ISRAEL JOURNAL OF MATHEMATICS 105 (1998), 93–104

ON-LINE COLORING k-COLORABLE GRAPHS

BY

H.A. KIERSTEAD

Department of Mathematics, Arizona State University
Tempe, AZ 85287, USA
e-mail: kierstead@math.la.asu.edu

ABSTRACT

We show that for any k, there exists an on-line algorithm that will color any k-colorable graph on n vertices with $O(n^{1-1/k!})$ colors. This improves the previous best upper bound of $O(n \log^{(2k-3)} n / \log^{(2k-4)} n)$ due to Lovász, Saks, and Trotter. In the special cases $k = 3$ and $k = 4$ we obtain on-line algorithms that use $O(n^{2/3} \log^{1/3} n)$ and $O(n^{5/6} \log^{1/6} n)$ colors, respectively.

1. Introduction

An **on-line graph** is a structure $G^{\ll} = (V, E, \ll)$, where $G = (V, E)$ is a graph and $\ll$ is a linear order on V. (Here V will always be finite.) The ordering $\ll$ is called an **input sequence**. Let $G_n^{\ll}$ denote the on-line graph induced by the $\ll$-first n elements $V_n = \{v_1 \ll \cdots \ll v_n\}$ of V. An algorithm A that properly colors the vertices of the on-line graph $G^{\ll}$ is said to be an **on-line coloring algorithm** if the color of the n-th vertex v_n is determined solely by the isomorphism type of $G_n^{\ll}$. Intuitively, the algorithm A colors the vertices of G one vertex at a time in the externally determined order $v_1 \ll \cdots \ll v_n$, and at the time a color is irrevocably assigned to v_n, the algorithm can only see G_n. For example, the on-line coloring algorithm First-Fit colors the vertices of $G^{\ll}$ with an initial sequence of the colors $\{1, 2, \ldots\}$ by assigning the vertex v the least color that has not already been assigned to any vertex adjacent to v. The number of colors that an algorithm A uses to color $G^{\ll}$ is denoted by $\chi_A(G^{\ll})$. For a graph G the maximum of $\chi_A(G^{\ll})$ over all input sequences $\ll$ is denoted

Received May 5, 1996

by $\chi_A(G)$. If Γ is a class of graphs, the maximum of $\chi_A(G)$ over all G in Γ is denoted by $\chi_A(\Gamma)$. The **on-line chromatic number** of Γ, denoted by $\chi_{\mathrm{ol}}(\Gamma)$, is the minimum of $\chi_A(\Gamma)$ over all on-line coloring algorithms.

Let $\Gamma(k,n)$ be the class of k-colorable graphs on n vertices. The following theorems summarize what is known about the on-line chromatic number of the classes $\Gamma(k,n)$, where $\log^{(k)} n$ denotes the log function iterated k times.

THEOREM 1.1 (Lovász, Saks, and Trotter [LST]): $\chi_{\mathrm{ol}}(\Gamma(2,n)) = \Theta(\log n)$; $\chi_{\mathrm{ol}}(\Gamma(k,n)) = O(n\log^{(2k-3)} n/\log^{(2k-4)} n)$.

THEOREM 1.2 (Vishwanathan [V]): $\chi_{\mathrm{ol}}(\Gamma(k,n)) = \Omega(\log^{k-1} n)$.

THEOREM 1.3 (Szegedy [S]): $\chi_{\mathrm{ol}}(\Gamma(k,k2^k)) \geq 2^k - 1$.

So we have good bounds for $\chi_{\mathrm{ol}}(\Gamma(2,n))$. However for $k \geq 3$, the bounds on $\chi(\Gamma(k,n))$ are very weak. For the case of perfect graphs, much more is known. Let $\Pi(k,n)$ be the class of k-colorable perfect graphs on n vertices.

THEOREM 1.4 (Kierstead and Kolossa [KK]): $\Omega(\log^{k-1} n) = \chi_{\mathrm{ol}}(\Pi(k,n)) = O(n^{10k/\log\log n})$.

The purpose of this article is to improve the upper bound for k-colorable graphs. Our main result is the following theorem, which we prove in the next section.

THEOREM 1.5: *For every positive integer k, there exists an on-line algorithm A_k and an integer N such that, for every on-line k-colorable graph $G^{\ll}$ on $n \geq N$ vertices, $\chi_{A_k}(G^{\ll}) \leq n^{1-1/k!}$.*

For the special cases $k = 3$ and $k = 4$ we obtain the following stronger results.

THEOREM 1.6: *There exists an on-line algorithm A_3 such that for every on-line 3-colorable graph $G^{\ll}$ on n vertices, $\chi_{A_3}(G^{\ll}) < 20n^{2/3}\log^{1/3} n$.*

THEOREM 1.7: *There exists an on-line algorithm A_4 such that for every on-line 4-colorable graph $G^{\ll}$ on n vertices, $\chi_{A_4}(G^{\ll}) < 120n^{5/6}\log^{1/6} n$.*

In order to prove Theorem 1.7, we will need the following theorem that has independent interest.

