# Entropy compression method

Zdeněk Dvořák

December 12, 2019

## 1 Star coloring

A proper coloring $\varphi$ of a graph $G$ is a *star coloring* if union of any two colors induces a star forest in $G$ (i.e., forest whose components are stars). Equivalently, at least three colors are used on any 4-vertex path in $G$.

**Theorem 1.** *If $G$ is a graph of maximum degree at most $d$, then $G$ has a star coloring by at most $\lceil 100d^{3/2} \rceil$ colors.*

*Proof.* Let $q = \lceil 100d^{3/2} \rceil$ and $n = |V(G)|$. Order the vertices of $G$ arbitrarily. For a path $v_1 v_2 \ldots v_k$, the *code* of the path is the sequence of numbers $p_1$, …, $p_{k-1}$, where $p_i$ is the position of $v_{i+1}$ among the neighbors of $v_i$ (sorted according to the fixed ordering of vertices of $G$). Consider the following procedure that attempts to find a star coloring $\varphi$ of $G$ using $q$ colors.

While there exists an uncolored vertex:

- Let $v$ be the smallest uncolored vertex.
- Give $v$ a random color from $[q]$, and write out "Color".
- If there exists $uv \in E(G)$ such that $\varphi(u) = \varphi(v)$, then uncolor $u$ and $v$, and write out "Uncolor1", $\varphi(v)$, and the code of the path $vu$; else,
- if there exists a path $vv_1v_2v_3$ such that $\varphi(v) = \varphi(v_2)$ and $\varphi(v_1) = \varphi(v_3)$, then uncolor the vertices of the path and write out "Uncolor2", $\varphi(v)$, $\varphi(v_1)$, and the code of the path $vv_1v_2v_3$; else,
- if there exists a path $v_1vv_2v_3$ such that $\varphi(v) = \varphi(v_3)$ and $\varphi(v_1) = \varphi(v_2)$, then uncolor the vertices of the path and write out "Uncolor3", $\varphi(v)$, $\varphi(v_1)$, and the codes of the paths $vv_1$ and $vv_2v_3$.

In the third and the fourth step of the procedure, if there are more edges or paths to choose from, we pick one arbitrarily, but in some deterministic fashion (say the one with the smallest code among all the choices). Suppose that the procedure runs for $t$ iterations of the cycle, giving colors $c_1, \dots,$ $c_t$ to vertices $v_1, \dots, v_t$. Note the colors $c_1, \dots, c_t$ uniquely determine the run, and thus the probability that the procedure follows this particular run is exactly $q^{-t}$.

On the other hand, we claim that the run is also uniquely determined by the output of the procedure and the final coloring $\varphi_t$. Indeed, even without knowing the colors that are assigned in the second statement, we can keep track of which vertices are being colored and uncolored by following the "Color" and "Uncolor" statements, and thus determine the sequence $v_1,$ $\dots, v_t$. Next, we reconstruct the partial colorings $\varphi_1, \dots, \varphi_t$ found by the algorithm: We are given $\varphi_t$. If we know $\varphi_i$, and $v_i$ was not uncolored, then $\varphi_{i-1} = \varphi_i$. Otherwise, the "Uncolor" statement gives the colors of uncolored vertices, and thus $\varphi_{i-1}$ is obtained from $\varphi_i$ by coloring the vertices of the given path in the described way. Consequently, we can also exactly reconstruct the sequence $c_1, \dots, c_t$.

