Switching Colouring of $G(n, d/n)$ for Sampling up to Gibbs Uniqueness Threshold

Charilaos Efthymiou

Goethe University of Frankfurt, 60325, Germany
efthymiou@gmail.com

Abstract. Approximate random k-colouring of a graph $G = (V, E)$, efficiently, is a very well studied problem in computer science and statistical physics. It amounts to constructing, in polynomial time, a k-colouring of G which is distributed close to *Gibbs distribution*. Here, we deal with the problem when the underlying graph is an instance of Erdős-Rényi random graph $G(n, d/n)$, where d is fixed.

This paper improves on the approximate sampling colouring algorithm proposed in SODA 2012. We provide improved performance guarantees for this efficient algorithm, as we reduce the lower bound of the number of colours required by a factor of 1/2. In particular, we show the following statement for the accuracy of algorithm: For typical instances of $G(n, d/n)$ the algorithm outputs a k-colouring of $G(n, d/n)$ which is asymptotically uniform as long $k \geq (1 + \epsilon)d$. For the improvement we make an extensive use of the spatial correlation decay properties of the Gibbs distribution and the local treelike structure of the underlying graph.

1 Introduction

Approximate random k-colouring of a graph $G = (V, E)$, efficiently, is a very well studied problem in computer science and statistical physics. It amounts to constructing, in polynomial time, a k-colouring of G which is distributed close to *Gibbs distribution*, i.e. the uniform distribution over all the k-colourings of G. Here, we deal with the problem when the underlying graph is an instance of Erdős-Rényi random graph $G(n, p)$, where $p = d/n$ and d is fixed. We say that $G(n, p)$ has a property with high probability (*w.h.p.*) if the probability that the property holds tends to 1 as $n \to \infty$.

The problem of sampling colourings when the underlying graph is $G(n, d/n)$ is rather interesting due to *high degree effect*. That is, there is a relative large fluctuation on the degrees of the vertices in the random graph. E.g. it is elementary to show that typical instances of $G(n, d/n)$ have maximum degree $\Theta\left(\frac{\log n}{\log\log n}\right)$, while more than $1 - e^{-O(d)}$ fraction of the vertices have degree in the interval $(1 \pm \epsilon)d$. Usually the bounds for sampling k-colourings w.r.t. k are expressed it terms of the maximum degree e.g. [14,5]. However, for $G(n, d/n)$ the natural bounds for k should be in terms of the expected degree, rather than the maximum.

A. Schulz and D. Wagner (Eds.): ESA 2014, LNCS 8737, pp. 371–381, 2014.

The most powerful and most popular algorithms for this kind of problems are based on the Markov Chain Monte Carlo (MCMC) method. There the main technical challenge is to establish that the underlying Markov chain mixes in polynomial time (see [11]). The MCMC version of sampling colourings of $G(n, d/n)$ is a well studied problem [7,13,4]. The work in [7] shows that the well known Markov chain *Glauber block dynamics* for k-colourings has polynomial time mixing for typical instances of $G(n, d/n)$ as long as the number of colours $k \geq \frac{11}{2}d$. This is the lowest bound for k as far as MCMC sampling is concerned.

Recently, in [8], the author of this paper suggested a novel non MCMC approach for the approximate sampling colouring problem on $G(n, d/n)$. Roughly, the idea is as follows: Given the input graph, first we remove sufficiently many vertices such that the resulting graph has a "very simple" structure and it can be randomly coloured *efficiently*. Once we have a random colouring of this, simple, graph we start adding one by one all the edges we have removed in the first place. Each time we add a new edge we *update* the colouring so as the graph with the new edge remains (asymptotically) randomly coloured. Once the algorithm has rebuilt the initial graph it returns its colouring.

