5-1-2019

Convergence Times of Decentralized Graph Coloring Algorithms

Paul B. de Supinski
Dartmouth College

TR2019-864

Convergence Times of Decentralized Graph Coloring Algorithms

Paul de Supinski

Spring 2019

Abstract

Ordinary graph coloring algorithms are nothing without their calculations, memorizations, and inter-vertex communications. We investigate a class of ultra simple algorithms which can find $(\Delta+1)$-colorings despite drastic restrictions. For each procedure, conflicted vertices randomly recolor one at a time until the graph coloring is valid. We provide an array of run time bounds for these processes, including an $O(n \log \Delta)$ bound for a variant we propose, and an $O(n\Delta)$ bound which applies to even the most adversarial scenarios.

1 Introduction

Even among timeless problems, graph coloring stands out as a gem. Not only is it beautiful, it is a natural model for applications from taxi scheduling to compiler register allocation [12]. Formally, a k-coloring of an undirected graph $G = (V, E)$ is a function $\chi : V \to [k]$, where $[k] = \{1, 2, \ldots, k\}$. We call χ **valid** iff there is no edge $\{u, v\} \in E$ such that $\chi(u) = \chi(v)$.

Traditionally, graph coloring has been viewed from the point of view of a centralized entity. For such an entity, finding valid (Δ+1)-colorings is trivial, where Δ is the maximum number of edges attached to any single vertex. In particular, this entity could simply iterate through the vertices and validly color them one by one, because there must always be at least one color not used by a vertex's neighbors.

Instead, suppose we abandon these centralized, sequential means of finding a (Δ+1)-coloring, and think of the vertices themselves as the acting entities. Additionally, suppose that the vertices cannot communicate, cannot perform any calculations, have no memory, and only know *whether* they are conflicted (but not their neighbors' actual colors). These constraints restrict us to only the most stringently and purely **decentralized** methods which, if fast, could be ideal for systems with dynamic structure, constrained resources, or lack of coordination. For example, Bhartia et al. studied an application to Wi-Fi channel selection, in which nearby routers need to operate on different channels, otherwise suffer interference [4].

Without the usual means of finding a valid (Δ+1)-coloring, one recourse would be to randomly recolor random vertices (whether on not they are conflicted) until the coloring is valid. However, this algorithm takes exponential time in even the simplest graphs.

A better approach, the one proposed by Bhartia et al., would be to randomly recolor random *conflicted* vertices until the coloring is valid. We call this procedure **non-persistent coloring**, because a different vertex could recolor at each timestep.

Algorithm 1 Non-persistent coloring, in general form

Input: A graph $G = (V, E)$
Output: A valid (Δ+1)-coloring of G

procedure NonPersistentColor(G)
  Initialize $\chi \leftarrow \chi_0$ ▷ Could be random or adversarial
  while χ is not valid **do**
    $v \leftarrow$ a conflicted vertex ▷ Could be random or adversarial
    Randomly set $\chi(v) \in [\Delta + 1]$ ▷ *Always* random
  end while
  return χ ▷ We use $\tau(G)$ to denote the total number of recolorings
end procedure

This algorithm can be faithfully implemented by having each vertex wait for a random amount of time before each recolor, and hence is truly decentralized [4]. (In particular, the distribution of times needs to be continuous and memoryless.)

As both an example and a powerful theorem, we begin by analyzing the run time of non-persistent coloring in the special case of cliques, graphs such that every two vertices share an edge.

Theorem 1.1. *Let K_n be the clique containing n vertices, and let $H_n = \sum_{i=1}^{n} \frac{1}{i}$, the n^{th} harmonic number. Then, no matter the initial coloring or the choices of conflicted vertices, $\mathbb{E}[\tau(K_n)] \leq nH_n$. If the initial coloring is random, then $\mathbb{E}[\tau(K_n)] = nH_n - n = \Theta(n \log \Delta)$.*

Proof: Observe that NonPersistentColor(K_n) terminates exactly when each color has been chosen at least once. This is because

1. The coloring is valid if all n colors are present in the graph.
2. The coloring is not valid if fewer than n distinct colors are present in the graph.
3. Colors never disappear from the graph once they are present.

Hence, we can couple NonPersistentColor(K_n) with the classic coupon-collector problem, in which a collector randomly chooses one of n coupons with replacement until he has at least one of each coupon. To do so, we simply recolor the next vertex to the color c whenever the collector draws the coupon c, for $c \in [n]$. Under this coupling, the above facts guarantee that NonPersistentColor(K_n) never takes longer than the coupon-collector. Furthermore, when χ_0 is random, we can use the first n coupon draws as the initial colors, and then proceed with the coupling as before. In that case, NonPersistentColor(K_n) and the coupon-collector finish simultaneously.

Simple calculation shows that the collector is expected to draw exactly nH_n coupons before completing his collection. Hence, $\mathbb{E}[\tau(K_n)] \leq nH_n$. In the random start case, we have $\mathbb{E}[\tau(K_n)] = nH_n - n$, because we only count *re*colorings. It is well-known that $H_n = \Theta(\log n)$. ■

So, pleasantly, non-persistent coloring is a generalization of the coupon-collector problem, and we will later explore relationships to the birthday and Monty Hall paradoxes. This motivates the following conjecture.

