THEORY OF COMPUTING, Volume 8 (2012), pp. 567–595
www.theoryofcomputing.org

SPECIAL ISSUE IN HONOR OF RAJEEV MOTWANI

Online Graph Edge-Coloring in the Random-Order Arrival Model*

Bahman Bahmani† Aranyak Mehta Rajeev Motwani

Received: July 31, 2010; published: December 9, 2012.

Abstract: A classic theorem by Vizing asserts that if the maximum degree of a graph is Δ, then it is possible to color its edges, in polynomial time, using at most $\Delta+1$ colors. However, this algorithm is offline, i. e., it assumes the whole graph is known in advance. A natural question then is how well we can do in the online setting, where the edges of the graph are revealed one by one, and we need to color each edge as soon as it is added to the graph. Online edge coloring has an important application in fast switch scheduling. A natural model is that edges arrive online, but in a random permutation. Even in the random permutation model, the best proven approximation factor for any algorithm is the factor 2 of the simple greedy algorithm (which holds even in the worst-case online model). The algorithm of Aggarwal et al. (FOCS'03) provides a 1+o(1) factor algorithm for the case of very dense multi-graphs, when $\Delta = \omega(n^2)$, where n is the number of vertices. In this paper, we show that for graphs with $\Delta = \omega(\log n)$, it is possible to color the graph with $\big(1+\mathrm{e}/(\mathrm{e}^2-1)+o(1)\big)\,\Delta \le 1.43\Delta$ colors, with high probability, in the online random-order model. Our algorithm is inspired by a 1.6-approximate distributed offline algorithm of Panconesi and Srinivasan (PODC'92), which we extend by reusing failed colors online.

*A preliminary version of this paper appeared in the Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2010).

†Supported by a Stanford Graduate Fellowship.

ACM Classification: F.2.2, G.2.2

AMS Classification: 05C85, 68W27, 68R10

Key words and phrases: graph edge coloring, online algorithm, random order

DOI: 10.4086/toc.2012.v008a025

Further, we show how we can extend the algorithm to reuse colors multiple times, which reduces the approximation factor below 1.43. We conjecture that the algorithm becomes nearly optimal (i. e., uses $\Delta + o(\Delta)$ colors) with $O(\log(\Delta/\log n))$ reuses. We reduce the question to proving the non-negativity of a certain recursively defined sequence, which looks true in computer simulations. This non-negativity can be proved explicitly for a small number of reuses, giving improved algorithms: e. g., the algorithm which reuses colors 5 times uses 1.26Δ colors.

1 Introduction

An edge coloring of a graph is a coloring of its edges so that no two edges incident on each other get the same color. If the maximum degree of a graph is Δ, then obviously, every edge coloring needs at least Δ colors. A classic theorem by Vizing [9] proves that it is possible to edge-color a graph using at most $\Delta + 1$ colors. However, determining whether the required number of colors is Δ or $\Delta + 1$ is known to be NP-complete for general graphs [7]. The proof of Vizing's theorem is constructive and actually gives a polynomial time algorithm to find an edge coloring using at most $\Delta + 1$ colors. For bipartite graphs there are fast algorithms to edge color using Δ colors [3]. However, all these algorithms are offline, i. e., they assume that the graph is known in advance. A natural question then is how well we can do in the online setting, where the edges of the graph are revealed one by one, and we need to color each edge as soon as it is added to the graph.

We study the online edge coloring problem for bipartite graphs, in the model in which edges arrive in a random permutation.[1] We also assume that the graph is regular (otherwise, one can add dummy edges to the graph to make it regular [1]). We note that the random order arrival model can simply be considered as an algorithmic technique for fast offline approximation: randomly permute the edges and run a (simple and local) online algorithm.

1.1 Problem definition

Let $G = (B, T, E)$ be a regular bipartite graph with degree Δ. Throughout, we will call the vertices in B as bottom vertices, and the vertices in T as top vertices. The vertices are known in advance, while the edges E are unknown. Edges arrive online in a random permutation of E. We have to color each edge as soon as it arrives, so as to get a valid edge coloring at the end of the algorithm. The objective is to do this using the smallest number of colors possible.

1.2 Prior work and our results

Prior work The simple greedy algorithm (Greedy), which colors each edge with the smallest color (in some fixed but arbitrary numbering of colors) not already used by a neighboring edge, will color the graph with no more than $2\Delta - 1$ colors. This is true in the worst-case online model, with adversarial input order on E. But in fact, this is the best known analysis for Greedy even in the random-order arrival model.

[1] Note that, as shown in [8], the edge coloring problem for non-bipartite graphs can be reduced to the problem for bipartite graphs, by considering a random balanced cut in the graph, coloring the edges of the cut, and then recursing on the two sides of the cut. Note that this can be done in the online setting as well.

	Δ	# colors used	Arrival order
Greedy Algorithm	no constraint	$\leq 2\Delta-1$	Adversarial
Lower Bound [2]	$O\left(\sqrt{\log n}\right)$	$\geq 2\Delta-1$	Adversarial
Algorithm [1]	$\omega(n^2)$	$\Delta+o(\Delta)$	Random
Here (Extension to 5 rounds)	$\omega(\log n)$	1.26Δ	Random

Figure 1: A summary of known upper and lower bounds in the Adversarial and Random-Order models

Bar-Noy et al. [2] prove that Greedy is optimal for both deterministic and randomized algorithms in the worst-case order model. However, the examples they construct to prove the lower bounds are very sparse, i. e., $\Delta = O(\log n)$ for deterministic algorithms, and $\Delta = O(\sqrt{\log n})$ for randomized algorithms. Then, the natural question is can we do better than Greedy if the graph is "dense."

Aggarwal et al. [1] gave an online algorithm which colors a bipartite graph using $\Delta+o(\Delta)$ colors in the random-order arrival model, when $\Delta = \omega(n^2)$. Thus, this achieves essentially optimal performance, but in an extremely dense *multigraph*. The motivation in [1] to study bipartite edge coloring in the random-order model was an application in high-speed switch scheduling. We point to [1] and the references therein for details, and provide only a brief outline of the application here. The input and output ports of a switch correspond to the two sides of a bipartite graph. At every time step, at most one request arrives at each input port, destined to some output port. Also, at each time step the switch can route a matching across from input to output ports: at most one packet leaving each input port and at most one arriving at each output port. A fast edge coloring algorithm which uses $\Delta+o(\Delta)$ colors for a graph of degree Δ can be used to schedule the switch as follows: for some Δ number of time steps, color each incoming request as an edge in the graph. Then spend the next $\Delta+o(\Delta)$ time steps routing the matchings corresponding to each color class from the edge-coloring, and at the same time coloring the newly incoming requests as the edges of a new graph. In this application, an algorithm which works with the density bound of $\omega(n^2)$ suffices, while causing a delay of $\omega(n^2)$ per packet. An algorithm with a smaller density bound will also work, and will reduce the wait time of each request, provided it uses $\Delta+o(\Delta)$ colors. We further discuss the connection of our algorithm to the one in [1] in Section 4.

Thus, no algorithm is known to perform better than Greedy's factor 2 (using fewer than $2\Delta-1$ colors) even in the random-order arrival online model, when the graph is denser than $\log n$, but sparser than n^2.

Our results In this paper, we provide an online algorithm which uses

$$\left(1+\frac{\mathrm{e}}{\mathrm{e}^2-1}\right)\Delta+o(\Delta) < 1.43\Delta$$

colors, with high probability, for graphs with $\Delta = \omega(\log n)$. We briefly describe the algorithm here (a detailed description is provided in Section 2): The algorithm has a number of palettes P^i, each with a different number of colors. It partitions the incoming edges into two types, "Early" and "Late," depending on the arrival time of the edge. For an early edge (b,t), the algorithm tries to color it with a random color from P^1 which b has not tried before. If it fails (because some previously arrived edge incident on t has used that color already), then it tries to color it with a random color from P^2 which b has not tried before, and so on, until success. After all the early edges have arrived, a subset $\mathrm{R}^i(b)$ of colors from P^i have failed to be used by each bottom vertex b. The algorithm augments this set by injecting a set of new colors N^i so that we have a sufficient number of colors. Then, for a late edge (b,t), the algorithm tries to color it using a random color chosen from $\mathrm{R}^1(b) \cup \mathrm{N}^1$ which b has not tried before for a late edge. If it fails to do so, it will try to color the edge with a random color from $\mathrm{R}^2(b) \cup \mathrm{N}^2$ (not tried by b for a late edge), and so on. Thus, the main idea in the algorithm is to *reuse* failed colors: each color from the palettes P^i gets a second chance (at each bottom vertex) before it is discarded.

Analysis techniques One of the main difficulties in the analysis of the algorithm lies in the correlations between the sets of reusable colors at bottom and top vertices when we process late edges. For example, it could be that bottom vertices can only use precisely those colors for late edges which the incident top vertices have already used up to color some early edges. In this pessimistic case, when the sets of available colors at bottom and top vertices are *disjoint*, we would not be able to reuse any colors and would get a factor of $\mathrm{e}/(\mathrm{e}-1) \simeq 1.59$. In the optimistic case, these sets of reusable colors are *identical* for all vertices, and the analysis would proceed to give a factor of $\mathrm{e}^2/(\mathrm{e}^2-1) \simeq 1.16$. If the sets were *independent*, we would get a factor of $(\mathrm{e}^2+\mathrm{e}-1)/(\mathrm{e}^2-1) \simeq 1.43$. What we prove is that these sets are *positively correlated* for bottom and top vertices, which still leads to the same 1.43 factor. Figure 2 provides an example to describe the importance of the correlation between reusable colors at bottom and top vertices in Round 2. In our analysis, we do not need to make any assumptions on the correlations between available colors at different bottom vertices.

A related issue is that, due to the non-independence of the reusable color sets, late edges can have unequal probabilities of succeeding to color themselves from different palettes (as opposed to early edges, where the probability of success depends only on the position in the random permutation). For example, due to the structure of the graph, some vertices may be "lucky" in the sense that their late edges succeed in coloring themselves from the first few palettes. While this is a good event as such, it leads to an uneven and unwieldy analysis. We rectify this by smoothing out success probabilities at every edge by *artificially rejecting* edges which succeed more than required, by flipping a coin with an appropriate bias. We provide an example in Section 2.3 (Figure 4 in Remark 2.11) to illustrate the reason probabilities of success can be different for different round 2 edges, even if they are incident on the same top vertex.

Extension We may hope to use a smaller number of colors by extending the algorithm to allow bottom nodes to possibly try each color more than two times. In Section 3, we show how we can extend the algorithm by dividing the edges to multiple rounds (instead of just two rounds, early and late). We show that by doing so we can indeed improve the approximation factor, and we conjecture that with $O(\log(\Delta/\log n))$ rounds, the algorithm uses a near-optimal $\Delta + o(\Delta)$ colors. We reduce this question to proving that a certain recursively defined numeric sequence is non-negative. This seems to be true from computer simulations. The non-negativity question can be resolved explicitly for small number of rounds, which gives, e. g., a 5-round algorithm using 1.26Δ colors.