Let us be more specific on how we *update* the colouring once we add an extra edge. Assume that we are given the fixed graphs $G = (V, E)$ and $G' = (V, E')$ such that $E' = E \cup \{v, u\}$ for some $v, u \in V$. Given X, a random k-colouring of G, we want to create efficiently a random k-colouring of the slightly more complex graph G'. It is easy to show that if the vertices v, u take different colour assignments under X, then the colouring X is a random k-colouring of G'. The interesting case is when $X(v) = X(u)$. Then, the algorithm in [8] uses an operation called "switching" so as to alter the colouring of only one of the two vertices. E.g. using switching, from X which assigns v, u the same colour, we get Y a colouring which assigns v, u different colours while Y is very close to being random. Essentially, switching repermutes the colour classes of an appropriate subgraph of G that contains v[1]. Of course the "switching" can be implemented efficiently.

The approximate sampling algorithm in [8] w.h.p. over the input graph instances returns, in polynomial time, a k-colouring which is asymptotically random as long as $k \geq (2 + \epsilon)d$, for any fixed $\epsilon > 0$. Until this work this was the best bound for sampling colourings of $G(n, d/n)$ in terms of the minimum number of colours required. In this paper we improve on this bound even further by reducing it to $k \geq (1 + \epsilon)d$. For this improvement, we make an extensive use of the the spatial mixing properties of the Gibbs distribution and the local tree structure of the underlying graph.

The technical challenge for the analysis is to show that a random k-colouring does not specify large paths in $G(n, d/n)$ which are coloured with exactly two colours. In particular, we have to argue about the probability of a specific path in $G(n, d/n)$ to be 2-coloured, under the Gibbs distribution. This is a challenging task because of the highly complex structure that $G(n, d/n)$, typically, has. In [8], we deal with this problem by following a very pessimistic scenario about the

[1] In different contexts this component is called Kempe chain.

colouring around the path. I.e. we simplify the problem by making a worst case assumptions about the colouring of the vertices which are incident to the path.

In this work, we follow a more elaborate approach. That is, we consider the colouring of vertices at sufficiently large distance from the path and we show that their colouring does not affect (very much) the colouring of the path. This allows a more accurate estimation of the probability of a path being two coloured (See Section 4).

For presenting our main result we need to use the total variation distance as a measure of distance between distributions.

Definition 1. *For the distributions ν_a, ν_b on $[k]^V$, let $||\nu_a - \nu_b||$ denote their* total variation distance*, i.e.*

$$||\nu_a - \nu_b|| = \max_{\Omega' \subseteq [k]^V} |\nu_a(\Omega') - \nu_b(\Omega')|.$$

For $\Lambda \subseteq V$ let $||\nu_a - \nu_b||_\Lambda$ denote the total variation distance between the projections of ν_a and ν_b on $[k]^\Lambda$.

Theorem 1. *Let $\epsilon > 0$ be fixed and $k = (1+\epsilon)d$. Assume that the input of our algorithm is an instance of $G(n, d/n)$ and we let μ be the uniform distribution over its k-colourings. Also, we let μ' be the distribution of the colouring that is returned by the algorithm. With probability at least $1 - n^{-c}$ over the input instances $G(n, d/n)$ it holds that*

$$||\mu - \mu'|| = O\left(n^{-c}\right),$$

for sufficiently large $c = c(\epsilon) > 0$ and any fixed $d > d_0(\epsilon)$.

The proof of Theorem 1 appears in the full version of this paper in [6].

As far as the time complexity of the algorithm is regarded we provide Theorem 2, its proof appears in the full version of the paper in [6].

Theorem 2. *With probability at least $1 - n^{-2/3}$ over the input instances $G(n, d/n)$, the time complexity of the random colouring algorithm is $O(n^2)$.*

In this extended abstract, we are going to omit most of the technical details which are already known from [8]. In the full version of the paper in [6] we provide the proofs of all the results above. As a matter of fact we provide an improved and to a large extend simplified presentation of the results and their proofs that appeared in [8].

Structure of the paper. In Section 2 we provide a basic description of the algorithm. In Section 3 we give an overview of how the analysis works. In Section 4 we give a sketch of how do we use correlation decay to bound the probability of a path being two coloured.

Notation. We denote with small letters of the greek alphabet the colourings of a graph G, e.g. σ, η, τ, while we use capital letters for the random variables which take values over the colourings e.g. X, Y, Z. We denote with $\sigma_v, X(v)$ the colour

assignment of the vertex v under the colouring σ and X, respectively. Finally, for an integer $k > 0$ let $[k] = \{1, \ldots, k\}$.