Conjecture 1.2. *The worst case expected run time of random start, random order non-persistent coloring is $O(n \log \Delta)$. Specifically, for any graph $G = (V, E)$, we have $\mathbb{E}[\tau(G)] \leq nH_{\Delta+1}$.*

Despite its simplicity, non-persistent coloring has proven difficult for us to analyze, so we propose a tweaked version called **persistent coloring**, which we now understand well. Persistent coloring is identical to non-persistent coloring, but the chosen vertex recolors *until* it is not conflicted.

Algorithm 2 Persistent coloring, in general form

Input: A graph $G = (V, E)$

Output: A valid (Δ+1)-coloring of G

```
1: procedure PersistentColor(G)
2:     Initialize χ ← χ_0                          ▷ Could be random or adversarial
3:     while χ is not valid do
4:         v ← a conflicted vertex                 ▷ Could be random or adversarial
5:         while v is conflicted do
6:             Randomly set χ(v) ∈ [Δ + 1]          ▷ Always random
7:         end while
8:     end while
9:     return χ        ▷ We use τ(G) to denote the total number of recolorings
10: end procedure
```

We generally think of PersistentColor as using a permutation of the vertices $\sigma : [n] \to V$, such that the vertices recolor (if necessary) in the ascending σ order during line 4 of Algorithm 2. This will be important in subsequent sections.

We subdivide non-persistent coloring and persistent coloring into four variants each. If χ_0 is random (line 2), then we say they use a **random start**. Otherwise, we say they use an **adversarial start**. If the algorithms select conflicted vertices randomly (line 4), then we say they use a **random order**. Otherwise, we say they use an **adversarial order**, and we allow the adversary to make his choices as the algorithm proceeds.

The focus of this thesis is bounding the worst case expected run times of non-persistent coloring and persistent coloring on graphs with n vertices and max degree Δ. Specifically, if $\mathcal{G}(n, \Delta)$ is the set of all graphs with n vertices and max degree Δ, we seek to asymptotically bound

$$\max_{G \in \mathcal{G}(n,\Delta)} \mathbb{E}\left[\tau(G)\right]$$

for each variant.

Most importantly, in order of appearance, we

1. Propose the persistent coloring algorithm
2. Bound the worst case expected run time of adversarial start, adversarial order non-persistent coloring to $\Theta(n\Delta)$
3. Bound the worst case expected run time of random start, random order persistent coloring to $\Theta(n \log \Delta)$
4. Bound the worst case expected run time of adversarial start persistent coloring to $\Theta(n\Delta)$

Although there is some prior understanding of slightly less restrictive algorithms ([4, 6, 7, 11]), we believe these results are novel.

2 Warmup (Blank Persistent)

We begin by analyzing persistent coloring for a case slightly outside the rules—when $\mathcal{X}_0$ starts out *blank*. That is, we start with $\mathcal{X}_0(v) = 0$ for all $v \in V$, where the "color" 0 represents blankness. We then update our definition such that a vertex is conflicted iff it has the same color as one of its neighbors *or* it is blank. In effect, the algorithm becomes the following: while there is some blank vertex, pick a random one and recolor it until it is not conflicted (and never make vertices blank again).

As a warmup, we prove following theorem, which has obvious similarity to Conjecture 1.2.

Theorem 2.1. *The worst case expected run time of blank start, random order persistent coloring is $O(n \log \Delta)$. Specifically, for any graph $G = (V, E)$, we have $\mathbb{E}[\tau(G)] \leq nH_{\Delta+1}$.*

2.1 Notation

We take this opportunity to introduce notation which we will utilize throughout our analysis of persistent coloring.

We use D as shorthand for $\Delta + 1$, and $\Gamma(v)$ to denote the neighborhood of v, *not* including v. Recall that we generally think of persistent coloring as using a permutation of the vertices $\sigma : [n] \to V$, such that the vertices recolor (if necessary) in the ascending σ order during line 4 of Algorithm 2. We can then define a vertex's local rank such that

$$\operatorname{rank}(v) \coloneqq D - \left|\left\{u \in \Gamma(v) \mid \sigma^{-1}(u) < \sigma^{-1}(v)\right\}\right|.$$

For example, if v is the k^{th} vertex to recolor in its neighborhood, then $\operatorname{rank}(v) = D - k + 1$. Note that rank is flipped from σ, but this will tend to simplify our calculations.

We use $\operatorname{recolors}(v)$ to denote the total number of times v eventually recolors, and $\operatorname{free}(v)$ to denote the number of colors not used by $\Gamma(v)$ when v begins recoloring. Finally, we say that a vertex becomes **fixed** when its turn to recolor finishes.

2.2 Upper Bound

We now prove our theorem.

Proof of Theorem 2.1: Let $G = (V, E)$ be an arbitrary graph with n vertices and max degree Δ, and $v \in V$. Let $d = \deg(v)$. Observe that $\operatorname{rank}(v)$ is uniformly random in $\{D - d, D - d + 1, \ldots, D\}$. Hence, we have

$$\mathbb{E}[\operatorname{recolors}(v)] = \frac{1}{d+1} \sum_{r=D-d}^{D} \mathbb{E}[\operatorname{recolors}(v) \mid \operatorname{rank}(v) = r]. \tag{1}$$