THEOREM 1.8: *There exists an on-line algorithm B such that for any on-line graph $G^{\ll}$ on n vertices that contains neither C_3 nor C_5, $\chi_B(G^{\ll}) < 3n^{1/2}$.*

The paper is organized as follows. In the remainder of this section, we review our notation and present two useful lemmas. In Section 2 we prove our main result, Theorem 1.5. In Section 3 we prove Theorem 1.8, and in Section 4 we prove Theorems 1.6 and 1.7. In Section 5 we make some concluding remarks.

Let $[n]$ denote the set $\{1, 2, \dots, n\}$. Then $[0] = \emptyset$. Let id denote the identity function. A cycle on t vertices is denoted by C_t. Let $G^{\ll} = (V, E, \ll)$ be an on-line graph and $W \subset V$ be a subset of the vertices of G. Then

$$N(W) = \{v \in V : v \text{ is adjacent to some } w \in W\} - W.$$

We may write $N(v)$ for $N(\{v\})$. We denote $N(v) \cap \{x \in V : x \ll v\}$ by $N^{\ll}(v)$ and $\mathbb{N}^{\ll}(v)|$ by $d^{\ll}(v)$.

Let $m(\theta, t, n)$ be the largest integer m such that there exists a family $\{E_i : i \in [m]\}$ of t-subsets of $[n]$ such that for all distinct $i, j \in [m]$, $|E_i \cap E_j| < \theta$. The following Lemma is fundamental to our proofs, as well as those of Lovász, Saks, and Trotter [LST]. The reason our bounds are stronger than theirs is that, while they had to iterate the lemma $\log\log(n)$ times, we only need to iterate it k times.

LEMMA 1.9 (Lovász [L], Problem 13.13): *If $\theta - 1 \le t \le n$ and $(\theta - 1)n \le t^2$, then $m(\theta, t, n) \le n(t - \theta + 1)/(t^2 - n(\theta - 1))$. In particular, if $0 < \delta < 1$, $m(\delta^2 n/2, \delta n, n) < 2/\delta$.*

Most of the on-line coloring algorithms we discuss are first designed to be efficient for on-line graphs with a specified number of vertices. However, we really want an algorithm that is efficient on any number of vertices. The following lemma shows that we can obtain such an algorithm from a sequence of algorithms that are efficient on specified numbers of vertices for the price of a factor of 4 in the performance.

LEMMA 1.10: *Let Γ be a class of graphs and g be an integer valued function on the positive integers such that $g(x) \le g(x+1) \le g(x) + 1$, for all x. If for every n, there exists an on-line coloring algorithm A_n such that for every graph $G \in \Gamma$ on n vertices, $\chi_{A_n}(G) \le g(n)$, then there exists a fixed on-line coloring algorithm A such that for every $G \in \Gamma$ on n vertices, $\chi_A(G) \le 4g(n)$.*

Proof: Choose a sequence of integers $c_0 = 1, c_1, c_2, \dots$ such that $2g(c_i) = g(c_{i+1})$. Let A be the algorithm that runs as follows. First apply A_{c_0} to the first c_0 vertices, then using a new palette of colors apply A_{c_1} to the next c_1 vertices, then using a new palette of colors apply A_{c_2} to the next c_2 vertices, etc. We claim that for every integer i and every graph $G \in \Gamma$ on $\sum_{0 \le h \le i} c_h$ vertices, $\chi_A(G) \le 2g(c_i)$. To see this, argue by induction on i. The base step $i = 0$ is trivial, so consider the induction step $i = j + 1$. Then A uses at most $g(c_i) = 2g(c_j)$ colors on the first $\sum_{0 \le h \le j} c_h$ vertices by the induction hypothesis, and A uses at most $g(c_i) = 2g(c_j)$ colors on the last c_i vertices. So A uses at most $2g(c_i)$ colors in all.

Now suppose that $G \in \Gamma$ is a graph on n vertices with $\sum_{0 \le h \le j} c_h \le n < \sum_{0 \le h \le i} c_h$. After coloring $\sum_{0 \le h \le j} c_h$ vertices the algorithm guesses that there are going to be $\sum_{0 \le h \le i} c_h$ vertices, which it will be able to color with the allotted number of colors, because by the claim it has accumulated a surplus of $2g(c_j) = g(c_i)$ colors. Thus A uses at most $4g(c_j) \le 4g(n)$ colors. ∎

2. The general bound

We begin by proving the following technical result.