In a run of length $t$, exactly $t$ vertices are given a color, and thus the number of uncolorings of single vertices performed by the procedure is also at most $t$; since we uncolor two or four vertices at a time, there are at most $t/2$ "Uncolor" statements in the output. Hence, the output of the procedure can be encoded as a string of at most $\frac{3}{2}t$ symbols ("Color", "Uncolor1", "Uncolor2", "Uncolor3"), accompanied by a string $C$ of colors (numbers from $[q]$) and a string $P$ of elements of path codes (numbers from $[d]$) used by the "Uncolor" statements in order. Note that the procedure each time either uncolors two vertices, contributing one symbol to $C$ and one symbol to $P$; or uncolors four vertices, contributing two symbols to $C$ and three symbols to $P$. Hence, $|C| \leq t/2$ and $|P| \leq \frac{3}{4}t$. Finally, there are at most $(q+1)^n$ possible choices for the partial coloring $\varphi_t$. We conclude that there are at most $(q+1)^n 3^{\frac{3}{2}t} q^{t/2} d^{\frac{3}{4}t}$ runs of length $t$, and since each of them is taken with probability $q^{-t}$, the probability that the procedure runs for $t$ steps is at most

$$(q+1)^n 4^{\frac{3}{2}t} q^{-t/2} d^{\frac{3}{4}t} \leq (q+1)^n 4^{\frac{3}{2}t} 100^{-t/2} = (q+1)^n (25/16)^{-t/2}.$$

For $t$ sufficiently large (linear in $n \log d$), this probability is smaller than 1, and thus with non-zero probability, the procedure produces a star coloring of $G$ using $q$ colors. $\square$

## 2 Coloring triangle-free graphs

We will use the following concentration bounds.

**Theorem 2** (Simple Concentration Bound)**.** *Suppose a random variable $X$ is determined by $n$ independent trials and changing the outcome of one trial can change $X$ by at most $c$. Then for any $t \geq 0$,*

$$Pr[|X - E[X]| > t] \leq 2e^{-\frac{t^2}{2c^2 n}}.$$

**Theorem 3** (Chernoff Bound)**.** *Suppose a random variable $X$ is a sum of independent boolean variables. Then for any $a \geq 1$,*

$$Pr[X \geq (1+a)E[X]] \leq e^{-aE[X]/3}.$$

Let $G$ be a triangle-free $d$-regular graph, and let $s = \sqrt{14d \log d}$. Let $L$ be an assignment of lists of size $q = \lceil 3d/\log d \rceil$ to vertices of $G$. Consider a partial proper $L$-coloring $\varphi$ of $G$. For $v \in V(G)$, let $F_v$ be the set of colors in $L(v)$ that $\varphi$ does not use on any of the neighbors of $v$. By *recoloring the neighborhood of* $v$, we mean changing the color of each neighbor $u$ of $v$ uniformly independently by a color from $F_u \cup \{\text{blank}\}$; if "blank" is chosen, we uncolor $u$, instead. Let $A_v$ denote the event that $|F_v| \leq s$, and $B_v$ the event that at least $s$ neighbors of $v$ are uncolored.

**Lemma 4.** *There exists $d_0$ such that for all $d \geq d_0$, after the neighborhood of $v$ is recolored, the probability that $A_v$ holds is at most $d^{-4}$.*

*Proof.* For each color $c \in L(v)$, let $\rho(c) = \sum_{uv \in E(G), c \in F_u} \frac{1}{|F_u|}$. Note that $\sum_{c \in L(v)} \rho(c) \leq \sum_{uv \in E(G)} \sum_{c \in F_u} \frac{1}{|F_u|} \leq d$. If $c \in F_u$, then $|F_u| \geq 1$, and thus $1 - \frac{1}{|F_u|+1} > e^{-1/|F_u|}$. Probability that $c$ belongs to $F_v$ after recoloring is $\prod_{uv \in E(G), c \in F_u} \left(1 - \frac{1}{|F_u|+1}\right)$, and thus

$$E[|F_v|] = \sum_{c \in L(v)} \prod_{uv \in E(G), c \in F_u} \left(1 - \frac{1}{|F_u|+1}\right) > \sum_{c \in L(v)} e^{-\rho(c)}.$$

Since $e^{-x}$ is convex, we have

$$\frac{1}{q} \sum_{c \in L(v)} e^{-\rho(c)} \geq e^{-\sum_{c \in L(v)} \rho(c)/q} \geq e^{-d/q}.$$

Consequently, $E[|F_v|] > qe^{-d/q} \geq \frac{3d}{\log d} e^{-\frac{1}{3}\log d} = 3d^{2/3}/\log d \geq 2s$, since $d$ is large enough.