Notice that $\operatorname{recolors}(v)$ is a geometric random variable with probability $\operatorname{free}(v)/D$. But $\operatorname{free}(v) \geq \operatorname{rank}(v)$, because the worst case is that each of v's fixed neighbors has a distinct color, and the remaining neighbors are blank. Hence, we have $\mathbb{E}[\operatorname{recolors}(v) \mid \operatorname{rank}(v) = r] \leq D/r$. Now, we manipulate eq. (1):

$$\frac{1}{d+1}\sum_{r=D-d}^{D}\mathbb{E}\left[\text{recolors}(v) \mid \text{rank}(v)=r\right] \leq \frac{1}{d+1}\sum_{r=D-d}^{D}\frac{D}{r} \tag{2}$$

$$= \frac{D}{d+1}\sum_{r=D-d}^{D}\frac{1}{r} \tag{3}$$

$$\leq \frac{D}{d+1}\sum_{r=D-d}^{D}\frac{1}{r} \tag{4}$$

Notice that eq. (4) maximizes when $d = \Delta$. This is because we can think of eq. (4) as D times the average of $\left\{\frac{1}{D-d}, \frac{1}{D-d+1}, \dots, \frac{1}{D}\right\}$. Hence, because $\frac{1}{D-(d+1)}$ is greater than the max of that set, increasing d by one increases the average. In total, we have

$$\mathbb{E}\left[\text{recolors}(v)\right] \leq \frac{D}{d+1}\sum_{r=D-d}^{D}\frac{1}{r} \leq \frac{D}{\Delta+1}\sum_{r=1}^{D}\frac{1}{r} = H_D.$$

So $\mathbb{E}\left[\text{recolors}(v)\right] = O\left(\log \Delta\right)$. Because v was arbitrary, by linearity of expectation, $\mathbb{E}\left[\tau(G)\right] = O\left(n \log \Delta\right)$. ■

The ease of this warmup should make our other $O\left(n \log \Delta\right)$ conjectures and theorems significantly more believable. However, this approach of bounding $\mathbb{E}\left[D/\text{free}(v) \mid \text{rank}(v) = r\right]$ *cannot* be used to show that any of the main variants takes $O\left(n \log \Delta\right)$ time. This is due to the birthday paradox.

Analysis of the birthday paradox shows that there is an $\Omega(1)$ chance that there are no duplicates among the first $\sqrt{n}$ random selections with replacement from a group of n items [13]. In the random start persistent coloring case, this means that if $\text{rank}(v) \leq \sqrt{n}$ in a clique, then there is an $\Omega(1)$ chance that $\text{free}(v) = 1$ and so we would expect to recolor v a total of n times, if it ever does recolor. In other words, whereas the unfixed neighbors of v are all blank in our warmup, in the random start case, the unfixed neighbors can also use up valuable colors. This analysis would seem to make the worst case $\Omega\left(n\sqrt{\Delta}\right)$, but we are disregarding the probability that a vertex may not even need to recolor.

That is, there is a proof similar to the warmup's which shows that random start, random order persistent coloring takes $O\left(n\sqrt{\Delta}\right)$ time, but achieving the tight $O\left(n \log \Delta\right)$ bound requires a better technique.

3 Non-persistent Coloring

Before we return to persistent coloring, we provide general bounds which apply to all non-persistent coloring variants.

3.1 General Upper Bound

As suggested by Table 1, our primary contribution to the analysis of non-persistent coloring is an all-purpose upper bound of $O\left(n\Delta\right)$:

Theorem 3.1. *The worst case expected run time of adversarial start, adversarial order non-persistent coloring is $O(n\Delta)$. Specifically, for any graph G with n vertices and max degree Δ, we have $\mathbb{E}[\tau(G)] \leq (n-1)(\Delta+1)$.*

This bound is fairly trivial for persistent coloring, because there is always at least a $1/D$ chance that the next recolor permanently satisfies the chosen vertex. However, in non-persistent coloring, there is no similar concept of vertices becoming fixed. Instead, we analyze the rate at which non-persistent coloring drifts toward convergence. We will need the following theorem.

Lemma 3.2 (Infinite linearity of expectation [14])**.** *Let $Z_1, Z_2, \ldots$ be random variables. If $\sum_{t=1}^{\infty} \mathbb{E}[|Z_t|]$ converges, then*

$$\mathbb{E}\left[\sum_{t=1}^{\infty} Z_t\right] = \sum_{t=1}^{\infty} \mathbb{E}[Z_t].$$

This will enable us to prove the following theorem, which bears similarity to Wald's Equation. With some assumptions, Wald's Equation says that for a counting random variable T, $\mathbb{E}\left[\sum_{t=1}^{T} Z_t\right] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}[Z_t]\right]$ [17]. However, Wald's Equation requires that each Z_t be independent of the event $T \geq t$, a condition our Z_t's will not satisfy. Instead we prove a suitably adjusted version of Wald's Equation:

Theorem 3.3. *Let φ be a real-valued function of colorings for a graph G such that $\mathcal{X}$ is valid iff $\varphi(\mathcal{X}) = \lambda$, for some constant λ. Let $\mathcal{X}_t$ be the state of $\mathcal{X}$ after t recolorings. If*

$$\mathbb{E}\left[|\lambda - \varphi(\mathcal{X}_{t-1})| - |\lambda - \varphi(\mathcal{X}_t)| \;\middle|\; \mathcal{X}_{t-1} \text{ invalid}\right] \geq C$$

for some positive constant C, then $\mathbb{E}[\tau(G)] \leq \mathbb{E}[|\lambda - \varphi(\mathcal{X}_0)|]/C$.