2 Basic Description

First we need the notion of *switching* colouring. For this we define the *disagreement graph* [2]. Consider a fixed graph G and let v be a distinguished vertex in G. Let σ be a k-colouring of G and let some colour $q \neq \sigma(v)$. Under the colouring σ, we denote by $V_{\sigma(v)}, V_q$ the colour classes of $\sigma(v)$, and q, respectively. We call *disagreement graph* $Q_{\sigma(v),q}$ the maximal, connected, induced subgraph of G which includes v and vertices only from the set $V_{\sigma_v} \cup V_q$. In the colouring of Figure 1, the disagreement graph $Q_{B,G}$ is the one with the fat lines.

Definition 2 (Switching). *Consider G, v, σ and q as specified above. The "q-switching of σ" corresponds to the proper colouring of G which is derived by exchanging the assignments in the two colour classes in $Q_{\sigma_v,q}$.*

We would like to emphasize that the q-switching of any proper colouring of G is always a proper colouring too. Figure 2 illustrates a switching of the colouring that appears in Figure 1. Observe that the colouring in Figure 2 differs from the colouring in Figure 1 to that we have exchanged the two colour classes of the subgraph with the fat lines.

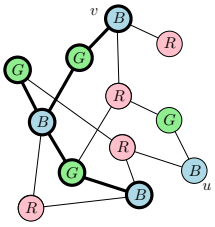

Fig. 1. "Disagreement graph"

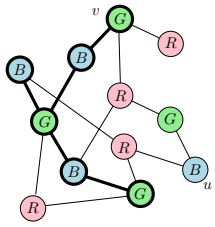

Fig. 2. "g-switching"

We assume that the input of the algorithm is an instance of $G(n, d/n)$ and k, the numbers of colours. The algorithm is as follows:

Set up: We construct a sequence $G_0, \ldots, G_r$ such that every G_i is a subgraph of $G(n, d/n)$. The graph G_r is identical to $G(n, d/n)$. Each G_i is derived by deleting $e_i = \{v_i, u_i\}$ from G_{i+1}, a random edge among those which do not belong to a *small cycle* of G_{i+1}. We call small, any cycle of length less than $(\log n)/(9 \log d)$. G_0 is the graph we get when there are no other edges to delete.

[2] What we call disagreement graph here, is also known as *Kempe chain* e.g. see [12].

With probability $1 - n^{-\Omega(1)}$, over the instances of $G(n, d/n)$, G_0 is simple enough and we can k-colour it randomly in polynomial time[3]. Assuming that we deal with such an instance, the algorithm works as follows:

Updates: Take a random colouring of G_0. Colour the rest of the graph according to the following inductive rule: Given that G_i is coloured Y_i, so as to get Y_{i+1}, the colouring of G_{i+1}, we distinguish two cases

Case a: In the colouring of the graph G_i the vertices v_i and u_i are assigned different colours (i.e. $Y_i(v_i) \neq Y(u_i)$).

Case b: In the colouring of the graph G_i the vertices v_i and u_i are assigned the same colour (i.e. $Y_i(v_i) = Y(u_i)$).

In the first case, we just set $Y_{i+1} = Y_i$, i.e. G_{i+1} gets the same colouring as G_i. In the second case, we choose q uniformly at random among all the colours but $Y_i(v_i)$. Then we set Y_{i+1} equal to the q-switching of Y_i. The q-switching is w.r.t. the vertex v_i.

With the above we conclude the description of the algorithm.

The reader may have observed that the switching does not necessarily provide a k-colouring where the assignments of v_i and u_i are different. That is, it may be that both vertices v_i, u_i belong to the disagreement graph. Then, after the q-switching of Y_i the colour assignments of v_i and u_i remain the same. We will show, that such a situation is rare as long as $k \geq (1 + \epsilon)d$. Typically, after the q-switching v_i, u_i get different colour assignments. The approximate nature of the algorithm amounts exactly to the fact that on some, rare, occasions the switching somehow fails[4].