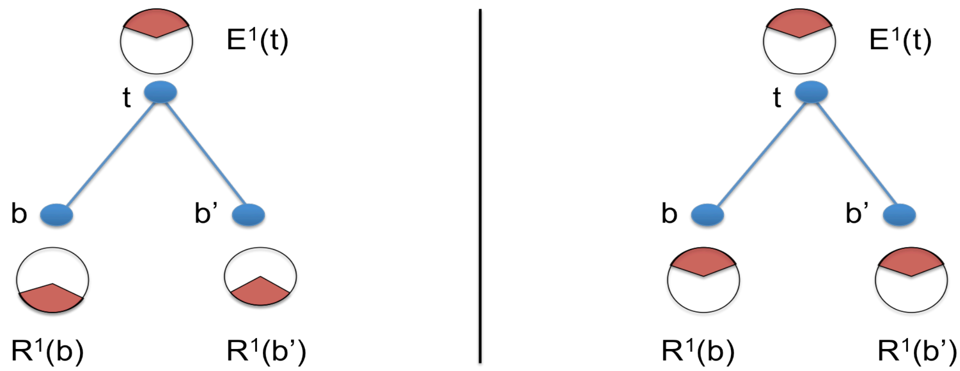

Figure 2: This example illustrates how the probability of successful coloring of round 2 edges depends on the correlation between the set of colors available at top and bottom vertices. $E^1(t)$ is the set of colors available at top vertex t, and $\mathrm{R}^1(b)$, $\mathrm{R}^1(b')$ are the sets available at bottom vertices b, b'. In the example on the left, since $\mathrm{R}^1(b) \cup E^1(t) = \emptyset$ and $\mathrm{R}^1(b') \cup E^1(t) = \emptyset$, the round 2 edges (b,t) and (b',t) have zero chance each of being colored successfully. In the example on the right, $\mathrm{R}^1(b) = \mathrm{R}^1(b') = E^1(t)$, so both the edges have a (constant) positive probability of success. (For simplicity, we have ignored the existence of the new palette N^1 in round 2 in this example.)

1.3 Previous related results on distributed algorithms

There is a related sequence of results in the literature on distributed offline algorithms for edge coloring. The first such algorithm with a factor less than 2 was provided by Panconesi and Srinivasan [8] (we will refer to the algorithm as PS), which uses $(\mathrm{e}/(\mathrm{e}-1))\Delta \simeq 1.59\Delta$ colors. Since our algorithm is inspired by PS, we describe it at some length here. PS runs in phases, where each phase has its own palette of colors. In each phase, each bottom vertex proposes colors for all its incident edges by taking a random permutation of the colors in the palette for that phase. Thus there are no color conflicts of proposed colors at bottom vertices. In the same phase, each top vertex accepts, for each color, exactly one incident edge chosen uniformly at random among those which propose that color. If an edge gets its proposed color accepted, then its color is fixed. Otherwise, it proceeds to the next phase. This propose-accept process guarantees that there are no color conflicts at any vertex. In each phase, some fraction of each vertex's incident edges get colored. It is proved that, with high probability, the vertex degrees reduce at a rate of $1/\mathrm{e}$, giving a total number of $1/(1-1/\mathrm{e}) \approx 1.58\Delta$ colors.

Our online algorithm is inspired by PS. Firstly, we show how to convert the idea behind PS to work in the online random-order setting. Secondly (and this is our main algorithmic contribution) we introduce the idea of *reusing* colors of a palette which a bottom vertex failed to use, in a next *round* (for edges which arrive later in the online order). So, at a high level (as will become clear in Section 2), our first round implements PS online for edges which arrive early, and the second round reuses the failed colors for late edges. We note that Aggarwal et al. [1] mention in their introduction that PS can be made online to use $1/(1-1/\mathrm{e})$ colors, although no details are given (presumably they meant a transformation similar to ours). Subsequent to [8], distributed algorithms were found [4, 6] using a near-optimal $\Delta + o(\Delta)$ number of colors (for $\Delta = \Omega(\text{polylog}(n))$). However, it is unclear how to make those algorithms work in the online model. As a related reference, we note the book [5], which includes a clear exposition of the concentration of measure techniques used in the analysis of these distributed algorithms.

2 The two round algorithm

The online algorithm is defined in Figure 3. Let $r = \mathrm{e}/(\mathrm{e}+1)$. We partition the input edges into two types: edges that arrive before time $rn\Delta$ are called Round 1 (or Early) edges, and edges which arrive later than time $rn\Delta$ are called Round 2 (or Late) edges. The algorithm has a collection of $L = O(\log(\Delta/\log n))$ main palettes P^1, P^2, P^3, $\ldots$, P^L, as well as L augmenting palettes N^1, N^2, $\mathrm{N}^3, \ldots, \mathrm{N}^L$, each with a distinct set of colors. Palette P^i has size $\Delta^{(i)}$ where $\Delta^{(i)}$ is recursively defined by: $\Delta^{(1)} = r\Delta - o(\Delta)$, and $\Delta^{(i+1)} = \Delta^{(i)}/\mathrm{e} - o(\Delta^{(i)})$. The size of the palette N^i will be determined later. The algorithm also has a special palette, P^∞, with $o(\Delta)$ colors.

Round 1 edges are treated as follows: For each bottom vertex b and each $i \in [1, L]$, we maintain a set $\mathrm{Tried}^i(b) \subseteq \mathrm{P}^i$, which is the set of colors from P^i that b has already proposed for any of its incident edges. When a round 1 edge (b,t) arrives, we try to color it using a random color c^* from $\mathrm{P}^1 \setminus \mathrm{Tried}^1$. If no previously arrived edge (b',t) was already colored c^*, then we succeed in coloring (b,t) with c^*. Otherwise, we fail to color (b,t) from P^1 and we try from $\mathrm{P}^2 \setminus \mathrm{Tried}^2$, and so on. If we fail to color (b,t) from all palettes $\mathrm{P}^1, \ldots, \mathrm{P}^L$, then we greedily color (b,t) using P^∞.[2] For each b and each $i \in [1, L]$, we also maintain $\mathrm{R}^i(b) \subseteq \mathrm{P}^i$, which is the set of colors from P^i which b tried to use to color some Round 1 edge (b,t), but failed because some previously arrived edge (b',t) had already taken that color.

Round 2 edges are treated in a similar manner, but with two differences:

(1) For every bottom vertex b, and for every $i = 1, \ldots, L$, we replace P^i with $\mathrm{R}^i(b) \bigcup \mathrm{N}^i$. Thus each bottom vertex *reuses* the colors $\mathrm{R}^i(b)$ it failed to use in Round 1. The palette N^i is added to provide enough additional colors to have a proposal for every round 2 edge.

(2) A top vertex may reject some proposals, even if there is no previously arrived incident edge which has used that color. This is done randomly by flipping a coin of an appropriate bias, and is done so that the probability that an edge succeeds in coloring itself from $\mathrm{R}^i(\cdot) \cup \mathrm{N}^i$ depends only on the position within the permutation and not on the identity of the edge. We describe the motivation and the details of this artificial rejection, including the definitions of $q^i(h)$ and $p^i(t,h)$ (as used in Figure 3), in detail in Section 2.3.

It is easy to see that the algorithm produces a valid coloring, i. e., it never uses the same color for two edges incident on each other: (1) bottom vertices propose colors for their edges by sampling without replacement, so there is no color conflict at bottom vertices, (2) we accept a proposed color only if no previously arrived edge incident on the same top vertex has already been colored that color. We will prove that, by choosing the right values for $|\mathrm{N}^i|$, with high probability, the algorithm does not abort in

[2] As a minor point, we also color (b,t) from P^∞ if it fails to get colored from $\mathrm{P}^1, \ldots, \mathrm{P}^i$, but there are already at least $\Delta^{(i+1)}$ edges incident on t which have failed to get colored from $\mathrm{P}^1, \ldots, \mathrm{P}^i$ (for some $i \in [1, L-1]$). This is done to just make some of the later analysis a bit cleaner.

Round 1 (Early edges):
For all $i \in [1,L]$, bottom nodes b, and top nodes t, initialize $\mathrm{Tried}^i(b) = \mathrm{R}^i(b) = \emptyset$, and $\overline{deg^i_1}(t) = 0$.

For $s \in [1, rn\Delta]$, when the s^{th} edge $e = (b,t)$ arrives in the online order:

- Set $i = 0$.
- While (e is not colored and $i < L$):
  - $i++$.
  - $\overline{\deg^i_1}(t)++$.
  - If $\left(\overline{\deg^i_1}(t) > \Delta^{(i)}\right)$:
    - Color e greedily from P^∞; If that is not possible, then abort.
  - Else:
    - Pick a color c^* uniformly at random from $\mathrm{P}^i \setminus \mathrm{Tried}^i(b)$. Set $\mathrm{Tried}^i(b) = \mathrm{Tried}^i(b) \cup \{c^*\}$. (If no such color exists then continue).
    - If no previously arrived edge incident on t was colored c^*, then color e with c^*; Else set $\mathrm{R}^i(b) = \mathrm{R}^i(b) \cup \{c^*\}$.
- If e is not yet colored, color it greedily from P^∞. If no such color is available in P^∞, then abort.

Round 2 (Late edges):
For all bottom vertices b and all $i \in [1,L]$, initialize $\mathrm{Tried}^i(b) = \emptyset$.

For $s \in [rn\Delta + 1, n\Delta]$, when the s^{th} edge $e = (b,t)$ arrives in the online order:

- Set $i = 0$.
- While (e is not colored and $i < L$):
  - $i++$.
  - Pick a color c^* uniformly at random from $\left(\mathrm{R}^i(b) \cup \mathrm{N}^i\right) \setminus \mathrm{Tried}^i(b)$. Set $\mathrm{Tried}^i(b) = \mathrm{Tried}^i(b) \cup \{c^*\}$. (If no such color exists then continue).
  - If no previously arrived edge incident on t has proposed c^*, then
    (a) If $c^* \in \mathrm{N}^i$ then color e with c^*.
    (b) If $c^* \in \mathrm{R}^i(b)$, and $e = (b,t)$ is the h^{th} edge incident on t proposing from a phase i palette (namely $\mathrm{R}^i(b) \cup \mathrm{N}^i$), then color e with c^* with probability $\frac{q^i(h)}{p^i(t,h)}$.
- If e is not yet colored, color it greedily from P^∞. If no such color is available in P^∞, then abort.

Figure 3: The Online Algorithm

the last step, i. e., each edge gets colored from one of the palettes. The total number of colors used is then

$$\sum_{i=1}^{L} |\mathrm{P}^i| + \sum_{i=1}^{L} |\mathrm{N}^i| + |\mathrm{P}^\infty|.$$

The analysis is in three steps: Firstly, we bound the number of colors used for round 1 edges, and prove correlations between the sets of rejected colors at different vertices (Section 2.1). Secondly, we bound the number of colors used for round 2 edges (Section 2.3). Finally we put all the bounds together to get the full count of the number of colors used (Section 2.4).