Proof: For each $t \in \mathbb{Z}^+$, let

$$Z_t := \begin{cases} |\lambda - \varphi(\mathcal{X}_{t-1})| - |\lambda - \varphi(\mathcal{X}_t)| & \tau(G) \geq t \\ 0 & \text{otherwise.} \end{cases}$$

By assumption, $\varphi\left(\mathcal{X}_{\tau(G)}\right) = \lambda$. Hence, $\sum_{t=1}^{\tau(G)} Z_t = |\lambda - \varphi(\mathcal{X}_0)|$. This gives us

$$\mathbb{E}[|\lambda - \varphi(\mathcal{X}_0)|] = \mathbb{E}\left[\sum_{t=1}^{\tau(G)} Z_t\right] = \mathbb{E}\left[\sum_{t=1}^{\infty} Z_t \cdot \mathbb{1}_{\{\tau(G) \geq t\}}\right]. \tag{5}$$

To apply Lemma 3.2, we need $\sum_{t=1}^{\infty} \mathbb{E}\left[\left|Z_t \cdot \mathbb{1}_{\{\tau(G)\geq t\}}\right|\right]$ to converge. Observe that φ must be bounded, because it is real-valued and there are only finitely many possible colorings of G. Thus, because λ and C are just constants, $|Z_t| \leq \rho$ for some constant ρ. Hence,

$$\begin{aligned} \sum_{t=1}^{\infty} \mathbb{E}\left[\left|Z_t \cdot \mathbb{1}_{\{\tau(G)\geq t\}}\right|\right] &= \sum_{t=1}^{\infty} \mathbb{E}\left[|Z_t| \cdot \mathbb{1}_{\{\tau(G)\geq t\}}\right] &(6)\\ &\leq \rho \sum_{t=1}^{\infty} \mathbb{E}\left[\mathbb{1}_{\{\tau(G)\geq t\}}\right] &(7)\\ &= \rho \cdot \mathbb{E}[\tau(G)] &(8) \end{aligned}$$

Trivially, we can upper-bound $\mathbb{E}[\tau(G)]$ by nD^n, because at worst we need to select the lone satisfying color for n consecutive vertices, which can be cast as a geometric random variable with probability $\left(\frac{1}{D}\right)^n$ that uses at most n recolors per trial. So our sum indeed converges. Thus, we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{\infty} Z_t \cdot \mathbb{1}_{\{\tau(G)\geq t\}}\right] &= \sum_{t=1}^{\infty} \mathbb{E}\left[Z_t \cdot \mathbb{1}_{\{\tau(G)\geq t\}}\right] && \textit{Lemma } 3.2 \qquad (9)\\
&= \sum_{t=1}^{\infty} \mathbb{E}\left[Z_t \mid \tau(G) \geq t\right] \cdot \Pr\left[\tau(G) \geq t\right] && (10)\\
&\geq C \sum_{t=1}^{\infty} \Pr\left[\tau(G) \geq t\right] && (11)\\
&= C \cdot \mathbb{E}\left[\tau(G)\right] && (12)
\end{aligned}$$

With eq. (5), we have $\mathbb{E}[\tau(G)] \leq \mathbb{E}[|\lambda - \varphi(\mathcal{X}_0)|] / C$. ■

Examples clarify this seemingly opaque theorem. A natural choice would be for $\varphi(\mathcal{X})$ to be the number of conflicted edges in G under $\mathcal{X}$. In that case $\lambda = 0$, because $\mathcal{X}$ is valid iff there are no conflicted edges. This was the approach Bhartia et al. took, because one can easily show that the expected number of conflicted edges decreases by $1/D$ with each recoloring. Because, with a random initial coloring, the expected number of initial conflicted edges is $O(n)$, this yields an $O(n\Delta)$ bound (although they did not make explicit use of Theorem 3.3). Unfortunately, this particular φ only produces an $O(n\Delta^2)$ bound in the adversarial start case, where there could be $\Omega(n\Delta)$ conflicted edges initially.

The other obvious choice is for $\varphi(\mathcal{X})$ to be the number of conflicted vertices in G under $\mathcal{X}$. Again, $\lambda = 0$. However, we can concoct examples where we would actually expect $\varphi(\mathcal{X}_t)$ to increase, given an adversarial selection. For example, if we recolor v in Figure 1, the number of conflicted vertices increases (additively) by $1/4$, on average.

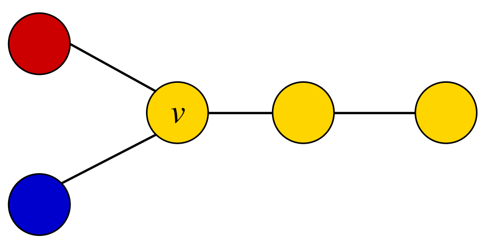

Figure 1: An example of a graph in which the number of conflicted vertices would be expected to increase, given an adversarial selection.

So the two natural choices of φ will not work for proving the desired $O(n\Delta)$ bound when non-persistent coloring uses an adversarial start and an adversarial order. Instead, we introduce a new notion, where $\varphi(\mathcal{X})$ is the number of **color components** of G. To do so, we define an equivalence relation $\sim_{\mathcal{X}}$, such that $v \sim_{\mathcal{X}} u$ iff there is some path $(v = v_0, v_1, \ldots, v_l = u)$ such that $\mathcal{X}(v_i) = \mathcal{X}(v_j)$ for all $0 \leq i, j \leq l$. Then the color component of v is its equivalence class $[v]_{\mathcal{X}}$. For example, in Figure 1, there are three color components.