3 The Setting for the Analysis of the Algorithm

In this section we provide the setting for analyzing the algorithm.

Definition 3 (Good & Bad Colourings). *Consider a graph G and let v, u be two distinguished vertices in this graph. Let σ be a proper k-colouring of G. We call σ a* bad colouring *w.r.t. the vertices v, u if $\sigma_v = \sigma_u$. Otherwise, we call σ* good.

The idea that underlies the algorithm, essentially, reduces the problem of sampling to dealing with the following problem.

Problem 1. Consider the graph G and two non adjacent vertices v, u. Given a *bad* random colouring of G w.r.t. v, u, turn it to a *good* random colouring, in polynomial time.

[3] The graph G_0, typically, contains connected components of two kinds. The first one is isolated vertices and the second one is simple cycles. Such graph can be randomly coloured trivially. Whether the algorithm is polynomial time or not depends exactly on whether G_0 can be coloured randomly efficiently. For more details see in the full version of this paper in [6].

[4] For $k \leq d$ our analysis cannot guarantee that these are fails sufficiently rare.

Consider two different colours $c, q \in [k]$ and let $\Omega_{c,c}$ and $\Omega_{q,c}$ be the set of colourings of G which assign the pair of vertices (v, u) colours (c, c) and (q, c), respectively. Essentially, Problem 1 asks to find a mapping $H_{c,q} : \Omega_{c,c} \to \Omega_{q,c}$, for every pair of different colours (c, q), such that the following two hold: (A) If Z is uniformly random in $\Omega_{c,c}$ then $H_{c,q}(Z)$ is uniformly random in $\Omega_{q,c}$. (B) The computation of $H_{c,q}(Z)$ can be accomplished in polynomial time.

For dealing with (A) an ideal, and to a great extent untrue, situation would have been if $\Omega_{c,c}$ and $\Omega_{q,c}$ admitted a bijection and $H_{c,q}$ was a bijection between the two sets. Since, this situation is not expected to hold in general, our approach is based on introducing an *approximate bijection* between $\Omega_{c,c}$ and $\Omega_{q,c}$. That is, we consider a mapping which is a bijection between two sufficiently large subsets of $\Omega_{c,c}$ and $\Omega_{q,c}$, respectively. Each of these two subsets will contain all but a vanishing fraction of the colourings of the original sets.

To be more specific, we assume that $H_{c,q}$ represents the operation of q-switching over the colourings in $\Omega_{c,c}$, as we described in Section 2. Then, there are sufficiently large sets $\Omega'_{c,c} \subseteq \Omega_{c,c}$ and $\Omega'_{q,c} \subseteq \Omega_{q,c}$ such that $H_{c,q}$ is a bijection between $\Omega'_{c,c}$ and $\Omega'_{q,c}$. In particular, we show the following: For Z which is distributed uniformly at random in $\Omega_{c,c}$, $H_{c,q}(Z)$ is distributed within total variation distance $\max\left\{\frac{\Omega_{c,c}\setminus\Omega'_{c,c}}{\Omega_{c,c}}, \frac{\Omega_{q,c}\setminus\Omega'_{q,c}}{\Omega_{q,c}}\right\}$ from the uniform distribution over $\Omega_{q,c}$. That is, the error we introduce depends on the relative size of the subset of colourings in $\Omega_{c,c}$ (resp. $\Omega_{q,c}$) for which $H_{c,q}$ fails to be a bijection. The colourings in $\Omega_{c,c}$ (resp. $\Omega_{q,c}$) that cannot be included in $\Omega'_{c,c}$ (resp. $\Omega'_{q,c}$) are called *pathological*.

We estimate the accuracy of the algorithm by providing upper bounds on the relative number of pathological colourings in $\Omega_{c,c}$ and $\Omega_{q,c}$, respectively. It turns out that the pathological colourings in $\Omega_{c,c}$ (resp. $\Omega_{q,c}$) are exactly these ones for which there is at least one path between v, u which is coloured only with c, q. Applying a q-switching to a pathological colouring in $\Omega_{c,c}$ we will get a new one which assigns both v, u the colour q, i.e. the switching fails. Also, it is direct to show that a pathological colouring in $\Omega_{q,c}$ cannot be generated by q-switching some colouring in $\Omega_{c,c}$.