But, before proceeding to the analysis, we first give some notation and definitions that will be useful later.

Definition 2.1. We say that a vertex or an edge *reaches phase i* if it chooses a color from P^i (for a round 1 edge) or from $\mathrm{R}^i(\cdot) \cup \mathrm{N}^i$ (round 2 edge). For any vertex v (top or bottom), $i \in [1, L]$, and $j \in \{1, 2\}$ define $\deg_j^i(v)$ as the number of Round j edges incident on v which propose a color in phase i. Also, for any top vertex t, $i \in [1, L]$, $j \in \{1, 2\}$, and $h \in \left[1, \deg_j^i(t)\right]$, define $e_j^i(t, h)$ to be the h^{th} edge (in the arrival order) incident on t in round j, which proposes a color in phase i. As a matter of notation, throughout we will use the term "with high probability" (or "w. h. p.") to mean "with probability at least $1 - \mathrm{poly}(1/n)$," and we will use $\sim$ to denote "equal up to higher order terms."

2.1 Round 1 analysis

In this section, we analyze Round 1 of the algorithm. Recall the recursive definition $\Delta^{(i+1)} = \Delta^{(i)}/\mathrm{e} - o(\Delta^{(i)})$, with $\Delta^{(1)} = r\Delta - o(\Delta)$.

Lemma 2.2. *For all $i \in [1, L]$, $t \in T$ and $h \in [1, \deg_1^i(t)]$, the probability that $e_1^i(t, h)$ is colored with a color from P^i is $\left(1 - 1/\Delta^{(i)}\right)^{h-1}$.*

Proof. $e_1^i(t, h)$ gets accepted if and only if none of the previous edges $e_1^i(t, h')$ $(1 \le h' \le h - 1)$ have chosen the same color as $e_1^i(t, h)$. The lemma then follows by noticing that each node picks its color independently and uniformly at random from P^i. □

Lemma 2.3. *For all $i \in [1, L]$ and $t \in T$, $\deg_1^i(t) = \Delta^{(i)}$ with high probability.*

Proof. Note that $\deg_1^i(t) = \min\left\{\Delta^{(i)}, \overline{\deg_1^i}(t)\right\}$. Thus, we only need to prove $\overline{\deg_1^i}(t) \ge \Delta^{(i)}$ w. h. p. We do the proof by induction on i. For $i = 1$, the result directly follows from the definition of early edges and the fact that the input order is a random permutation. Now, assume the statement is true for i, and we will prove it for $i + 1$. Note that for any top node t and any $h \le \Delta^{(i)}$, we have by Lemma 2.2:

$$\Pr[e_1^i(t, h) \text{ gets accepted}] = \left(1 - \frac{1}{\Delta^{(i)}}\right)^{h-1}.$$

Therefore

$$\mathrm{E}\left[\overline{\deg_1^{i+1}}(t) \;\middle|\; \deg_1^i(t) = \Delta^{(i)}\right] = \Delta^{(i)} - \sum_{h=1}^{\Delta^{(i)}} \left(1 - \frac{1}{\Delta^{(i)}}\right)^{h-1} = \Delta^{(i)} \left(1 - \frac{1}{\Delta^{(i)}}\right)^{\Delta^{(i)}}$$

and hence

$$\Delta^{(i)} \geq \mathrm{E}\left[\overline{\deg_1^{i+1}}(t)\right] \geq \frac{\Delta^{(i)}}{\mathrm{e}}(1-o(1))\,.$$

A simple application of the method of bounded differences, as in [5], section 6.5.1, shows that $\overline{\deg_1^{i+1}}(t)$ is sharply concentrated around its mean. Therefore, w. h. p. $\overline{\deg_1^{i+1}}(t) \geq \Delta^{(i+1)}$, that is $\deg_1^{i+1}(t) = \Delta^{(i+1)}$. □

Lemma 2.4. *For any $i \in [1,L]$, h, $(b,t) \in E$, $\Pr\left[(b,t) = e_1^i(t,h)\right]$ only depends on i, h (i. e., is independent of (b,t)).*

Proof. We do the proof by induction on i. For $i = 1$, the statement follows directly from the symmetry of the input order (random permutation). Now, assume the claim is true for i, and we will prove it for $i+1$. By definition of the algorithm, for any $h > \Delta^{(i+1)}$, $\Pr\left[(b,t) = e_1^{i+1}(t,h)\right] = 0$. If $h \leq \Delta^{(i+1)}$, we have

$$\Pr\left[(b,t) = e_1^{i+1}(t,h)\right] = \sum_{h' \geq h} \Pr\left[(b,t) = e_1^{i+1}(t,h) \mid (b,t) = e_1^i(t,h')\right] \Pr\left[(b,t) = e_1^i(t,h')\right]\,.$$

For each h', we know by the inductive assumption that $\Pr\left[(b,t) = e_1^i(t,h')\right]$ is independent of (b,t). Hence, we only need to show that $\Pr\left[(b,t) = e_1^{i+1}(t,h) \mid (b,t) = e_1^i(t,h')\right]$ is only dependent on i,h,h'. Assume that

$$\begin{aligned} C = \{(c_1, c_2, \ldots, c_{h'}) \mid \forall 1 \leq j \leq h' : c_j \in \mathrm{P}^i, &\\ &\text{the number of different colors among } c_1, \ldots, c_{h'-1} \text{ is } h'-h,\\ &\text{and } \exists 1 \leq j \leq h'-1 : c_{h'} = c_j\}\,. \end{aligned}$$

Then,

$$\Pr\left[(b,t) = e_1^{i+1}(t,h) \mid (b,t) = e_1^i(t,h')\right] = |C| \left(\frac{1}{\Delta^{(i)}}\right)^{h'}\,,$$

which clearly only depends on i,h,h'. □

Lemma 2.5. *For all $i \in [1,L]$ and $b \in B$, $\deg_1^i(b) = \Delta^{(i)} \pm o\left(\Delta^{(i)}\right)$ w. h. p.*

Proof. First, notice that from Lemma 2.4, for any $(b',t) \in E$ and any h, we have

$$\Delta \cdot \Pr\left[(b',t) = e_1^i(t,h)\right] = \sum_{\{b'' \mid (b'',t) \in E\}} \Pr\left[(b'',t) = e_1^i(t,h)\right] = \Pr\left[\deg_1^i(t) \geq h\right]\,.$$

Then, from Lemma 2.3, we have (for any $(b',t) \in E$)

$$\Pr\left[(b',t) = e_1^i(t,h)\right] = \begin{cases} \frac{1-o(1)}{\Delta} & \text{if } h \leq \Delta^{(i)},\\ 0 & \text{if } h > \Delta^{(i)}. \end{cases}$$

Then, with t being an arbitrary neighbor of b, we have

$$\begin{aligned} E\left[\deg_1^i(b)\right] &= \sum_{\{t'|(b,t')\in E\}} \Pr\left[(b,t') \text{ reaches phase } i\right] = \Delta\cdot\Pr\left[(b,t) \text{ reaches phase } i\right] \\ &= \Delta\cdot\sum_{h=1}^{\Delta^{(i)}} \Pr\left[(b,t) = e_1^i(t,h)\right] = \Delta^{(i)}(1-o(1)). \end{aligned}$$

The sharp concentration proof, using the method of average bounded differences, then exactly follows the one in [5], section 7.5.3. □

Therefore, from Lemma 2.3 and Lemma 2.5, the node degrees drop exponentially quickly, and we get the following theorem:

Theorem 2.6. *With high probability, all round 1 edges get colored from* $\bigcup \mathrm{P}^i \cup \mathrm{P}^\infty$*, i. e., the algorithm does not abort during Round 1.*

2.2 Correlations between rejected colors at the end of round 1

In this section, we prove a proposition which will prove very useful later in the analysis of the second round of the algorithm. First, we give a definition.

Definition 2.7. For $i \in [1, L]$, and for a top vertex t, let $E^i(t)$ be the set of colors in P^i which were not used to color a Round 1 edge incident on t. (In other words, it is the set of colors from P^i which no bottom vertex proposes for a Round 1 edge incident on t.)

Before proceeding to the main proposition, we present the following lemma which is a direct consequence of the definitions of $\mathrm{R}^i(b), E^i(t)$ and Lemma 2.3, Lemma 2.5.

Lemma 2.8. *For all* $i \in [1,L]$*,* $b \in B$*,* $t \in T$*, w. h. p.* $|\mathrm{R}^i(b)| \sim \Delta^{i+1}$ *and* $|E^i(t)| \sim \Delta^{(i+1)}$.

Now, we present the main proposition.

Proposition 2.9. *For all* $i \in [1,L], b \in B, t \in T$ *such that there is no Round 1 edge* (b,t)*, we have w. h. p.*

$$\left|\mathrm{R}^i(b) \cap E^i(t)\right| \geq \left(\frac{1}{\mathrm{e}} - o(1)\right)\left|\mathrm{R}^i(b)\right|.$$

Proof. To simplify the presentation of the proof, assume $(b,t) \notin E$. The proof for the case where (b,t) is an edge, but not a Round 1 edge, is exactly similar. We first prove that

$$\mathrm{E}\left[\left|\mathrm{R}^i(b) \cap E^i(t)\right|\right] \geq \left(\frac{1}{\mathrm{e}} - o(1)\right)\mathrm{E}\left[\left|\mathrm{R}^i(b)\right|\right].$$

Let the event F^i be defined as follows:

$$F^i = \left\{\exists b' \in B \text{ or } t' \in T : \deg_1^j(b') < \Delta^{(j)} - o(\Delta^j) \text{ or } \deg_1^j(t') < \Delta^{(j)} \text{ for some } 1 \leq j \leq i\right\}.$$

We know, from Lemma 2.3 and Lemma 2.5, that $\Pr\left[F^i\right] = o(1)$. Furthermore, we have $0 \le \left|\mathrm{R}^i(b)\right| \le \Delta$ and hence

$$\mathrm{E}\left[\left|\mathrm{R}^i(b)\right|\right] = \mathrm{E}\left[\left|\mathrm{R}^i(b)\right| \;\middle|\; F^i\right]\Pr\left[F^i\right] + \mathrm{E}\left[\left|\mathrm{R}^i(b)\right| \;\middle|\; \overline{F^i}\right]\Pr\left[\overline{F^i}\right] = o(1) + (1-o(1))\,\mathrm{E}\left[\left|\mathrm{R}^i(b)\right| \;\middle|\; \overline{F^i}\right],$$

where $\overline{F^i}$ is the complement of F^i. Similarly, one can see

$$\mathrm{E}\left[\left|\mathrm{R}^i(b)\cap E^i(t)\right|\right] \ge (1-o(1))\,\mathrm{E}\left[\left|\mathrm{R}^i(b)\cap E^i(t)\right| \;\middle|\; \overline{F^i}\right].$$

Hence, to prove the inequality in expectation, it suffices to prove that

$$\mathrm{E}\left[\left|\mathrm{R}^i(b)\cap E^i(t)\right| \;\middle|\; \overline{F^i}\right] \ge \left(\frac{1}{\mathrm{e}} - o(1)\right)\mathrm{E}\left[\left|\mathrm{R}^i(b)\right| \;\middle|\; \overline{F^i}\right].$$