Observe that $\mathcal{X}$ is valid iff $[v]_{\mathcal{X}} = \{v\}$ for all $v \in V$, i.e. when $\varphi(\mathcal{X}) = n$. This helps us prove our $O(n\Delta)$ bound.

Proof of Theorem 3.1: Let $G = (V, E)$ be an arbitrary graph with n vertices and max degree Δ. For each coloring $\mathcal{X}$ of G, let $\varphi(\mathcal{X}) = |\{[v]_\mathcal{X} \mid v \in V\}|$, the number of color components of G under $\mathcal{X}$.

Suppose we recolor an arbitrary conflicted vertex v at time t (so that $\mathcal{X}_{t-1}(v)$ is the old color and $\mathcal{X}_t(v)$ is the new color). It may not be obvious that $\varphi(\mathcal{X}_t)$ even could be less than $\varphi(\mathcal{X}_{t-1})$. However, if v has two neighbors u and w such that $\mathcal{X}_{t-1}(u) = \mathcal{X}_{t-1}(w) \neq \mathcal{X}_{t-1}(v)$ and $u \not\sim_{\mathcal{X}_{t-1}} w$, then v could conjoin the color components of u and w and hence reduce φ. Nevertheless, we can show that the expected drift is sufficiently positive.

Clearly, the color components which do not contain vertices in $\Gamma(v) \cup \{v\}$ cannot immediately be affected by v's recoloring. So, let $m_t(c) \coloneqq |\{[u]_{\mathcal{X}_t} \mid u \in \Gamma(v) \cup \{v\} \text{ and } \mathcal{X}_t(u) = c\}|$, the number of separate color components of color c connected to v at time t. Observe that $m_t(\mathcal{X}_t(v)) = 1$, because all adjacent components of v's color are connected through v. For each other $c \in [D]$, we have $m_{t-1}(c) \leq m_t(c)$, because there are no new paths of color c. Finally, observe that $\sum_{c \in [D]} m_{t-1}(c) \leq \Delta$, because v has the same color as at least one of its neighbors. Hence, we have

$$\mathbb{E}\left[\varphi(\mathcal{X}_t) - \varphi(\mathcal{X}_{t-1})\right] = \mathbb{E}\left[\sum_{c \in [D]} m_t(c) - m_{t-1}(c)\right] \geq \sum_{c \in [D]} \frac{1 - m_{t-1}(c)}{D} \geq 1 - \frac{\Delta}{D} = \frac{1}{D}.$$

We now apply Theorem 3.3. We have that $\mathcal{X}$ is valid iff $\varphi(\mathcal{X}) = n$, and

$$\mathbb{E}\left[|n - \varphi(\mathcal{X}_{t-1})| - |n - \varphi(\mathcal{X}_t)| \;\middle|\; \mathcal{X}_{t-1} \text{ invalid}\right] = \mathbb{E}\left[\varphi(\mathcal{X}_t) - \varphi(\mathcal{X}_{t-1}) \;\middle|\; \mathcal{X}_{t-1} \text{ invalid}\right] \geq \frac{1}{D}.$$

Because $1 \leq \varphi(\mathcal{X}) \leq n$, we have $\mathbb{E}\left[|n - \varphi(\mathcal{X}_0)|\right] \leq n - 1$. Hence, $\mathbb{E}\left[\tau(G)\right] \leq (n-1)D$. ■

This theorem finally gives us a general bound for undirected variants of non-persistent coloring and persistent coloring (because persistent coloring can be seen a specific case of non-persistent coloring).

3.2 General Lower Bound

As mentioned, we can refer back to the coupon-collector coupling in Theorem 1.1 to lower-bound the worst case for any variant of non-persistent coloring, including persistent coloring:

Theorem 3.4. *The worst case expected run time of non-persistent coloring is $\Omega(n \log \Delta)$. That is, given any $n, \Delta \in \mathbb{Z}_{\geq 0}$ such that $\Delta < n$, there is a graph with n vertices and max degree Δ such that $\mathbb{E}\left[\tau(G)\right] = \Omega(n \log \Delta)$.*

Proof: Let $n, \Delta \in \mathbb{Z}_{\geq 0}$ such that $\Delta < n$ be given. Let G be the graph consisting of $\lfloor n/D \rfloor$ copies of the clique K_D, with the remaining vertices disconnected. By comparison to the coupon-collector problem with D coupons, the total run time is $\Omega(n \log \Delta)$. ■

4 Persistent Coloring

4.1 Monty Hall

We include Section 4.1 simply as an interesting observation.

Let $n \in \mathbb{Z}^+$ such that $n > 2$. Suppose we run random start, random order persistent coloring on K_n, and are informed of the colors of the first $n-2$ vertices, just before recoloring the final two vertices, u and v. Clearly, we can deduce the only two colors which u and v could have, because we can eliminate the colors of the first $n-2$ fixed vertices. Furthermore, by symmetry, $\mathcal{X}(u)$ and $\mathcal{X}(v)$ are equally likely to be either color. So, what is the probability that $\mathcal{X}(u) = \mathcal{X}(v)$ currently?