3.1 Bounding the Error

The ratio $\frac{\Omega_{c,c}\setminus\Omega'_{c,c}}{\Omega_{c,c}}$, $\left(\text{resp. } \frac{\Omega_{q,c}\setminus\Omega'_{q,c}}{\Omega_{q,c}}\right)$ essentially expresses the probability of getting a pathological colouring if we choose uniformly at random from $\Omega_{c,c}$ (resp. $\Omega_{q,c}$).

Assume that we choose u.a.r. from $\Omega_{c,c}$. For every path P that connects v, u in the graph G, we let $I_{\{P\}}$ be the indicator variable which is one if the vertices in the path P are coloured only with colours c, q[5]. It is elementary to verify that

$$\frac{\Omega_{c,c}\setminus\Omega'_{c,c}}{\Omega_{c,c}} \leq \sum_{P} \Pr\left[I_{\{P\}} = 1\right].$$

Of course the same holds for $\Omega_{q,c}$.

[5] Observe that this is equivalent to having P in the disagreement graph $Q_{q,c}$.

In general, computing the probability $\Pr\left[I_{\{P\}} = 1\right]$ exactly is a formidable task. The challenging part in analyzing the performance of the algorithm is to bound this probability as precisely as possible. In [8] we used the idea of the so-called "Disagreement percolation" [3], illustrated in Figure 3. That is, we consider a path $P = (v, a, b, c, d, e, u)$. The vertices with the lines, in the figure, are exactly these vertices which are adjacent to the path. So as to bound the probability that P is coloured with c, q, we assume a worst case boundary colouring for the lined vertices and fixed colouring for v, u[6]. That is, given the fixed colourings we take a random colouring of the uncoloured vertices in the path and estimate the probability that P is coloured using only c, q. Observe that the exact probability is derived by considering an appropriate convex combination of boundary conditions for the lined vertices.

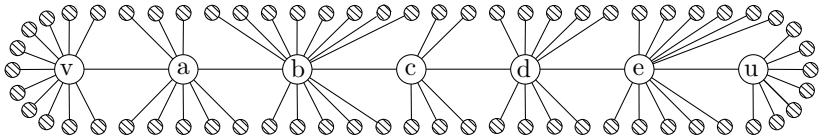

Fig. 3. Boundary at distance 1 from the path

Considering the worst case boundary condition around P is too pessimistic. Our improvement relies on dropping this assumption. The new approach is illustrated in Figure 4. Roughly speaking, we consider boundary conditions at the vertices around P which are at graph distance r, where r is a sufficiently large fixed number. We assume that the colouring of the vertices at the boundary is still a worst case one. However, now we can exploit spatial mixing properties of the Gibbs distributions. That is, the colouring of the distant vertices does not bias the colour of the vertices in P by too much. The weak dependence between the colourings of the vertices in the path and the boundary is achieved by choosing sufficiently large r and k. This allows to estimate more accurately the probability that the path P is 2-coloured.

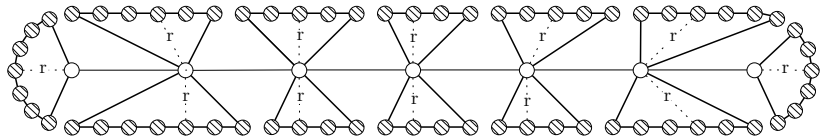

Fig. 4. Boundary at distance r from the path

[6] The vertices v, u are coloured c

4 Correlation Decay to Bound the Number of Bichromatic Paths

In this section we provide a sketch of how do we use correlation decay to show that the bichromatic paths between v_i and u_i are sufficiently rare. In what follows we let k, d, ϵ as in the statement of Theorem 1.