Note that $|F_v|$ is a random variable determined by $d$ independent trials (choices of colors at the neighbors of $v$). Furthermore, changing one of these trials may add or remove occurrence of at most one color in the neighborhood of $v$, and thus $|F_v|$ is changed by at most 1. Consequently, the Simple Concentration Bound implies that

$$\begin{aligned}\Pr[|F_v| \leq s] &\leq \Pr[||F_v| - E[|F_v|]| > s] \\ &\leq 2e^{-\frac{s^2}{2d}} = 2e^{-7\log d} < d^{-4},\end{aligned}$$

as required. □

**Lemma 5.** *There exists $d_0$ such that the following holds for all $d \geq d_0$. Suppose that $A_u$ is false for all neighbors $u$ of a vertex $v$. After the neighborhood of $v$ is recolored, the probability that $B_v$ holds is at most $d^{-4}$.*

*Proof.* Let $X$ be the number of neighbors of $v$ that are uncolored after recoloring. Consider a neighbor $u$ of $v$. Since $A_u$ is false, we have $|F_u| \geq s$, and thus after recoloring $u$ is uncolored with probability less than $1/s$. Hence, $E[X] \leq d/s$. On the other hand, a neighbor $u$ is uncolored with probability at least $1/(q+1)$, and thus $E[X] \geq d/(q+1) \geq 1$ for large enough $d$. Note that $X$ is a sum of independent boolean variables, and thus by Chernoff Bound, we have

$$\begin{aligned}\Pr[X \geq s] &\leq \Pr[X \geq \frac{s^2}{d} E[X]] \\ &= \Pr[X \geq 14 \log d E[X]] \\ &\leq e^{-\frac{(14\log d - 1)}{3} E[X]} \leq e^{-4\log d} = d^{-4}\end{aligned}$$

for large enough $d$. □

Order the vertices of $G$ arbitrarily. We also fix an ordering on the events $A_v$ and $B_v$: $A_u \prec B_v$ for all $u, v$, and $A_u \prec A_v$ and $B_u \prec B_v$ whenever $u < v$. Consider the following recursive procedure $\texttt{Fix}(X_v)$, called on an event $X_v$ which holds at a vertex $v$, such that either $X_v = A_v$, or $X_v = B_v$ and none of the events $A_u$ for neighbors $u$ of $v$ holds.

- Write out the colors of neighbors of $v$.
- Recolor the neighborhood of $v$.
- While $A_u$ holds for some vertex $u$ at distance at most three from $v$, or $B_u$ holds for some vertex $u$ at distance at most two from $v$, then let $X_u$ be the minimal such event according to the ordering fixed above, and

  - Write out whether $X_u$ is $A_u$ or $B_u$, and the code of a shortest path from $v$ to $u$.
  - Call `Fix`$(X_u)$.

- Write out "Return".

Note that if `Fix`$(X_v)$ finishes, then $X_v$ does not hold, and if $A_u$ or $B_u$ holds afterwards for some $u \in V(G)$, then it used to hold before the call as well (new events may appear due to the recoloring of the neighborhood of $v$, but these events are contained in the second neighborhood of $v$, and thus they are fixed recursively before the procedure ends).

Let us also make the "Write out the colors of neighbors of $v$" statement more precise. There are $C_v = \prod_{uv \in E(G)}(|F_u| + 1)$ possible valid colorings of the neighbors of $v$, but by Lemmas 4 and 5, at most $d^{-4}C_v$ of them have the property that $X_v$ holds. We order such colorings arbitrarily, and write out just the position of the current coloring of the neighborhood of $v$ in this ordering.