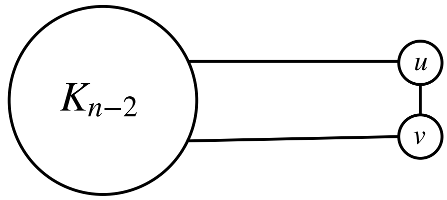

Figure 2: The graph in Section 4.1

A natural guess would be $\frac{1}{2}$. However, u and v still have their original colors, so, without conditioning on the known fixed colors, $\Pr[\mathcal{X}_0(u) = \mathcal{X}(u) = \mathcal{X}(v) = \mathcal{X}_0(v)] = \frac{1}{n}$. But, by symmetry,

$$\Pr[\mathcal{X}(u) = \mathcal{X}(v) \mid \mathcal{X}(V-u-v) = S] = \Pr[\mathcal{X}(u) = \mathcal{X}(v) \mid \mathcal{X}(V-u-v) = S'],$$

for all $S, S' \subseteq [n]$ such that $|S| = |S'| = n-2$. Therefore, *all* of these conditional probabilities must equal $\frac{1}{n}$. That is, the answer is $\frac{1}{n}$, *not* $\frac{1}{2}$. In some sense, informing us of the first $n-2$ colors is akin to Monty Hall's deliberately opening doors with goats behind them, in the Monty Hall problem. As we further discuss in Section 5, this paradox prevents us from using an approach which relies on the misconception that vertices' colors are *independently* uniformly random among the colors not used by their neighbors.

4.2 Random Start, Random Order

As explained in the warmup, bounding random start, random order persistent coloring's run time requires a method surprisingly dissimilar to the one used to bound the blank start version's. Here, we implicitly utilize the fact that some vertices may even never need to recolor.

Although a cursory estimate may suggest that considering this probability should only affect the run time by a constant factor, the chance that a vertex needs to recolor is not simply the probability that one of its neighbors initially uses its color. Rather, for a vertex to ever recolor, its initial color must be initially present in the subset of its neighborhood which recolors *after* it, because the other adjacent vertices have already been fixed to non-conflicting colors.

We now begin the proof, which uses coupling.

Theorem 4.1. *Let v be an arbitrary vertex in an arbitrary graph G. Let $d = \deg(v)$. Then $\mathbb{E}[\text{recolors}(v)] \leq H_{d+1} = O(\log d)$.*

Proof: We couple $\Gamma(v) \cup \{v\}$ and K_{d+1}. To begin, denote v by v_1 and arbitrarily order the rest of $\Gamma(v)$ as $v_2, v_3, \ldots, v_{d+1}$. Similarly, arbitrarily order K_{d+1}'s vertices by $w_1, w_2, \ldots, w_{d+1}$, and denote w_1 by w. Suppose we use the ordering σ and initial coloring $\mathcal{X}_0$ on G. For each $i \in [d+1]$, let $\pi(v_i) = \left|\left\{k \in [d+1] \mid \sigma^{-1}(v_k) \leq \sigma^{-1}(v_i)\right\}\right|$, roughly the σ ordering but restricted to our $d+1$ vertices. Then we will use σ' and $\mathcal{X}'_0$ on K_{d+1}, such that $\sigma'(\pi(v_i)) = w_i$ and $\mathcal{X}'_0(w_i) = \mathcal{X}_0(v_i)$. Assume that we use D colors on both G and K_{d+1}, although d itself may be less than Δ.

Observe that v has to recolor iff $\mathcal{X}_0(v) \in \mathcal{X}_0(A(v))$, where $A(v)$ is the subset of v's neighborhood which recolors after v. Identically, w has to recolor iff $\mathcal{X}'_0(w) \in \mathcal{X}'_0(A(w))$. But $\mathcal{X}_0(A(v)) = \mathcal{X}'_0(A(w))$ and $\mathcal{X}_0(v) = \mathcal{X}'_0(w)$, under our coupling. Hence, $\text{recolors}(v) > 0$ iff $\text{recolors}(w_1) > 0$. Similarly, we have $\text{free}(v) \geq \text{free}(w)$, because $\mathcal{X}_0(A(v)) = \mathcal{X}'_0(A(w))$, and w's fixed neighbors must have completely distinct colors, which is the worst case. Using the fact that $\text{recolors}(v)$ is either 0 or the geometric random variable with probability $\text{free}(v)/D$, we have

$$\begin{aligned}
\mathbb{E}[\text{recolors}(v)] &= \mathbb{E}\left[\mathbb{1}_{\{\text{recolors}(v)>0\}} \cdot \frac{D}{\text{free}(v)}\right] && (13)\\
&\leq \mathbb{E}\left[\mathbb{1}_{\{\text{recolors}(w)>0\}} \cdot \frac{D}{\text{free}(w)}\right] && (14)\\
&= \mathbb{E}\left[\mathbb{1}_{\{\text{recolors}(w)>0\}} \cdot \frac{D}{\text{free}(w)}\right] && (15)\\
&= \mathbb{E}[\text{recolors}(w)] && (16)
\end{aligned}$$

Hence, $\mathbb{E}[\text{recolors}(v)] \leq \mathbb{E}[\text{recolors}(w)]$. By symmetry, we can divide $\mathbb{E}[\tau(K_{d+1})]$ by $d+1$ to get $\mathbb{E}[\text{recolors}(w)]$. But $\mathbb{E}[\tau(K_{d+1})]$ is the the expected time to get the first $d+1$ out of D coupons, which is certainly less than the expected time to get all $d+1$ coupons if there are only $d+1$ total. That is,

$$\mathbb{E}[\text{recolors}(v)] \leq \mathbb{E}[\text{recolors}(w)] \leq \frac{1}{d+1}((d+1)H_{d+1}) = H_{d+1} = O(\log d),$$

which completes the proof. ■

Our main theorem now easily follows.