To be more precise, the main task is the following one: Consider an instance of $G(n, d/n)$ conditional that the path $P = (v_0, \ldots, v_l)$ appears in the graph, for some integer $l > 0$. Let X be a random k-colouring of the graph, and let $\mathcal{D}_P$ be the event that the path P is coloured only by the colours $c, q \in [k]$. Show that for any positive integer $l \leq \log^2 n$ it holds that

$$\Pr[\mathcal{D}_P] \leq \left(\frac{1}{(1+\epsilon/4)d}\right)^l. \tag{1}$$

For more details of how do we use the above bound in the analysis of the algorithm, see the full version in [6].

Observe that the probability term above is w.r.t. two levels of randomness. The underlying random graph and the random colouring X. To this end, we work as described in the following paragraphs.

Instead of revealing the whole graph, we restrict ourselves to revealing a small area around the path P as well as the edges and vertices between this area and the rest of the graph (outside the area). Let $\mathcal{N}$ denote this area around P[7] Also, let $\partial\mathcal{N}$ be the set of vertices outside $\mathcal{N}$ which are adjacent to some vertex in $\mathcal{N}$.

We consider the subgraph induced by $\mathcal{N}$ and $\partial\mathcal{N}$. We are going to consider a random k-colouring in this graph, conditional some worst case colouring at the vertices in $\partial\mathcal{N}$. That is, the colouring at $\partial\mathcal{N}$ maximizes the probability of the event $\mathcal{D}_P$.

Now, essentially, we have to deal with the randomness of $\mathcal{N}$ and the worst case colourings in $\partial\mathcal{N}$. However, for the vast majority of the vertices in P their neighbourhood (outside P) will be a tree of sufficiently large height and maximum degree at most $(1+\epsilon/3)d+1$. For such a good case of vertex v the boundary colouring will be at a relatively large distance away from v and, essentially, it won't affect its colouring too much. For the exceptional vertices which do not have such well behaved neighbourhood we are very pessimistic, i.e. we give the vertex on the path the appropriate colour for free. The bad cases are expected to be very rare. Somehow this setting reduces the randomness to considering how many good (resp. bad) behaved neighbourhood we have along the path.

In what follows we describe in detail how do we consider the subgraph around the path P. We use some integer parameter $r > 0$. Around each vertex $v_i \in P$ we are going to reveal a subgraph of maximum radius r.

Area Around Path P. For each vertex $v_i \in P$ we need to define the sets of vertices $\mathcal{L}_s(v_i)$, for integer s such that $0 \leq s \leq r$. By definition, we set $\mathcal{L}_0(v_i) = \{v_i\}$. Also, we let $\mathcal{N}_r(v_i)$ be the induced subgraph of $G(n, d/n)$ which contains the vertices $\bigcup_{s=0}^{r} \mathcal{L}_s(v_i)$.

[7] $\mathcal{N}$ is a subgraph which also contains P.

We are going to describe how do we get each $\mathcal{L}_s(v_i)$, inductively. Given $\mathcal{L}_s(v_i)$ we get $\mathcal{L}_{s+1}(v_i)$ by working as follows: Let $\mathcal{R}_{i,s}$ be the set that contains all the vertices of $G(n, d/n)$ but those which belong in the path P, those which belong in $\bigcup_{j\leq s} \mathcal{L}_j(v_i)$ and those which belong in $\mathcal{N}_r(v_j)$, for $j < i$. The set $\mathcal{L}_{s+1}(v_i)$ contains (possibly all) neighbours that the vertices in $\mathcal{L}_s(v_i)$ have in $\mathcal{R}_{i,s}$.

Consider a specific (arbitrary) ordering of the vertices in $\mathcal{R}_{i,s}$. For each vertex $u \in \mathcal{L}_s(v_i)$ we examine adjacency with the vertices in $\mathcal{R}_{i,s}$ in the predefined order. We stop revealing once we either have revealed $(1+\epsilon/3)d+1$ neighbours of u into $\mathcal{R}_{i,s}$ or if we have checked all the possible adjacencies of u with $\mathcal{R}_{i,s}$ (whatever happens first). In both cases, the number of neighbours of u we reveal is at most $(1+\epsilon/3)d+1$.