Thus, assume $\overline{F^i}$ happens. (I. e., all probabilities and expectations are conditioned on $\overline{F^i}$. We will omit explicitly writing this conditioning for the sake of brevity of expressions.) Then, we have $\mathrm{E}\left[\left|\mathrm{R}^i(b)\right|\right] = \sum_{c\in\mathrm{P}^i}\Pr\left[c\in\mathrm{R}^i(b)\right]$. Denoting by $b \xrightarrow{c} z$ that the algorithm chooses the tentative proposal of $c \in \mathrm{P}^i$ for (b,z) and denoting the event

$$\left\{(b,t') = e_1^i(t',h) \text{ and } b \xrightarrow{c} t'\right\}$$

by $H^i(t',h,c)$ (for any $c\in\mathrm{P}^i$, $h\in[1,\Delta^{(i)}]$, $(t',b)\in E$), we have for any $c\in\mathrm{P}^i$

$$\Pr\left[c\in\mathrm{R}^i(b)\right] = \sum_{\substack{(b,t')\in E,\\ 1\le h\le\Delta^{(i)}}} \Pr\left[H^i(t',h,c)\right]\,\Pr\left[t' \text{ rejects } b \;\middle|\; H^i(t',h,c)\right]. \tag{2.1}$$

Similarly, $\mathrm{E}\left[\left|\mathrm{R}^i(b)\cap E^i(t)\right|\right] = \sum_{c\in\mathrm{P}^i}\Pr\left[c\in\mathrm{R}^i(b)\cap E^i(t)\right]$ and, for any $c\in\mathrm{P}^i$,

$$\begin{aligned}\Pr\left[c\in\mathrm{R}^i(b)\cap E^i(t)\right] = \sum_{\substack{(b,t')\in E\\ 1\le h\le\Delta^{(i)}}} &\Pr\left[H^i(t',h,c)\right]\cdot\Pr\left[c\in E^i(t) \;\middle|\; H^i(t',h,c)\right]\\ &\cdot\Pr\left[t' \text{ rejects } b \;\middle|\; H^i(t',h,c), c\in E^i(t)\right].\end{aligned} \tag{2.2}$$

But,

$$\Pr\left[c\in E^i(t) \;\middle|\; H^i(t',h,c)\right] = \left(1-\frac{1}{\Delta^{(i)}}\right)^{\Delta^{(i)}} = \frac{1}{\mathrm{e}} - o(1)$$

and

$$\Pr\left[t' \text{ rejects } b \;\middle|\; H^i(t',h,c), c\in E^i(t)\right] \ge \Pr\left[t' \text{ rejects } b \mid H^i(t',h,c)\right],$$

as knowing $c\in E^i(t)$ increases the chance of c being proposed to t' by nodes other than b, which increases the chance that t' rejects b's proposal of c. Hence, comparing equations (2.1) and (2.2), we get

$$\Pr\left[c\in\mathrm{R}^i(b)\cap E^i(t)\right] \ge \left(\frac{1}{\mathrm{e}} - o(1)\right)\Pr\left[c\in\mathrm{R}^i(b)\right] \quad (\forall c\in\mathrm{P}^i)$$

and hence

$$\mathrm{E}\left[\left|\mathrm{R}^i(b)\cap E^i(t)\right| \,\middle|\, \overline{F^i}\right] \geq \left(\frac{1}{\mathrm{e}} - o(1)\right)\mathrm{E}\left[\left|\mathrm{R}^i(b)\right| \,\middle|\, \overline{F^i}\right],$$

which implies that

$$\mathrm{E}\left[\left|\mathrm{R}^i(b)\cap E^i(t)\right|\right] \geq \left(\frac{1}{\mathrm{e}} - o(1)\right)\mathrm{E}\left[\left|\mathrm{R}^i(b)\right|\right].$$

We know $|\mathrm{R}^i(b)|$ is sharply concentrated around its mean. The proof for the sharp concentration of $|\mathrm{R}^i(b)\cap E^i(t)|$, using the method of average bounded differences, also closely follows the proof in [5], section 7.5.3. Therefore, the inequality also holds with high probability. □

Note that, for any $i \in [1,L]$, both $\mathrm{R}^i(b)$ and $E^i(t)$ are random subsets of P^i, with

$$\left|\mathrm{R}^i(b)\right| \sim \left|E^i(t)\right| \sim \frac{1}{\mathrm{e}}\left|\mathrm{P}^i\right|.$$

The above proposition shows these sets are also w. h. p. positively correlated. This will be important in our analysis of the second round of the algorithm, presented in the next section.

2.3 Round 2 analysis

Round 2 begins at time $rn\Delta$. At this point, we have the following set-up, as proved above: Every bottom vertex b has, for each $i \in [1,L]$, a set $\mathrm{R}^i(b) \subseteq \mathrm{P}^i$ of colors rejected in Round 1. Every top vertex has, for each $i \in [1,L]$, a set of unused colors $E^i(t) \subseteq \mathrm{P}^i$. We proved that w. h. p.

$$\left|\mathrm{R}^i(b)\right| \sim |E^i(t)| \sim \Delta^{(i+1)} \quad \text{and} \quad \frac{\left|\mathrm{R}^i(b)\cap E^i(t)\right|}{\mathrm{R}^i(b)|} \geq \frac{1}{\mathrm{e}}.$$

For each $i \in [1,L]$, we also have the augmenting palette N^i of size x_i (to be determined later).

Definition 2.10 (Probabilities $p^i(t,h), q^i(h)$)**.** For $i \in [1,L]$, define $m_i = \Delta^{(i+1)} + x_i$ to be the total number of colors that a bottom vertex has in Round 2 to propose for its phase i incident edges (recall that by Lemma 2.8, $\forall b$, $|\mathrm{R}^i(b)| \sim \Delta^{(i+1)}$ w. h. p.).

For $i \in [1,L]$, $t \in T$, and $h \in [1, \deg_2^i(t)]$, and denoting $e_2^i(t,h)$ by (b,t), define $p^i(t,h)$ to be the probability that, given that $e_2^i(t,h)$ proposes a (randomly chosen) color from $\mathrm{R}^i(b)$, this color is in $E^i(t)$, and has not been proposed by any previously arrived edge $e_2^i(t,h')$ ($h' < h$). In other words $p^i(t,h)$ is the "natural probability" (over the choice of colors proposed for each round 2 edge) of the proposal for $e_2^i(t,h)$ succeeding, given that it chose a color from $\mathrm{R}^i(b)$. This probability value can be calculated by the online algorithm when the edge arrives.

Finally, define

$$q^i(h) = \frac{1}{\mathrm{e}}\left(1 - \frac{1}{m_i}\right)^{h-1}.$$

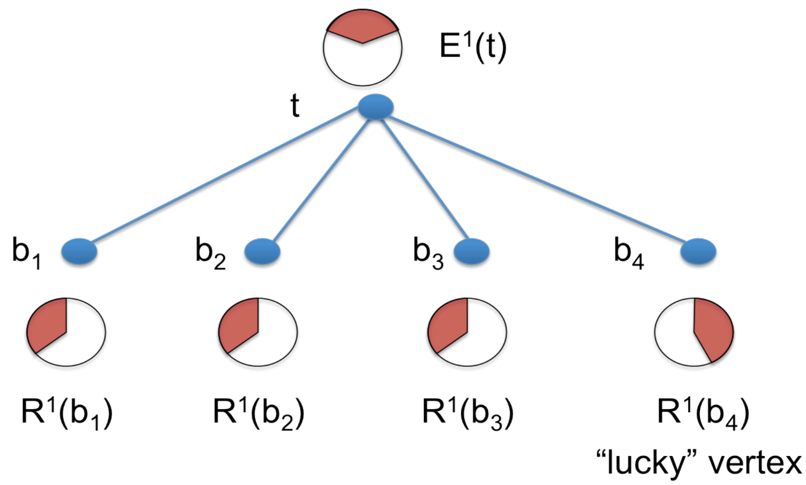

Figure 4: In this example, each of $\mathrm{R}^1(b_1)$, $\mathrm{R}^1(b_2)$, $\mathrm{R}^1(b_3)$ and $\mathrm{R}^1(b_4)$ are positively correlated to $E^1(t)$ (as required by Proposition 2.9). However, $\mathrm{R}^1(b_1) = \mathrm{R}^1(b_2) = \mathrm{R}^1(b_3)$, while $\mathrm{R}^1(b_4) \cup \mathrm{R}^1(b_1) = \emptyset$. The edge (b_4,t) has a much greater chance of success than the other three edges. To see this, note that the color choices of the other three edges may conflict with each other, and hence lead to failure in this palette for all but one of the conflicting edges. But the color choice of (b_4,t) would never face a conflict—whenever the color belongs to $E^1(t)$, it will be a success. (For simplicity, we have ignored the existence of the new palette N^1 in round 2 in this example.)

Remark 2.11. Note that $p^i(t,h)$ depends crucially on which edge (b,t) is in the h^{th} position among the phase i edges incident on t in Round 2. This is because the edge succeeds in coloring itself if no previous such edge chose the same color. Since we have no guarantee on how the different $\mathrm{R}^i(\cdot)$ sets intersect, this probability may depend on the identities of all the first h such edges. From this observation, we see that, unlike in Round 1, different bottom vertices may have very different probabilities of success, even if their edges were in the same position with respect to a top vertex t. This may result in the degrees of different vertices evolving differently over the phases, which makes the analysis very difficult. We rectify this issue by showing that $q^i(h)$ is a lower bound on all of these probabilities $p^i(t,h)$, and by artificially rejecting each possible coloring using random coin tosses with appropriate bias $(q^i(h)/p^i(t,h))$ to make the probability of the ith edge succeeding to become exactly $q^i(h)$. Note that the values of $p^i(t,h)$ are available to the online algorithm at the time that the edge arrives.

Figure 4 illustrates the reason why some round 2 edges can be luckier than others, based on the intersection patterns of the sets of available colors at bottom vertices (even though the sets at every pair of bottom and top vertices are positively correlated).

The following lemma lower bounds all the acceptance probabilities $p^i(t,h)$.

Lemma 2.12. $p^i(t,h) \geq q^i(h)$.