Theorem 4.2. *The worst case expected run time of random start, random order persistent coloring is $O(n \log \Delta)$. Specifically, for any graph $G = (V, E)$ be with n vertices and max degree Δ, we have $\mathbb{E}[\tau(G)] \leq \sum_{v \in V} H_{\deg(v)+1}$.*

Proof: Using Theorem 4.1, we have that

$$\mathbb{E}[\tau(G)] = \sum_{v \in V} \mathbb{E}[\text{recolors}(v)] \leq \sum_{v \in V} H_{\deg(v)+1} = O(n \log \Delta),$$

which completes the proof. ■

4.3 Adversarial Start, Random Order

The original version of Conjecture 1.2 was actually that *no* variant of non-persistent coloring or persistent coloring could be expected to take more than nH_D time. However we realized that we could concoct a graph G with $\Omega(n)$ vertices that have the worst possible rank, which leads to the following theorem:

Theorem 4.3. *The worst case expected run time of persistent coloring is $\Omega(n\Delta)$. That is, given any $\Delta \in \mathbb{Z}_{\geq 0}$, there is a graph G with n vertices and max degree Δ such that $\mathbb{E}[\tau(G)] = \Omega(n\Delta)$.*

Proof: Let $\Delta \in \mathbb{Z}_{\geq 0}$ be given. Let G be the complete bipartite graph of degree Δ, which we will think of as a left half and a right half. Suppose we initially color every left side vertex green, and use Δ colors on the right half, including one vertex with color green. In this configuration, every left side vertex is conflicted, and there is only one conflicted right side vertex. Moreover, the left side vertices have only one free color, and hence expect to recolor D times each if selected. On average, we recolor half of the left side vertices before fixing right side vertex (at which point the process terminates). Hence, the total expected run time is at least $D \cdot \frac{\Delta}{2} = D \cdot \frac{n}{4} = O(n\Delta)$. ■

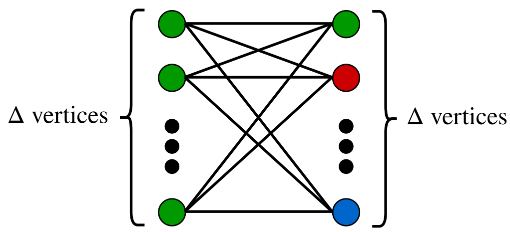

Figure 3: Graph with maximum degree Δ and $n = 2\Delta$ vertices from Theorem 4.3

5 Related Work

For general graph coloring references, see [12, 16].

This work, specifically the non-persistent coloring algorithm, derives from the IQ-Hopping algorithm in the paper by Bhartia et al.. Their paper provides a real-world application for these algorithms, proof that they can be implemented in a decentralized fashion, and empirical evidence of their speed [4]. Independently, there appears to be at least one other paper concerned with applying decentralized graph coloring to wireless networks [11].

In terms of prior work on decentralized run time analysis, most notably, Checco et al., analyze a similar algorithm in great detail. Their algorithms are slightly more complex and less restrictive, but provably obey an $O(n \log n)$ bound. Specifically, to overcome some of the issues outlined in Section 3.1, they use a periodic fixing mechanic, in which vertices can, oxymoronically, become fixed and then unfixed [6].

Before Checco et al., there were several papers which also mathematically analyzed decentralized graph coloring algorithms, although "decentalized" may have varying definitions [7–10].

As mentioned, we once attempted to prove our random start, random order persistent coloring bound based on the misconception that each unfixed vertex's color was *independently* random out of the set of colors not used by its fixed neighbors. Although we disproved this for persistent coloring in Section 4.1, a paper by Johansson investigates a similar but round-based algorithm for which our misconception actually *is* true, and indeed achieved an $O(\log n)$ bound on the expected number of rounds [15].

Recently, there have been interesting graph coloring results in the field of streaming algorithms, where information about graphs comes in pieces which we may not be able to memorize. In particular, Assadi et al. have found randomized algorithms for finding valid $(\Delta+1)$-colorings which counterintuitively take time sublinear in the number of edges [1]. Bera et al. have extended some of their results to $(\kappa + O(\kappa))$-colorings, where κ is the graph's degeneracy [3]. We are especially interested in Assadi et al.'s Palette Sparsification Theorem, which says that if each vertex picks a

Table 1: Non-persistent Coloring

Start $\rightarrow$ Order $\downarrow$	Random	Adversarial
Random	$O(n\Delta)^*$	$O(n\Delta)$
Adversarial	$O(n\Delta)$	$\boldsymbol{\Theta(n\Delta)}$

Table 2: Persistent Coloring

Start $\rightarrow$ Order $\downarrow$	Random	Adversarial
Random	$\boldsymbol{\Theta(n \log \Delta)}$	$\boldsymbol{\Theta(n\Delta)}$
Adversarial	$O(n\Delta)$	$\Theta(n\Delta)$

† Previously known [4]

Table 3: A summary of our contributions, with the most important results bolded

random $\log n$ sized subset of the $\Delta + 1$ colors, then with high probability there is a valid $(\Delta+1)$-coloring of the graph where each vertex uses a color from its subset. A conjecture of ours related to Conjecture 1.2 is that $\mathbb{E}[\max_{v \in V} \text{recolors}(v)] = O(\log n)$. If this is true, we believe it may provide an alternate proof of their wonderful theorem.

