus, the probability there exists any maximal family $\mathcal{F}$ of disjoint paths X_i of size *not* within $\delta\mu^*$ of μ^*, is also $o(n^{-c})$. This finishes the second part of the argument. Since as noted above, the third part follows immediately from the result for $\mathcal{G}(n,p)$, we have proved the theorem. ∎

Proof of Theorem 13: We may assume that the edge probability p is such that the expectation μ is at least $n^{\epsilon-1}$ for ϵ the constant of Theorem 12.

Write $p = Q\hat{p}$ where $\hat{p}$ is such that Theorem 12 works for edge probabilities $\hat{p}, 2\hat{p}, \ldots, l\hat{p}$ and $n^{\epsilon_1} < Q < n$. Now create G from $\mathcal{G}^j(n,p,k)$ in the usual way but then give each edge of G a label uniformly and independently from 1 to Q. Thus the edges with a particular label have distribution $\mathcal{G}^j(n,\hat{p},k)$. If $S \subseteq \{1,\ldots,Q\}$ then the graph of edges with labels in S has distribution $\mathcal{G}^j(n,\hat{p}|S|,k)$. Fixing x,y,l, let $N(S)$ denote the number of l-paths between x and y with labels in S. There are $O(n^l)$ sets S of size at most l so, selecting $c > l+2$ in Theorem 12, we have a.s. that

$$\left|\frac{N(S)}{E[N(S)]} - 1\right| < \delta$$

for all x, y and all S of size at most l.

Let's write T for $\mathrm{Num}_l(x,y)$ for notational convenience. How can we express T in terms of the $N(S)$? We would like to simply say that T is the sum of the $N(S)$ over all l-sets S but that would be multiply counting since those l-paths that have less than l labels would be counted for many S. To count T we use a generalization of Inclusion-Exclusion known as Mobius Inversion. Let N^i denote the sum of $N(S)$ over all i-element sets S. Then

$$T = \gamma_0 N^l + \gamma_1 N^{l-1} + \ldots + \gamma_{l-1}N^1$$

where the γ are given by the equations

$$1 = \gamma_0$$

$$1 = \gamma_0\binom{Q-(l-1)}{1} + \gamma_1$$

$$1 = \gamma_0 \binom{Q-(l-2)}{2} + \gamma_1 \binom{Q-(l-2)}{1} + \gamma_2$$

and, in general, for $t \leq l-1$,

$$1 = \sum_{s=0}^{t} \gamma_s \binom{Q-(l-t)}{t-s}$$

Why does this work? Suppose, for example, there is a path with precisely $l-2$ different labels. It will be counted $\binom{Q-(l-2)}{2}$ times by S of size l and $\binom{Q-(l-2)}{1}$ times by S of size $l-1$ and once by an S of size $l-2$ (i.e., itself) so that in the sum weighted by the γ it will be counted precisely once. And this holds for any number of different labels. One can show that (asymptotically in Q for fixed l)

$$\gamma_i \sim (-1)^i \frac{Q^i}{i!}.$$

We know a.s. that all of the $N(S)$ are within $1 \pm \delta$ of their expectation, and thus each N^i is within $1 \pm \delta$ of its expectation. There is a curve ball here in that T is a weighted signed sum of the N^i. Here we know that

$$E[T] = (n)_{l-1}(Q\hat{p})^l$$

and that

$$E[N^{l-i}] = \binom{Q}{l-i}(n)_{l-1}(\hat{p}i)^l$$

so:

$$|\gamma_i E[N^{l-i}]| \sim \frac{Q^i}{i!}\frac{Q^{l-i}}{(l-i)!}(n)_{l-1}\hat{p}^l i^l \sim E[T]\frac{i^l}{i!(l-i)!}.$$

With all the N^{l-i} within $1 \pm \delta$ of their expectations, $T = \sum \gamma_i N^{l-i}$ is within

$$\delta \sum_i \frac{i^l}{i!(l-i)!} + o(1)$$

of its expectation. The key point is not so much the exact expression but that these factors depend only on l so that for fixed l we can get T within $1 \pm \delta^*$ of its expectation by making δ appropriately low. ■