Proof. Suppose $e^i_2(t,h)$ is the edge (b,t). Let new_h be the event that the color proposed by $e^i_2(t,h)$ from $\mathrm{R}^i(b) \cup \mathrm{N}^i$ is in $E^i(t)$ and is not proposed by any of the previous edges $e^i_2(t,h')$ $(1 \leq h' \leq h-1)$. For

$h' \in [1, h-1]$, let $\mathrm{R}^i(h')$ denote the set $\mathrm{R}^i(b')$, where $e_2^i(t, h')$ has b' as its bottom vertex. Define

$$v_c^h := \left|\left\{h' \;\middle|\; 1 \le h' < h \text{ and } c \in \mathrm{R}^i(h')\right\}\right| .$$

We have

$$\begin{aligned}
p^i(t,h) &:= \Pr\left[\mathrm{new}_h \;\middle|\; (b,t) \text{ gets a proposal from } \mathrm{R}^i(b)\right] \\
&= \sum_{c \in \mathrm{R}^i(b)} \Pr\left[\mathrm{new}_h \;\middle|\; b \xrightarrow{c} t\right] \Pr\left[b \xrightarrow{c} t \;\middle|\; (b,t) \text{ gets a proposal from } \mathrm{R}^i(b)\right] \\
&= \sum_{c \in \mathrm{R}^i(b) \cap E^i(t)} \left(1 - \frac{1}{m_i}\right)^{v_c^h} \frac{1}{\left|\mathrm{R}^i(b)\right|} \\
&\ge \frac{\left|\mathrm{R}^i(b) \cap E^i(t)\right|}{\left|\mathrm{R}^i(b)\right|} \left(1 - \frac{1}{m_i}\right)^{h-1} \\
&\ge \frac{1}{\mathrm{e}} \left(1 - \frac{1}{m_i}\right)^{h-1} \quad =: \; q^i(h) ,
\end{aligned}$$

where the last inequality uses the positive correlation between $\mathrm{R}^i(b)$ and $E^i(t)$, from Proposition 2.9. □

Definition 2.13. Define

$$\mathrm{succ}^i(h) := \frac{x_i}{m_i}\left(1 - \frac{1}{m_i}\right)^{h-1} + \frac{\Delta^{(i+1)}}{m_i} q^i(h) = \left(1 - \frac{1}{m_i}\right)^{h-1} \left(\frac{x_i}{m_i} + \frac{1}{\mathrm{e}} \frac{\Delta^{(i+1)}}{m_i}\right) .$$

Corollary 2.14. *The probability that the edge $e_2^i(t,h)$ succeeds in coloring itself (in phase i of Round 2) is exactly* $\mathrm{succ}^i(h)$.

Proof. Using the notation of the proof of Lemma 2.12, and defining acc_h to be the event that $e_2^i(t,h)$ succeeds (i. e., the event of interest), we have:

$$\Pr[\mathrm{acc}_h] = \Pr\left[\mathrm{acc}_h \;\middle|\; c_h \in \mathrm{N}^i\right] \Pr\left[c_h \in \mathrm{N}^i\right] + \Pr\left[\mathrm{acc}_h \;\middle|\; c_h \in \mathrm{R}^i(b)\right] \Pr\left[c_h \in \mathrm{R}^i(b)\right] .$$

But,

$$\Pr\left[c_h \in \mathrm{N}^i\right] = \frac{\left|\mathrm{N}^i\right|}{m_i} = \frac{x_i}{m_i}, \quad \Pr\left[c_h \in \mathrm{R}^i(b)\right] = \frac{\Delta^{i+1}}{m_i}, \quad \Pr\left[\mathrm{acc}_h \;\middle|\; c_h \in \mathrm{N}^i\right] = \left(1 - \frac{1}{m_i}\right)^{h-1},$$

and

$$\begin{aligned}
\Pr\left[\mathrm{acc}_h \;\middle|\; c_h \in \mathrm{R}^i(b)\right] &= \Pr\left[\mathrm{acc}_h \;\middle|\; \mathrm{new}_h, c_h \in \mathrm{R}^i(b)\right] \Pr\left[\mathrm{new}_h \;\middle|\; c_h \in \mathrm{R}^i(b)\right] \\
&= \frac{q^i(h)}{p^i(t,h)} p^i(t,h) = q^i(h) .
\end{aligned}$$

Putting all these together proves the corollary. □

Remark 2.15. By using the extra randomness in artificially reducing the probabilities of success for "lucky" edges, we have decoupled the probability of success from the identities of the vertices, and how the different $\mathrm{R}^i(b)$ sets intersect. Note that all we used was that the $\mathrm{R}^i(b)$ and $E^i(t)$ sets are positively correlated (Proposition 2.9)—we made no assumption on how different $\mathrm{R}^i(b)$ and $\mathrm{R}^i(b')$ intersect. Having managed this, the rest of the proof is basically identical to the Round 1 analysis.

Definition 2.16. Define the sequence Γ^i by the following recursion:

$$\Gamma^1 = \frac{r\Delta}{\mathrm{e}},$$
$$\Gamma^{i+1} = \frac{\Gamma^i}{\mathrm{e}} + \left(1-\frac{1}{\mathrm{e}}\right)^2 \Delta^{(i+1)}.$$

Choose $x_i := \Gamma^i - \Delta^{(i+1)}$. Thus, $m_i = \Gamma^i$.

Lemma 2.17. *For all $i \in [1,L]$ and for all nodes v, $\deg_2^i(v) \sim \Gamma^i$ with high probability.*

Proof. The proof closely follows the one in round 1. So, we only show the recursion. For an arbitrary top node t, the expected number of Round 2 edges incident on t which succeed in coloring themselves in phase i (of Round 2) is

$$E\left[\sum_{h=1}^{\deg_2^i(t)} \mathrm{succ}^i(h)\right] \sim \sum_{h=1}^{m_i}\left(1-\frac{1}{m_i}\right)^{h-1}\left(\frac{x_i}{m_i}+\frac{1}{\mathrm{e}}\frac{\Delta^{(i+1)}}{m_i}\right)$$
$$\simeq \left(x_i + \frac{1}{\mathrm{e}}\Delta^{(i+1)}\right)\left(1-\frac{1}{\mathrm{e}}\right)$$

where the first equality uses Corollary 2.14. Therefore, the number of unsuccessful edges proceeding to the next phase is

$$\begin{aligned}
\deg_2^{i+1} &\sim \deg_2^i(t) - \left(x_i + \frac{1}{\mathrm{e}}\Delta^{(i+1)}\right)\left(1-\frac{1}{\mathrm{e}}\right) \\
&\sim \Gamma^i - \left(x_i + \frac{1}{\mathrm{e}}\Delta^{(i+1)}\right)\left(1-\frac{1}{\mathrm{e}}\right) \\
&= \frac{\Gamma^i}{\mathrm{e}} + \left(1-\frac{1}{\mathrm{e}}\right)^2\Delta^{(i+1)} \\
&= \Gamma^{i+1}
\end{aligned}$$

where the second equation is by the induction hypothesis, the third is because $\Gamma^i = x_i + \Delta^{i+1}$ by definition of x_i, and the last is by definition of the sequence $\{\Gamma^i\}_{i\geq 1}$. □

Corollary 2.18. *The algorithm succeeds in coloring all edges w. h. p., i. e., it doesn't abort in Round 2.*

Proof. It can then be easily seen that the recursion for Γ^i makes all the vertex degrees fall exponentially quickly (e. g., at a rate at least as large as $1/\mathrm{e} + (1-1/\mathrm{e})^2$). So, the palette P^∞ suffices to greedily color the edges not colored by $\bigcup_{i=1}^{L}\mathrm{P}^i$. □

2.4 Putting it together

First note that each probabilistic statement claimed in the proof holds w. h. p., i. e., with probability of error at most $1/\text{poly}(n)$. We only have $\text{poly}(n)$ such statements ($2n$ vertices, 2 rounds and $L = O(\log n)$ palettes per round). So, by a simple union bound, we know that w. h. p. all of those results hold, in which case the total number of colors that we use is at most

$$\sum_{i\geq 1} |\mathrm{P}^i| + \sum_{i\geq 1} x_i + |\mathrm{P}^\infty|\,.$$

Now, $|\mathrm{P}^i| = \Delta^{(i)}$, and by choice (Definition 2.16), $x_i = \Gamma^i - \Delta^{(i+1)}$. Thus, the above summation is at most

$$r\Delta + \sum_{i\geq 1} \Gamma_i + o(\Delta)\,.$$

By definition,

$$\Gamma_{i+1} = \frac{\Gamma_i}{\mathrm{e}} + \left(1 - \frac{1}{\mathrm{e}}\right)^2 \frac{r\Delta}{\mathrm{e}^i}\,.$$

Summing up all these equalities (for all $i \geq 1$) and defining $S = \sum_{i\geq 1} \Gamma_i$, we have

$$S - \Gamma_1 \;=\; \frac{S}{\mathrm{e}} + \left(1 - \frac{1}{\mathrm{e}}\right)^2 \sum_{i\geq 1} \frac{r\Delta}{\mathrm{e}^i}$$

from which we can calculate

$$S = r\Delta\left(\frac{1}{\mathrm{e}} + \frac{1}{\mathrm{e}-1}\right)\,.$$

Thus, recalling that $r = \mathrm{e}/(\mathrm{e}+1)$, we have

$$r\Delta + S \;=\; r\Delta\left(1 + \frac{1}{\mathrm{e}} + \frac{1}{\mathrm{e}-1}\right) \;=\; \frac{1 + \frac{1}{\mathrm{e}} + \frac{1}{\mathrm{e}-1}}{1 + \frac{1}{\mathrm{e}}}\Delta \;<\; 1.43\Delta\,.$$

Therefore, the total number of colors used is at most $1.43\Delta + o(\Delta)$. We record this result in the following theorem.

Theorem 2.19. *The online algorithm colors the edges of a regular bipartite graph with degree $\Delta = \omega(\log n)$, w. h. p., using at most $1.43\Delta + o(\Delta)$ colors.*

3 Extending the algorithm to more than two rounds

A natural idea is to extend the algorithm to more than two rounds, say to K rounds, so that each color is tried at each bottom vertex up to K times. This way, we may expect to reduce the total number of colors used. We define such an algorithm in Figure 5, and describe it informally below.

The algorithm partitions the input edges into K rounds: The first $r_1 n\Delta$ edges to arrive are called Round 1 edges, the next $r_2 n\Delta$ edges are Round 2 edges, and so on, where the sequence of numbers $r_1, r_2, \cdots, r_K$

For all bottom vertices b, all $i \in [1,L]$, and all $j \in [1,K]$, initialize $\mathrm{Tried}_j^i(b) = \mathrm{R}_j^i(b) = \emptyset$.

When the s^{th} edge $e = (b,t)$ arrives in the online order:

- Let j be s.t. $\sum_{\ell=1}^{j-1} r_\ell < s \le \sum_{\ell=1}^{j} r_\ell$ (the edge is a Round j edge).
- Set $i = 0$
- While (e is not colored and $i < L$):
  - $i{+}{+}$
  - Pick a color c^* uniformly at random from $\left(\mathrm{N}_j^i \cup \left(\bigcup_{\ell=1}^{j-1} \mathrm{R}_\ell^i(b)\right)\right) \setminus \mathrm{Tried}_j^i(b)$. Set $\mathrm{Tried}_j^i(b) = \mathrm{Tried}_j^i(b) \cup \{c^*\}$. (If no such color exists then continue).
  - If no previously arrived edge incident on t was proposed c^*, then
    (a) If $c^* \in \mathrm{N}_j^i$ then color e with c^*
    (b) If $c^* \in \mathrm{R}_\ell^i(b)$ (for some $\ell < j$) then color e with c^* with probability $\frac{q_{\ell,h}^i}{p_{\ell,h}^i(t)}$, and set $\mathrm{R}_\ell^i(b) = \mathrm{R}_\ell^i(b) - c^*$.
  - Else: If $c^* \in \mathrm{N}_j^i$, set $\mathrm{R}_j^i(b) = \mathrm{R}_j^i(b) \cup \{c^*\}$.
- If e is not yet colored, color it greedily from P^∞. If no such color is available in P^∞, then abort.