Additionally, there has been much work on the problem of distributed graph coloring, in which vertices can communicate [2]. For example, Bhattacharya et al. have found an algorithm which repairs graph colorings in $O(\log \Delta)$ amortized time in response to changes to the graph's structure [5].

6 Conclusion

We have thoroughly investigated the decentralized graph coloring algorithms we call non-persistent coloring and persistent coloring, and an array of their variants. Refer to Table 3 for a graphical summary of the current state of this problem and our contributions. Most importantly, in order of appearance, we have

1. Proposed the persistent coloring algorithm
2. Bounded the worst case expected run time of adversarial start, adversarial order non-persistent coloring to $\Theta(n\Delta)$
3. Bounded the worst case expected run time of random start, random order persistent coloring to $\Theta(n \log \Delta)$
4. Bounded the worst case expected run time of adversarial start persistent coloring to $\Theta(n\Delta)$

We used three primary approaches to achieve bounds:

1. Comparison to the clique case via coupling
2. Analyzing drift toward convergence
3. Bounding $\mathbb{E}\left[\frac{D}{\text{free}(v)} \mid \text{rank}(v) = r\right]$ with concentration inequalities.

The conjecture that inspired this work, that the worst case expected run time of fully randomized non-persistent coloring is $\Theta(n \log \Delta)$, remains both not proven and not disproven. However, we have successfully applied the same bound to fully randomized persistent coloring, whose run time can be neatly segmented into one geometric random process per vertex. In short, we reiterate the question of whether $P = NP$, but where the P is for persistent and the NP is for non-persistent. And in keeping with tradition, we fail to answer it.

Nevertheless, we have significantly increased our knowledge of the speed of decentralized graph coloring algorithms.

7 Acknowledgements

I would like to thank my thesis advisor, Deeparnab Chakrabarty, for working through this beautiful problem with me and providing mentorship and friendship throughout my senior year. Without his guidance, this research would not have been possible.

References

[1] Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Sublinear algorithms for $(\Delta + 1)$ vertex coloring. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, pages 767–786, Philadelphia, PA, USA, 2019. Society for Industrial and Applied Mathematics.

[2] Leonid Barenboim and Michael Elkin. *Distributed Graph Coloring: Fundamentals and Recent Developments*. Morgan & Claypool Publishers, 2013.

[3] Suman K. Bera, Amit Chakrabarti, and Prantar Ghosh. Graph coloring via degeneracy in streaming and other space-conscious models. May 2019.

[4] Apurv Bhartia, Deeparnab Chakrabarty, Krishna Chintalapudi, Lili Qiu, Bozidar Radunovic, and Ramachandran Ramjee. Iq-hopping: Distributed oblivious channel selection for wireless networks. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '16, pages 81–90, New York, NY, USA, 2016. ACM.

[5] Sayan Bhattacharya, Deeparnab Chakrabarty, Monika Henzinger, and Danupon Nanongkai. Dynamic algorithms for graph coloring. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '18, pages 1–20, Philadelphia, PA, USA, 2018. Society for Industrial and Applied Mathematics.

[6] Alessandro Checco and Doug J. Leith. Fast, responsive decentralized graph coloring. *IEEE/ACM Trans. Netw.*, 25(6):3628–3640, December 2017.

[7] K. R. Duffy, N. O'Connell, and A. Sapozhnikov. Complexity analysis of a decentralised graph colouring algorithm. *Inf. Process. Lett.*, 107(2):60–63, July 2008.

[8] Ken R. Duffy, Charles Bordenave, and Douglas J. Leith. Decentralized constraint satisfaction. *IEEE/ACM Trans. Netw.*, 21(4):1298–1308, August 2013.

[9] Stephen Fitzpatrick and Lambert Meertens. Soft, real-time, distributed graph coloring using decentralized, synchronous, stochastic, iterative-repair, anytime algorithms. *Kestrel Institute Technical Report*, May 2001.

[10] Severino F. Galán. Simple decentralized graph coloring. *Computational Optimization and Applications*, 66(1):163–185, Jan 2017.

[11] Douglas Leith and Peter Clifford. Convergence of distributed learning algorithms for optimal wireless channel allocation. pages 2980 – 2985, 01 2007.

[12] R.M. R. Lewis. *A Guide to Graph Colouring: Algorithms and Applications.* Springer Publishing Company, Incorporated, 1st edition, 2015.

[13] Frank H. Mathis. A generalized birthday problem. *SIAM Rev.*, 33(2):265–270, May 1991.

[14] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, New York, NY, USA, 2005.

[15] Öjvind Johansson. Simple distributed (Δ+1)-coloring of graphs. *Information Processing Letters 70*, 70:229–232, 1999.

[16] Bruce Reed. Graph coloring via the probabilistic method. In *Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 184–184, Philadelphia, PA, USA, 2011. Society for Industrial and Applied Mathematics.

[17] Abraham Wald. Some generalizations of the theory of cumulative sums of random variables. *Ann. Math. Statist.*, 16(3):287–293, 09 1945.