Once we have the sets $\mathcal{L}_s(v_i)$, for $0 \leq s \leq r$, it is direct to get the subgraph $\mathcal{N}_r(v_i)$. Ideally we would like each of these $\mathcal{N}_r(v_i)$ to be a tree of sufficiently small maximum degree[8]. Also, have would like these $\mathcal{N}_r(v_i)$s to not intersect with each other. If any of these conditions is not true for some $\mathcal{N}_r(v_i)$, then $\mathcal{N}_r(v_i)$ is considered `Fail`. To be more specific, once we the subgraphs $\mathcal{N}_r(v_i)$, for each $0 \leq i \leq l$ we let the following: $\mathcal{N}_r(v_i)$ is `Fail` if at least one of the following happens:

- The maximum degree in $\mathcal{N}_r(v_i)$ is equal to $(1+\epsilon/3)d+2$.
- The graph $\mathcal{N}_r(v_i)$ is not a tree.
- There is at least one integer $j \neq i$ such that some vertex $w_0 \in \mathcal{N}_r(v_j)$ is adjacent to some vertex $w_1 \in \mathcal{N}_r(v_i)$, unless w_0, w_1 are consecutive vertices in the path P.

Having specified $\mathcal{N}_r(v_i)$ for every $v_i \in P$, the sets $\mathcal{N}$ and $\partial\mathcal{N}$ are defined as follows: The set $\partial\mathcal{N}$ contains the vertices $v_i \in P$ for which $\mathcal{N}_r(v_i)$ is `Fail` and the vertices at distance exactly r from v_i for which $\mathcal{N}_r(v_i)$ is not `Fail`. The set $\mathcal{N}$ includes the vertices of the sets $\mathcal{N}_r(v_i)$ which are not `Fail` and do not belong to $\partial\mathcal{N}$.

A vertex v_i in the path is called disagreeing if the following holds: For i even, the color of v_i is c. For i odd, the colour is q . Let the event D_i that "v_i is disagreeing". Clearly it holds that

$$\Pr\left[D_P\right] \leq \Pr\left[\cap_{i=1}^{l} D_i\right]. \tag{2}$$

The proposition will follow by bounding appropriately $\Pr\left[\cap_{i=1}^{l} D_i\right]$.

Let the event A_i, B_i, C_i be defined as follows: A_i = "$\mathcal{N}_r(x_i)$ is `Fail`". B_i = "$\mathcal{N}_r(x_i)$ is not `Fail` and x_i is disagreeing". Finally, $C_i = A_i \cup B_i$. In the full version of this work in [6] we get the following inequality

$$\Pr\left[\cap_{i=1}^{l} D_i\right] \leq \Pr\left[\cap_{i=1}^{l} C_i\right]. \tag{3}$$

The above inequality, which is easy to prove, somehow, follows from the discussion at the beginning of this section. From (2) and (3) we get that

$$\Pr\left[\mathcal{D}_P\right] \leq \prod_{i=1}^{l} \Pr\left[C_i | \cap_{j=1}^{i-1} C_j\right] \leq \prod_{i=1}^{l} \left(\Pr\left[A_i | \cap_{j=1}^{i-1} C_j\right] + \Pr\left[B_i | \cap_{j=1}^{i-1} C_j\right]\right), \tag{4}$$

[8] Maximum degree at most $(1+\epsilon/3)d+1$

where the second inequality follows from a simple the union bound. Then, (1) follows by bounding appropriately the rightmost part in (4).

For bounding the terms $\Pr\left[A_i|\cap_{j=1}^{i-1} C_j\right]$ we use the following lemma.

Lemma 1. *Consider a sufficiently large fixed integer $r > 0$, independent of d. Consider the set $\mathcal{N}_r(v_j)$, for every $v_j \in P$. For sufficiently large d it holds that*

$$\Pr\left[\mathcal{N}_r(v_i) \text{ is } \texttt{Fail} \,|\mathcal{N}_r(v_j) \text{ for } j \neq i\right] \leq \exp\left(-\epsilon^2 d/35\right).$$

That is, it holds that

$$\Pr\left[A_i|\cap_{j=1}^{i-1} C_j\right] \leq \exp\left(-\epsilon^2 d/35\right). \tag{5}$$

For bounding the terms $\Pr\left[B_i|\cap_{j=1}^{i-1} C_j\right]$ we use the following spatial mixing result.