Figure 5: The Online Algorithm

will be chosen later. The algorithm keeps L palettes for each round, so that we have a total of KL palettes N_j^i, ($i \in [1,L], j \in [1,K]$), where the size of each palette will be determined later. The algorithm also has a special palette P^∞ with $o(\Delta)$ colors.

So, as an edge (b,t) arrives, the arrival time determines which round it belongs to—say it belongs to round j. Vertex b proposes a random color c for this edge from the union of N_j^1 and the sets of colors rejected from b in the earlier palettes $\mathrm{N}_1^1, \ldots, \mathrm{N}_{j-1}^1$. If t has not used the color c earlier, then the edge is colored c. Else, b proposes a random color for this edge from the union of N_j^2 and the sets of colors rejected from b in the earlier palettes $\mathrm{N}_1^2, \ldots, \mathrm{N}_{j-1}^2$, and t decides whether or not to accept it, and so on. If after L such attempts, the edge is still not colored, we will color it greedily from P^∞. Also, as in the 2 round algorithm, we perform some additional *artificial rejections* of the proposed colors for an edge with random coin tosses of appropriate biases, so as to keep the success probabilities across all vertices the same.

In this section, we show how we can analyze the above algorithm. We start with some definitions.

Definition 3.1. The sequence $\{g_\ell\}_{\ell\geq 0}$ is defined recursively as follows:

$$g_0 = 1,$$
$$g_{\ell+1} = g_\ell - \left(1-\frac{1}{\mathrm{e}}\right)g_\ell^2.$$

Definition 3.2. We say that a round j edge ($1\leq j\leq K$) reaches phase i ($1\leq i\leq L$), if it ever proposes a color from $\bigcup_{r\leq j} N_r^i$. For a vertex v, define $\deg_j^i(v)$ as the number of round j edges incident on v which reach phase i. For $t\in T$ (respectively, $b\in B$), $i\in[1,L]$, $j\in[1,K]$, and $r\leq j$, define $E_{r,j}^i(t)$ (respectively, $A_{r,j}^i(b)$) to be the set of colors from N_r^i which have still not been used to color any edges incident on t (respectively, b), at the beginning of round j. Thus, $A_{j,j}^i(b)=N_j^i$, and $A_{r,j}^i(b)$ equals the value of $R_r^i(b)$ (as in the Online Algorithm in Figure 5) at the beginning of round j. For $i\in[1,L]$, $j\in[1,K]$, $t\in T$ and $h\in[1,\deg_j^i(t)]$, let $e_j^i(t,h)$ be the h^{th} edge incident on t which arrives in round j and reaches phase i. Suppose $e_j^i(t,h)=(b,t)$, and that b proposes (in the i^{th} phase) the color c for this edge. Then, define $p_{r,j}^i(t,h)$ ($\forall r\leq j$) as the probability that, given $c\in A_{r,j}^i(b)$, c is also in $E_{r,j}^i(t)$, and that no edge $e_j^i(t,h')$, for $h'<h$, also proposed c. For $i\in[1,L]$, $j\in[1,K]$, define $d_j^i=\sum_{r=1}^{j} g_{j-r}|N_r^i|$. Also, with $r\leq j$, define

$$q_{r,j}^i(h) := g_{j-r}\left(1-\frac{1}{d_j^i}\right)^{h-1}.$$

Finally, define $\mathrm{acc}_j^i(t,h)$ as the event that $e_j^i(t,h)$ gets colored at phase i (i. e., does not reach phase $i+1$).

Now, using the above definitions, we present the following theorem:

Theorem 3.3. *For all $i\in[1,L]$, $j\in[1,K]$, $1\leq r\leq j$, $b\in B$, $t\in T$, and $h\in[1,\deg_j^i(t)]$, we have:*

1. $|A_{r,j}^i(b)|\sim|E_{r,j}^i|\sim g_{j-r}|N_r^i|$ *w. h. p.*
2. $\deg_j^i(b)\sim\deg_j^i(t)\sim d_j^i$ *w. h. p.*
3. *If there is no edge (b,t) belonging to the rounds earlier than j, then w. h. p. $A_{r,j}^i(b)$ and $E_{r,j}^i(t)$ are (essentially) positively correlated, i. e., $|A_{r,j}^i(b)\cap E_{r,j}^i(t)|\geq g_{j-r}|A_{r,j}^i(b)|$.*
4. $p_{r,j}^i(t,h)\geq q_{r,j}^i(h)$.

Proof (sketch): The proof is done by induction on the round number j. For $j=1$, all the items in the theorem are trivially true. So, the basis of the induction is fine. The proofs for the inductive step are very similar to the ones done for the corresponding statements for the 2 Round algorithm. So, we only provide proof sketches here. We do the inductive proof for the items in the following order: $3,4,1,2$.

The proof for item 3 follows exactly the proof of Proposition 2.9. The only difference between Round 1 and any other Round is the artificial rejections done by the random coin tosses. But, clearly, that does not change the correlation structure between the unused color sets (i. e., $A_{r,j}^i(b)$ and $E_{r,j}^i(t)$).

Then, we prove the item 4. The proof is very similar to that of Lemma 2.12. Suppose $e^i_j(t,h) = (b',t)$. Let new_h be the event that the color proposed for $e^i_j(t,h)$, by b' in phase i, has not been already used by t, and that no $e^i_j(t,h')$ chose the same color. Define

$$v^h_c = \left| \left\{ 1 \le h' < h \;\middle|\; c \in \bigcup_{r' \le j} A^i_{r',j}(h') \right\} \right|$$

where by $A^i_{r',j}(h')$, we mean the set $A^i_{r',j}(b'')$ for bottom node b'' such that $e^i_j(t,h') = (b'',t)$. Then,

$$\begin{aligned}
p^i_{r,j}(t,h) &= \Pr\left[\mathrm{new}_h \mid e^i_j(t,h) \text{ gets a proposal from } A^i_{r,j}(b')\right] \\
&= \sum_{c \in A^i_{r,j}(b')} \Pr\left[\mathrm{new}_h \mid b' \xrightarrow{c} t\right] \cdot \Pr\left[b' \xrightarrow{c} t \mid e^i_j(t,h) \text{ gets a proposal from } A^i_{r,j}(b')\right] \\
&= \sum_{c \in A^i_{r,j}(b') \bigcap E^i_{r,j}(t)} \left(1 - \frac{1}{d^i_j}\right)^{v^h_c} \frac{1}{|A^i_{r,j}(b)|} \\
&\ge \frac{|A^i_{r,j}(b') \bigcap E^i_{r,j}(t)|}{|A^i_{r,j}(b)|} \left(1 - \frac{1}{d^i_j}\right)^{h-1} \\
&\ge g_{j-r} \left(1 - \frac{1}{d^i_j}\right)^{h-1} = q^i_{r,j}(h)
\end{aligned}$$

where, in the third line, we are using (the inductive assumption for) item 2, and in the inequality in the last line, we are using item 3.

Next, we prove the item 1. Assume $c^i_j(t,h)$ is the color proposed by $e^i_j(t,h)$ at phase i. Then, we first show that w. h. p. we have

$$\Pr\left[\mathrm{acc}^i_j(t,h) \mid c^i_j(t,h) \in A^i_{r,j}(b')\right] \sim q^i_{r,j}(h)$$

and hence

$$\Pr\left[\mathrm{acc}^i_j(t,h)\right] \sim \sum_{r=1}^{j} \frac{g_{j-r}|N^i_r|}{d^i_j} q^i_{r,j}(h)\,. \tag{3.1}$$

This is because

$$\begin{aligned}
\Pr\left[\mathrm{acc}^i_j(t,h) \mid c^i_j(t,h) \in A^i_{r,j}(b')\right] &= \Pr\left[\mathrm{new}_h \mid c^i_j(t,h) \in A^i_{r,j}(b)\right] \cdot \Pr\left[\mathrm{acc}^i_j(t,h) \mid \mathrm{new}_h\right] \\
&= p^i_{r,j}(t,h) \frac{q^i_{r,j}(h)}{p^i_{r,j}(t,h)} = q^i_{r,j}(h)
\end{aligned}$$

and thus

$$\begin{aligned}
\Pr\left[\mathrm{acc}_j^i(t,h)\right] &= \sum_{r=1}^{j} \Pr\left[c_j^i(t,h) \in A_{r,j}^i(b)\right] \cdot \Pr\left[\mathrm{acc}_j^i(t,h) \mid c_j^i(t,h) \in A_{r,j}^i(b)\right] \\
&= \sum_{r=1}^{j} \frac{A_{r,j}^i}{d_j^i} q_{r,j}^i(h) \\
&= \sum_{r=1}^{j} \frac{g_{j-r}|N_r^i|}{d_j^i} q_{r,j}^i(h)\,.
\end{aligned}$$

Now, let $X_{r,j}^i(t)$ be the number of colors from $E_{r,j}^i(t)$ which get used in phase i of round j which get used to color some edge incident on t. Then,

$$\begin{aligned}
\mathrm{E}[X_{r,j}^i(t)] &= E\left[\sum_{h=1}^{d_j^i} 1_{c_j^i(t,h)\in A_{r,j}^i(h),\, \mathrm{acc}_j^i(t,h)}\right] \\
&= \sum_{h=1}^{d_j^i} \Pr\left[c_j^i(t,h) \in A_{r,j}^i(h),\, \mathrm{acc}_j^i(t,h)\right] \\
&= \frac{A_{r,j}^i(h)}{d_j^i} q_{r,j}^i(h) \\
&= \frac{g_{j-r}|A_{r,j}^i(h)|}{d_j^i} \sum_{h=1}^{d_j^i} \left(1 - \frac{1}{d_j^i}\right)^{h-1} \\
&= g_{j-r}^2 |N_r^i| \left(1 - \frac{1}{\mathrm{e}}\right).
\end{aligned}$$

Thus, the expected size of $E_{r,j+1}^i(t)$ equals

$$|E_{r,j}^i(t)| - \mathrm{E}[X_{r,j}^i(t)] \sim g_{j-r}|N_r^i| - g_{j-r}^2|N_r^i| \left(1 - \frac{1}{\mathrm{e}}\right) = g_{j+1-r}|N_r^i|\,.$$

The w. h. p. proof is then standard. Also, a similar calculation, summing over round j edges incident on node b, gives $|A_{r,j+1}^i(b)| = g_{j-r+1}|N_r^i|$. This finishes the proof of item 1.