6 Conclusion and open problems

Focusing on the case of $k = 3$, our main results were as follows. In the balanced semi-random model, we described an algorithm that 3-colors $G \leftarrow \mathcal{G}_{SB}(n, p, 3)$ with high probability for $p \geq n^{-0.6+\epsilon}$. For the random model, we were able to provably 3-color for p as low as $n^{-1+\epsilon}$. One immediate open problem suggested by this work is: can one beat the $n^{-0.6}$ bound, or more ambitiously, 3-color for $p = n^{\epsilon-1}$ in the (balanced) semi-random model? More generally, is the semi-random model really harder than the random model for 3-coloring? In the random model, can one 3-color graphs from $\mathcal{G}(n, p, 3)$ with high probability for $p = n^{-1}\text{polylog}(n)$, or perhaps even for *all* values of p? The first problem ($p = n^{-1}\text{polylog}(n)$) may be possible by careful extensions of the path algorithm we have presented; for the second problem (all p) we really have no idea. Petford and Welsh [19] have recently experimented with heuristics on large random 3-colorable graphs, and these appear to find 3-colorings quickly for all but a quite small range of edge probabilities. While they do not have theoretical analyses of bounds for their heuristics, their results suggest that there is no intrinsic reason why the theoretical bounds for the random case could not be improved further.

Added in Print: Alon and Kahale [1], using an eigenvalue analysis, have recently been able to improve our Theorem 3, giving a polynomial time algorithm to find a 3-coloring when $p = \frac{(\log n)^c}{n}$. This answers one of the open questions mentioned above.

Acknowledgments: We would like to thank the anonymous referees for many helpful comments.

References

[1] N. Alon and N. Kahale. A spectral technique for coloring random 3-colorable graphs. Manuscript.

[2] C. Berge. *Graphs and Hypergraphs.* North-Holland, 1973.

[3] B. Berger and J. Rompel. A better performance guarantee for approximate graph coloring. *Algorithmica*, 1988.

[4] A. Blum. Some tools for approximate 3-coloring. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, St. Louis, October 1990.

[5] A. Blum. *Algorithms for Approximate Graph Coloring.* PhD thesis, Massachusetts Institute of Technology, May 1991. (MIT Laboratory for Computer Science Technical Report MIT/LCS/TR-506, June 1991).

[6] R. Boppana and J. Spencer. A useful elementary correlation inequality. *J. Combin. Theory Ser. A*, 50:305–307, 1989.

[7] P. Briggs, K. D. Cooper, K. Kennedy, and L. Torczon. Coloring heuristics for register allocation. In *Proceedings of the SIGPLAN '89 Conference on Programming Language Design and Implementation*, pages 275–284, Portland, June 1989.

[8] G. J. Chaitin, M. A. Auslander, A. K. Chandra, J. Cocke, M. E. Hopkins, and P. W. Markstein. Register allocation via coloring. *Computer Languages*, 6:47–57, 1981.

[9] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM J. Computing*, 17(2):230–261, April 1988.

[10] 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.

[11] L. Kucera. Expected behavior of graph colouring algorithms. In *Lecture Notes in Computer Science No. 56*, pages 447–451. Springer-Verlag, 1977.

[12] A. D. Petford and D. J. A. Welsh. A randomised 3-colouring algorithm. *Discrete Mathematics*, 74:253–261, 1989.

[13] M. Santha and U. V. Vazirani. Generating quasi-random sequences from semi-random sources. *JCSS*, 33:75–87, 1986.

[14] J. Spencer. Counting extensions. *J. Combin. Theory Ser. A*, 55:247–255, 1990.

[15] J. Spencer. Threshold functions for extension statements. *J. Combin. Theory Ser. A*, 53:286–305, 1990.

[16] J. S. Turner. Almost all k-colorable graphs are easy to color. *Journal of Algorithms*, 9:63–82, 1988.

[17] 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.

[18] 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.

[19] D. J. A. Welsh. Personal communication. 1991.

[20] A. Wigderson. Improving the performance guarantee for approximate graph coloring. *JACM*, 30(4):729–735, 1983.