Proposition 1. *Let k, ϵ and d as in Theorem 1. Consider a path $P = (v_0, \ldots, v_l)$ in $G(n, d/n)$ such that $\mathcal{N}_r(v_i)$ is not* `Fail`*. Let X be a random colouring of $G(n, d/n)$. Let M denote the set of vertices outside $\mathcal{N}_r(v_i)$. For any proper colouring σ of $G(n, d/n)$ and any colour $c \in [k]\backslash\{\sigma_{v_{i-1}}, \sigma_{v_{i+1}}\}$ it holds that*

$$\left|\Pr[X(v_i) = c|X(M) = \sigma_M] - \frac{1}{k - t_\sigma}\right| \leq \frac{f_{\epsilon,r}}{k - t_\sigma},$$

where for a fixed $\epsilon > 0$, $f_{\epsilon,r} > 0$ is a decreasing function of r. Also, t_σ is number of different colours that σ_M uses for colouring v_{i-1} and v_{i+1}.

Using the proposition above, we get that

$$\Pr\left[B_i|\cap_{j=1}^{i-1} C_j\right] \leq \frac{1}{k-2} + \frac{f_{\epsilon,r}}{k-2},$$

where $f_{\epsilon,r}$ is defined in the statement of Proposition 1. Taking sufficiently large d and $r = r(\epsilon)$ we bound appropriately the rightmost part of (4).

Acknowledgement. The author would like to thank the anonymous reviewers for their very useful comments and remarks.

References

1. Achlioptas, D., Coja-Oghlan, A.: Algorithmic Barriers from Phase Transitions. In: Proc. of FOCS 2008, pp. 793–802 (2008)
2. Aldous, D.: Random walks of finite groups and rapidly mixing Markov chains. In: Séminaire de Probabilités XVII 1981/82, pp. 243–297. Springer, Heidelberg (1983)
3. van den Berg, J., Maes, C.: Disagreement percolation in the study of Markov fields. Annals of Probability 22, 749–763 (1994)
4. Dyer, M., Flaxman, A., Frieze, A.M., Vigoda, E.: Random colouring sparse random graphs with fewer colours than the maximum degree. Journal R.S.A. 29, 450–465 (2006)

5. Dyer, M., Frieze, A.M., Hayes, A., Vigoda, E.: Randomly colouring constant degree graphs. In: Proc. of 45th FOCS, pp. 582–589 (2004)
6. Efthymiou, C.: Switching colouring of G(n,d/n) for sampling up to Gibbs Uniqueness Threshold. Technical Report on arxiv.org
7. Efthymiou, C.: MCMC sampling colourings and independent sets of G(n, d/n) near uniqueness threshold. In: Proc. of SODA 2014, pp. 305–316 (2014)
8. Efthymiou, C.: A simple algorithm for random colouring $G(n, d/n)$ using $(2 + \epsilon)d$ colours. In: Proc. of SODA 2012, pp. 272–280 (2012)
9. Jonasson, J.: Uniqueness of Uniform Random Colorings of Regular Trees. Statistics & Probability Letters 57, 243–248 (2001)
10. Janson, S., Luczak, T., Ruciński, A.: Random graphs. Wiley and Sons, Inc. (2000)
11. Jerrum, M., Sinclair, A.: The Markov chain Monte Carlo method:an approach to approximate counting and integration. In: Dorit, H. (ed.) Approximation Algorithms for NP-Hard Problems. PWS (1996)
12. Molloy, M.: The freezing threshold for k-colourings of a random graph. In: Proc. of the 44th ACM Symposium on Theory of Computing (STOC 2012), pp. 921–930 (2012)
13. Mossel, E., Sly, A.: Gibbs Rapidly Samples Colorings of $G_{n,d/n}$. Journal Probability Theory and Related Fields 148(1-2) (2010)
14. Vigoda, E.: Improved bounds for sampling colorings. Journal of Mathematical Physics 41(3), 1555–1569 (2000); A preliminary version appears in FOCS 1999