Finally, for the proof of item 2, notice that equation (3.1) shows that the probability that a round j edge gets colored at some phase i depends solely on i, j and the position of that edge among the other round j edges incident on the same top vertex which reach phase i. Note that we get this property in spite of the fact that the "natural probabilities" of success (namely the $p_{r,j}^i(t,h)$'s) depend crucially on the identities of the first h edges, since they depend on how the various $A_{r,j}^i(b)$ sets intersect. Using this property, the proof of item 2, for top vertices, is similar to the above calculations (for item 1) using equation (3.1), and, for bottom vertices, can be done in almost exactly the same way as in Round 1. So we omit the details here. □

3.1 Counting the number of colors used

Theorem 3.3 gives us the essential ingredients for analyzing the multi-round algorithm. So, next, we proceed to calculating the total number of colors used by the algorithm.

The number of edges (for each vertex) reaching round j of phase i is $\deg_i^j$, while the number of old colors of phase i (from rounds less than j) reaching round j of phase i is equal to the number of edges (for each vertex) reaching round $j-1$ of phase $i+1$, i. e., equal to $\deg_{j-1}^{i+1}$. Therefore, we have:

$$\forall i \geq 1, j \geq 2 : \quad |N_j^i| = \max\left\{0, d_j^i - d_{j-1}^{i+1}\right\}. \tag{3.2}$$

If this number were 0, we would simply not introduce any new colors. However, for the calculation of the total number of colors (Lemma 3.4), it is easier to drop the max with 0. This may cause us to undercount the number of colors if the second term in the max was negative.

For now, we assume that in each phase of each round we need to introduce some new colors. That is, we never have more old colors remaining from the previous rounds than the degree of the nodes in the current round. Thus our assumption can be stated as:

$$\forall i \geq 1, j \geq 2 : \; d_j^i \geq d_{j-1}^{i+1}. \tag{3.3}$$

From now onwards, we proceed with this assumption and bound the total number of colors used. We will try to prove the assumption in Section 3.1.1 by reducing it to the question of non-negativity of a certain sequence of numbers.

Lemma 3.4. *The total number of colors used is, w. h. p., $\sum_{j=1}^{K-1} r_j\Delta + S_K + |\mathrm{P}^\infty|$, where $S_K = \sum_{i=1}^{L} d_K^i$.*

Proof. From equation (3.2) and Assumption (3.3) we have

$$|N_j^i| = d_j^i - d_{j-1}^{i+1}. \tag{3.4}$$

Hence,

$$\begin{aligned}
\#\text{colors} &= \sum_{1\leq i\leq L, 1\leq j\leq K} |N_j^i| + |\mathrm{P}^\infty| \\
&= \sum_{j=1}^{K}\sum_{i=1}^{L}\left[d_j^i - d_{j-1}^{i+1}\right] + |\mathrm{P}^\infty| \\
&= \sum_{j=1}^{K-1} d_j^1 + \sum_{i=1}^{L} d_K^i + |\mathrm{P}^\infty| \\
&= \sum_{j=1}^{K-1} r_j\Delta + S_K + |\mathrm{P}^\infty|.
\end{aligned}$$

where, in the last line, we are using $d_j^1 = r_j\Delta$, which holds because the phase 1 degrees in round j are equal to $r_j\Delta$. □

The next step is to bound S_K:

Lemma 3.5.

$$S_K \le \frac{r_K \Delta}{\left(1-\frac{1}{\mathrm{e}}\right) g_{K-1}}\,.$$

Proof. From Theorem 3.3, we have for all $i \in [1, L-1]$, $j \in [1, K]$

$$d_j^i = \sum_{r=1}^{j} g_{j-r} |N_r^i|$$

and from equation (3.4)

$$d_j^{i+1} = d_{j+1}^i - |N_{j+1}^i| = \sum_{r=1}^{j} g_{j-r+1} |N_r^i|\,.$$

Hence,

$$\begin{aligned} d_j^{i+1} &\le \max_{0 \le \ell \le j-1} \left\{ \frac{g_{\ell+1}}{g_\ell} \right\} d_j^i \\ &= \max_{0 \le l \le j-1} \left\{ 1 - \left(1 - \frac{1}{\mathrm{e}}\right) g_\ell \right\} d_j^i \\ &= \left(1 - \left(1 - \frac{1}{\mathrm{e}}\right) g_{j-1}\right) d_j^i\,. \end{aligned}$$

Thus, letting $j = K$, and doing a simple induction (and recalling $d_K^1 = r_K \Delta$), we have

$$d_K^i \le \left(1 - \left(1 - \frac{1}{\mathrm{e}}\right) g_{K-1}\right)^{i-1} r_K \Delta\,.$$

Hence,

$$\begin{aligned} S_K &= \sum_{i=1}^{L} d_K^i \\ &\le \sum_{i \ge 1} \left(1 - \left(1 - \frac{1}{\mathrm{e}}\right) g_{K-1}\right)^{i-1} r_K \Delta \\ &= \frac{r_K \Delta}{1 - \left(1 - \left(1 - \frac{1}{\mathrm{e}}\right) g_{K-1}\right)} \\ &= \frac{r_K \Delta}{\left(1 - \frac{1}{\mathrm{e}}\right) g_{K-1}}\,. \end{aligned}$$

□

From the above lemma, and also noting that $\sum_{j=1}^{K} r_j = 1$ and recalling $\Delta = \omega(\log n)$, the total number of colors can be bounded as follows:

$$\frac{\#\text{colors}}{\Delta} \le 1 - r_K + \frac{r_K}{\left(1 - \frac{1}{\mathrm{e}}\right) g_{K-1}} + o(1)\,. \tag{3.5}$$

Note that so far we have not yet defined the values for $r_1, r_2, \ldots, r_K$, and the above results were independent of the exact values of these parameters. But now, to get the final bound, we need to determine the values for these parameters. We do that in the following definition.

Definition 3.6. Define $\{r_j\}_{1\le j\le K}$ as follows:

$$\begin{aligned} r_1 &= \frac{1}{\sum_{j=0}^{K-1} g_j}, \\ r_j &= g_{j-1} r_1 \ (\forall 1 < j \le K). \end{aligned}$$

Notice that these values do satisfy $\sum_{j=1}^{K} r_j = 1$.

Using the above definition for $\{r_j\}_{1\le j\le K}$, from equation (3.5) we have

$$\frac{\#\text{colors}}{\Delta} \le 1 - g_{K-1} r_1 + \frac{r_1}{\left(1-\frac{1}{\mathrm{e}}\right)} + o(1). \tag{3.6}$$

Hence, to bound the total number of colors used, we need to bound the sequence $\{g_\ell\}_{\ell\ge 0}$. We do so in the following lemma:

Lemma 3.7. *For any $\ell \ge 1$, we have*

$$\frac{1}{5\left(1-\frac{1}{\mathrm{e}}\right)\ell} \le g_\ell \le \frac{1}{\left(1-\frac{1}{\mathrm{e}}\right)\ell}.$$

Proof. We prove both bounds by induction. We start with proving the upper-bound on g_ℓ. For $\ell = 1$, we have $g_1 = 1/\mathrm{e} \le 1/(1-1/\mathrm{e})$. Now, for the inductive step, assume the upper-bound is true for ℓ; we will prove it for $\ell+1$. Since $\ell \ge 1$, we have $g_\ell \le 1/\mathrm{e}$. Also, the function $f(x) = x - (1-1/\mathrm{e})x^2$ is monotone increasing over $[0, 1/\mathrm{e}]$. Hence,

$$\begin{aligned} g_{\ell+1} = g_\ell - \left(1-\frac{1}{\mathrm{e}}\right) g_\ell^2 &\le \frac{1}{\left(1-\frac{1}{\mathrm{e}}\right)\ell} - \left(1-\frac{1}{\mathrm{e}}\right)\frac{1}{\left(1-\frac{1}{\mathrm{e}}\right)^2 \ell^2} \\ = \frac{1}{1-1/\mathrm{e}}\left(\frac{1}{\ell} - \frac{1}{\ell^2}\right) &= \frac{1}{1-1/\mathrm{e}}\left(\frac{\ell-1}{\ell^2}\right) \le \frac{1}{\left(1-\frac{1}{\mathrm{e}}\right)(\ell+1)}. \end{aligned}$$

This finishes the proof for the upper-bound. For the lower bound, we do another induction. For $\ell = 1$,

$$g_1 = \frac{1}{\mathrm{e}} \ge \frac{1}{5\left(1-\frac{1}{\mathrm{e}}\right)}.$$

For the inductive step, we start with the following obvious inequality:

$$\frac{\ell}{\ell+1} \ge \frac{1}{5} \Rightarrow \frac{1}{\ell} - \frac{1}{\ell+1} = \frac{1}{\ell(\ell+1)} \ge \frac{1}{5\ell^2}.$$

Thus,

$$g_{\ell+1} = g_\ell - \left(1-\frac{1}{\mathrm{e}}\right) g_\ell^2 \ge \frac{1}{5\left(1-\frac{1}{\mathrm{e}}\right)\ell} - \left(1-\frac{1}{\mathrm{e}}\right)\left(\frac{1}{5\left(1-\frac{1}{\mathrm{e}}\right)\ell}\right)^2 \ge \frac{1}{5\left(1-\frac{1}{\mathrm{e}}\right)(\ell+1)}$$

which finishes the proof. □

Proposition 3.8. *If Assumption (3.3) holds, the K-round online algorithm achieves a competitive ratio of* $1+O(1/\log K)+o(1)$ *for any graph with* $\Delta=\omega(\log n)$.

Proof. From Lemma 3.7 and Definition 3.6, we get

$$\frac{1}{1+5\left(1-\frac{1}{\mathrm{e}}\right)H_{K-1}} \le r_1 \le \frac{1}{1+\left(1-\frac{1}{\mathrm{e}}\right)H_{K-1}}$$

where $H_{K-1}=\sum_{j=1}^{K-1} 1/j \simeq \ln(K-1)$ is the $(K-1)^{st}$ harmonic number. Thus, from equation (3.6), we see that the competitive ratio of the algorithm with K rounds, converges, as $1+O(1/\log K)+o(1)$, to 1 when K increases. Thus, with $\Delta=\omega(\log n)$, this algorithm achieves, w. h. p., a coloring with $\Delta+o(\Delta)$ number of colors (for K large enough, e. g., $K=O(\log(\Delta/\log n))$). □

3.1.1 The non-negativity assumption

This concludes the analysis under the assumption (3.3) that $\forall i\in[1,L],\ j\in[1,K]$, the number of old colors is not more that the degree at phase i, round j. In this section we attempt to prove this assumption. We reduce the question to proving the non-negativity of the following sequence of numbers.

Definition 3.9. The sequence $\{c_\ell\}_{\ell\ge1}$ is defined recursively as follows:

$$c_\ell = g_\ell - \sum_{j=1}^{\ell-1} c_j g_{\ell-j}\,.$$

The sequence $|\mathrm{N}_j^i|$ follows a recurrence based on the sequence we just defined.