**Lemma 6.** *Let $n = |V(G)|$. With high probability, the procedure* `Fix`$(X_v)$ *finishes in $O(n)$ steps.*

*Proof.* Consider the state after `Fix` has been called $t$ times during the execution. Suppose the current coloring $\varphi_t$ and the initial event $X_v$ are given, together with the list of things written out by the procedure. We claim that from this information, we can exactly reconstruct the run of the procedure, including the exact colors assigned to each vertex during the recoloring steps.

Indeed, since we write out the description of the path that identifies the vertex $u$, as well as the type of the event $X_u$ on that we recurse, and since we write out the "Return" statements, we can at any moment keep track of which event $X_u$ is being processed in the current call. Consider the last call `Fix`$(X_u)$, which produces $\varphi_t$ from a coloring $\varphi_{t-1}$ by recoloring the neighborhood of $u$. Note that $\varphi_t$ matches $\varphi_{t-1}$ on the neighborhoods of neighbors of $u$, and thus we know the sets $F_w$ for neighbors $w$ of $u$ at the time of recoloring. Thus, we can decode the colors of these neighbors $w$ before recoloring from the written out record, and to reconstruct $\varphi_{t-1}$. Going back in time, we analogously reconstruct all the colorings up to the original one.

Now, consider any run of the procedure with $t$ steps, calling it on $X_{v_1}$, $\ldots$, $X_{v_t}$ in order. Let $C_i = \prod_{uv_i \in E(G)}(|F_u|+1)$, for the sets $F_u$ at the moment of the call `Fix`$(X_{v_i})$. The run is uniquely determined by the inital event $X_v$, the initial coloring, and the choice of one of $C_i$ colorings of the neihborhood of $v_i$ at each step $i = 1, \ldots, t$. Since the recolorings are performed uniformly independently, the probability of this particular run is $\frac{1}{C_1 C_2 \cdots C_t}$. Let us call

$\lambda = \lfloor \log_2(C_1 \cdots C_t) \rfloor$ the *order* of the run; note that the probability of the run is at most $2^{-\lambda}$. Furthermore, $|F_u| + 1 \leq q + 1 \leq d$, and thus $C_i \leq d^d$ and $\lambda \leq td \log_2 d$.

On the other hand, during the call to $\texttt{Fix}(X_{v_i})$, the program writes out one of at most $d^{-4}C_i$ recolorings, for each recursive call one of 2 types and at most $d^3$ paths that identify it, and the return statement. This gives altogether at most $d^{-4t}2^{\lambda+1}(3d^3)^t$ possible outputs for runs of $\texttt{Fix}(X_v)$ of order $\lambda$, which combined with one of at most $(q+1)^n \leq d^n$ final partial colorings $\varphi_t$ also uniquely determines the run. Hence, there are at most $3^t d^{n-t} 2^{\lambda+1}$ runs of order $\lambda$.

Consequently, the probability that $\texttt{Fix}(X_v)$ runs for $t$ steps is at most

$$\sum_{\lambda=1}^{dt \log_2 d} \frac{3^t d^{n-t} 2^{\lambda+1}}{2^\lambda} \leq 2dt \log_2 d \cdot 3^t d^{n-t}.$$

For $t = 3n$ and sufficiently large $d$, this is at most $d^{-n} \ll 1$. □

**Theorem 7.** *There exists $d_0$ such that for each $d \geq d_0$, every triangle-free graph of maximum degree at most $d$ has choosability at most $\lceil 3d/\log d \rceil$.*

*Proof.* Without loss of generality, we can assume that the graph $G$ is $d$-regular (otherwise, consider some $d$-regular triangle-free supergraph). Start with $G$ completely uncolored (so $B_v$ holds at every vertex $v$). As long as there exists an event $X_v$ (either $A_v$ or $B_v$) that holds, call $\texttt{Fix}(X_v)$ for the minimum such event; by Lemma 6, this with non-zero probability succeeds in eliminating this event. Consequently, there exists a partial coloring from the lists such that neither $A_v$ nor $B_v$ holds at any vertex. This coloring can be extended to a full list coloring of $G$ greedily. □