Lemma 3.10. *For any* $1\le i<L$, $1\le j\le K$, *we have*

$$|N_j^{i+1}| = \sum_{\ell=1}^{j} c_\ell |N_{j-\ell+1}^i|\,.$$

Proof. We have $\deg_j^{i+1}=\sum_{\ell\le j}\mathrm{N}_j^i g_{j-\ell+1}$, and the number of old colors from phase $i+1$, round $\ell<j$ reaching round j is $\mathrm{N}_r^{i+1} g_{j-\ell}$. A simple induction leads to the proof of the lemma. □

Corollary 3.11. *Assuming* $c_\ell\ge0$, $(\forall\ell\in[1,K])$, *with the* r_j*'s as defined in Definition 3.6, we have* $|N_j^i|\ge0$ *(*$\forall i\in[1,L], j\in[1,K]$*).*

Proof. From Definition 3.6, we have: $|N_1^1|=r_1\Delta$ and $|N_j^1|=0$ for any $1<j\le K$, which shows claim is true for $j=1$. Then, the proof can be completed by a straightforward induction on j. We omit the details. □

This corollary shows that it is sufficient (for non-negativity of all $|N_j^i|$'s ($i\in[1,L],\ j\in[1,K]$)) to prove that c_ℓ is non-negative for $1\le\ell\le K$. If K is a small constant (e. g., say 3,4,5) , this can be proved by direct calculation of the values c_ℓ up to $\ell=K$. Our computer simulation results, presented in Figure 6, also show that c_ℓ is non-negative up to very large values of K. However, we don't have a formal proof to show that all c_ℓ's (i. e., $\forall\ell\ge1$) are non-negative. However, based on the presented computer simulation results and our current understanding, we give the following conjecture which we believe to be true.

Conjecture 3.12. *Defining* $\{g_\ell\}_{\ell\geq 0}$ *as in Definition 3.1, and* $\{c_\ell\}_{\ell\geq 1}$ *as in Definition 3.9, we have* $\forall\,\ell \geq 1$*:* $c_\ell \geq 0$*.*

A proof of this conjecture completes the analysis of the near-optimal $\Delta + o(\Delta)$ coloring. Alternatively, below we present a simple way to exactly compute the competitive ratio for any K, given we know the values c_ℓ are non-negative for ℓ up to K.

3.1.2 Computing the number of colors

We show how one can exactly compute the number of colors used by the K-round algorithm, assuming the non-negativity assumption holds for the first K values of $\{c_\ell\}_{\ell\geq 1}$. As proved in Lemma 3.4, the total number of colors used by the algorithm is, up to lower order terms, $(1-r_K)\Delta + S_K$. So, we need to compute $S_K = \sum_{i\geq 1} d^i_K$. We have, from Theorem 3.3,

$$d^i_K = \sum_{\ell=0}^{K-1} g_\ell |\mathrm{N}^i_{j-\ell}|\,.$$

Hence,

$$S_K = \sum_{i\geq 1} d^i_K = \sum_{i\geq 1}\sum_{\ell=0}^{K-1} g_\ell |\mathrm{N}^i_{K-\ell}| = \sum_{\ell=0}^{K-1} g_\ell \sum_{i\geq 1} |\mathrm{N}^i_{K-\ell}|\,.$$

Define

$$A_j = \sum_{i\geq 1} |\mathrm{N}^i_j|\,.$$

Then, from the recursion in Lemma 3.10, one can easily observe

$$A_1 = \frac{r_1}{\left(1-\frac{1}{\mathrm{e}}\right)}\,,$$

$$A_j = \sum_{\ell=1}^{j} c_{j-\ell+1} A_\ell\,.$$

So, one can first recursively compute A_j's using the above equation, and then S_K can be computed as

$$S_K = \sum_{j=1}^{K} g_{K-j} A_j$$

which then allows computation of the total number of colors used by the K-round algorithm. Using this method, we can, for instance, compute the exact competitive ratio for the K-round algorithm for small constants, such as 3,4,5:

Number of Rounds	#colors/Δ
1	1.6
2	1.43
3	1.35
4	1.30
5	1.26

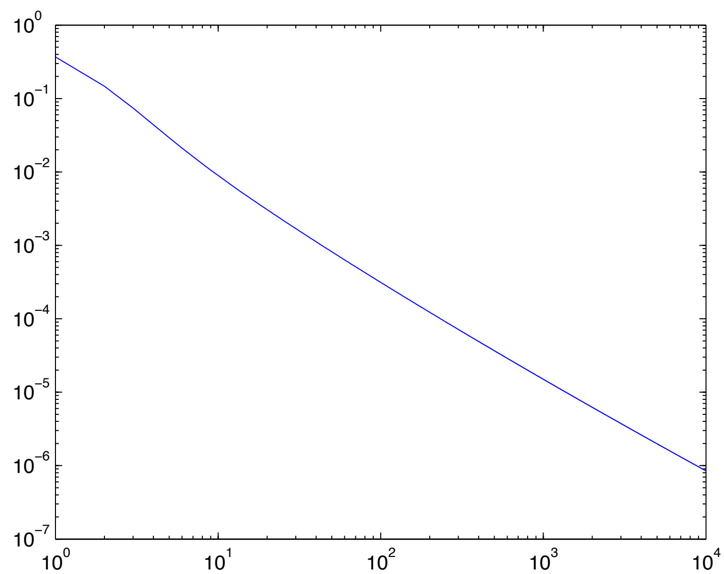

Figure 6: $\log\log$ plot of c_ℓ vs. ℓ

4 Conclusions and open questions

The following are the three main open questions:

Calculations for the K-round algorithm Prove Conjecture 3.12 or find an alternate way of calculating the number of colors used in the k-round algorithm.

Analyze the Random algorithm The Random algorithm is possibly the simplest randomized algorithm: It starts with a palette of a sufficient number of colors. When an edge (b,t) arrives, it is given a random color from the palette which does not conflict with already arrived neighboring edges on either side. A variant of this algorithm is studied in [1], proving that, it uses only $\Delta + o(\Delta)$ colors, when the degree of the (multi-)graph is $\omega(n^2)$.

One may think of this process as the following propose-accept process (similar to our algorithm): b picks a random color which has not been used to color previously arrived edges incident on b, and proposes this color for (b,t). If no edge incident on t has used this color then (b,t) is colored with this color, else b repeats with a new random choice from the palette, until success. Our algorithm can be considered a variant of Random which breaks up the palette into several disjoint palettes, and the edges into K rounds. It allows each color to be tried up to K times at each bottom vertex, once for an edge from each round. By doing this we manage the correlations between available color sets. The open question is to analyze the Random algorithm in random or adversarial arrival orders and prove that, with

$\Delta = \omega(\text{polylog}(n))$, a palette of size $\Delta + o(\Delta)$ suffices.

Adversarial order Find an algorithm with an optimal competitive ratio in the adversarial order.

References

[1] GAGAN AGGARWAL, RAJEEV MOTWANI, DEVAVRAT SHAH, AND AN ZHU: Switch scheduling via randomized edge coloring. In *Proc. 44th FOCS*, pp. 502–512. IEEE Comp. Soc. Press, 2003. [doi:10.1109/SFCS.2003.1238223] 568, 569, 571, 592

[2] AMOTZ BAR-NOY, RAJEEV MOTWANI, AND JOSEPH NAOR: The greedy algorithm is optimal for on-line edge coloring. *Inform. Process. Lett.*, 44(5):251–253, 1992. [doi:10.1016/0020-0190(92)90209-E] 569

[3] RICHARD COLE, KIRSTIN OST, AND STEFAN SCHIRRA: Edge-coloring bipartite multigraphs in $O(E \log D)$ time. *Combinatorica*, 21(1):5–12, 2001. [doi:10.1007/s004930170002] 568

[4] DEVDATT DUBHASHI, DAVID A. GRABLE, AND ALESSANDRO PANCONESI: Near-optimal, distributed edge colouring via the nibble method. *Theoret. Comput. Sci.*, 203(2):225–251, 1998. Preliminary version in ESA'95. [doi:10.1016/S0304-3975(98)00022-X] 571

[5] DEVDATT DUBHASHI AND ALESSANDRO PANCONESI: *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2009. 572, 575, 576, 578

[6] DAVID A. GRABLE AND ALESSANDRO PANCONESI: Nearly optimal distributed edge coloring in $o(\log\log n)$ rounds. *Random Structures Algorithms*, 10(3):385–405, 1997. Preliminary version in SODA'97. [doi:10.1002/(SICI)1098-2418(199705)10:3<385::AID-RSA6>3.0.CO;2-S] 571

[7] IAN HOLYER: The NP-completeness of edge-coloring. *SIAM J. Comput.*, 10(4):718–720, 1981. [doi:10.1137/0210055] 568

[8] ALESSANDRO PANCONESI AND ARAVIND SRINIVASAN: Randomized distributed edge coloring via an extension of the Chernoff-Hoeffding bounds. *SIAM J. Comput.*, 26(2):350–368, 1997. Preliminary version in PODC'92. [doi:10.1137/S0097539793250767] 568, 571

[9] VADIM G. VIZING: On an estimate of the chromatic class of a p-graph. *Metody Diskret. Analiz.*, 3:25–30, 1964. 568

AUTHORS

Bahman Bahmani
Ph. D. Student
Stanford University
bahman@stanford.edu

Aranyak Mehta
Research Scientist
Google Inc., Mountain View, CA
aranyak@google.com

Rajeev Motwani (1962 - 2009)
Former Professor
Stanford University
rajeev@cs.stanford.edu

ABOUT THE AUTHORS

BAHMAN BAHMANI is a Ph. D. student at Stanford University supported by the William R. Hewlett Stanford Graduate Fellowship. His research interests are in algorithmic and architectural aspects of web and large data applications. His Ph. D. advisor was Rajeev Motwani. After Rajeev's passing, Ashish Goel and Prabhakar Raghavan became his advisor and coadvisor. He is a recipient of the Yahoo Key Scientific Challenges Award for his contributions to the area of search technologies.

ARANYAK MEHTA is a Research Scientist at Google Research, based in Mountain View, CA. He received his Ph. D. from Georgia Tech in 2005, advised by Dick Lipton and Vijay Vazirani, with a thesis on algorithmic game theory. He received a B. Tech from I. I. T. Bombay in 2000, where he started research in theoretical computer science with the support of Milind Sohoni and Sundar Vishwanathan. His interests lie in online and approximation algorithms, auction and mechanism design, and in algorithmic and auction theoretic applications in industry. He grew up in Bombay, inevitably becoming a fan of cricket and Bollywood, and currently enjoys living in the San Francisco Bay Area.

RAJEEV MOTWANI was born on March 24, 1962 in Jammu, India. He died on June 5, 2009. He received a B. Tech degree in Computer Science from I. I. T. Kanpur in 1983 and a Ph. D. in Computer Science from University of California at Berkeley in 1988 under the supervision of Richard Karp. The list of his research interests is long and eclectic, and includes graph theory, approximation algorithms, randomized algorithms, online algorithms, complexity theory, web search and information retrieval, databases, data mining, computational drug design, robotics, streaming algorithms, and data privacy. He received the Gödel Prize in 2001 for his research on probabilistically checkable proofs and hardness of approximation. Dr. Motwani successfully spanned both theory and practice, being an early advisor and supporter of Google, in addition to many other successful startups and venture firms in Silicon Valley.