THEOREM 2.1: *Fix k. Suppose that there exist positive numbers ϵ and C such that for all $i \in [k]$ and all positive integers n, there exists an on-line coloring algorithm $A_{i,n}$ such that for all graphs $G \in \Gamma(i,n)$, $\chi_{A_{i,n}}(G) < Cn^{1-\epsilon}$. Then, for all positive integers n, there exists an on-line coloring algorithm A such that, for all graphs $G \in \Gamma(k+1,n)$, $\chi_A(G^{\ll}) \le (2C+1)n^{1-\epsilon(1-k^2/\log n)/(k+1+k\epsilon)}$.*

Proof: We shall describe A in terms of two parameters α and δ, where $0 < \alpha, \delta < 1$. Later we will optimize the choice of these parameters. For $i \le k$, let $A_i = A_{i,n'}$, where $n' = n^{\alpha}$. Set $\delta_i = 2^{-i}\delta, s_0 = n$, and $s_i = \delta_{i-1}s_{i-1}$, for $i \le k$. Then, by Lemma 1.9, $m(s_{i+2}, s_{i+1}, s_i) < 2/\delta_i$. Let $G^{\ll} = (V, E, \ll)$ be an on-line graph with $G \in \Gamma(k+1, n)$, and let Z be a subset of V.

First we describe a dynamic data structure in terms of the life cycle of the strange mythical species of **witnesses.** **Male** witnesses are **witness vertices** in $V - Z$. **Female** witnesses are certain **witness sets** contained in Z. A **witness tree** records the female genealogy of witnesses starting from the original witness set Z (Eve). From time to time witness sets will **spawn** large **litters** of **daughters**. Each of the daughters in a litter is a subset of its **mother**. Each daughter

D in the litter has a distinct (!) **father** $F(D)$, who is a witness vertex that is adjacent in G to every vertex in D. Once a witness set is spawned, it will never gain or loose elements. However, Eve is special in that Eve was not spawned and will gain, but not loose, elements. The witness sets form a tree with Eve at the 0-th level, the daughters of Eve at the 1-st level, their daughters at the 2-nd level, and so on, through the k-th generation. A witness set at the i-th level is called an i-witness set. For all $i > 0$, an i-witness set has size s_i. At some times some of the witness sets may **die**. Once they die, they will never **live** again. If they never die, they are **immortal**. Otherwise they are **mortal**. If all the daughters in a single litter die, then the mother also dies (of grief).

Next we describe the on-line coloring algorithm A, using the above data structure. For any i-witness set W, with $i < k$, let

$$N^*(W) = \{v \in V - Z : |N(v) \cap W| \geq s_{i+1}\}.$$

If W is a k-witness, then $N^*(W) = N(W)$. The algorithm will maintain a partition $\{S_W : W \text{ is a witness set}\}$ of $V - Z$ such that each $S_W \subset N^*(W)$. Each S_W will be partitioned by $P_W = \{X_W(j) : j \in [t_W]\}$ of S_W. The last part $X_W(t_W)$ of this partition is called the **active** part. When new elements enter S_W they will be put in the active part. Let $X = \bigcup\{P_W\colon W \text{ is a witness set}\}$. Call $X_W(j) \in X$ **small** if it has size less than n^α. Otherwise it is **large**. The algorithm will partition V into Z, at most n^{1-a} large subsets of size n^α, and a bounded number of small subsets. The algorithm will color each of these subsets with disjoint palettes of colors. The palette for Z will have δn colors and each of the other palettes will have at most $Cn^{\alpha(1-\epsilon)}$ colors; colors are allocated to each palette as they are needed.

Consider the input sequence $v_1 \ll v_2 \ll \cdots \ll v_n$ of $G^{\ll}$. At the s-th stage the algorithm processes the vertex v_s as follows.

1. If $d^{\ll}(v_s) < \delta n$, then put v_s in Z. Color v_s by First-Fit applied to Z, using a palette of size at most δn, whose colors are dynamically allocated to Z as needed.

2.1. Otherwise v_s is a witness vertex. Find a *live* i-witness set W, with i as large as possible subject to the condition that $v_s \in N^*(W)$. Such a witness set exists by the fact that $|N(v_s) \cap Z| \geq \delta n$ and so $v_s \in N^*(Z)$, provided we can prove (Lemma 2.2) that Eve is immortal.

2.2. Put v_s in the active part $X_W(t)$, $t = t_W$, of P_W. Color v_s by A_i

applied to $X_W(t)$, using a palette of size at most $Cn^{\alpha(1-\epsilon)}$, whose colors are dynamically allocated to $X_W(t)$ as needed. (By Step 2.3, $|X_W(t)| \leq n^\alpha$.)

2.3. If after the addition of v_s, $|X_W(t)| = n^\alpha$, then set $t_W = t+1$ and set $X_W(t_W) = \emptyset$. Then $X_W(t)$ is large.

2.4. If $n^{\alpha(1-\epsilon)}$ colors have been used on $X_W(t)$, then we have a proof that $\chi(X_W(t)) = i+1$. Set $t_W = t+1$ and set $X_W(t+1) = \emptyset$. (We may have just done this.) In this case, if $i = k$, then W dies. (This may cause some female ancestors of W to die of grief.) Otherwise $i < k$ and W spawns a new litter consisting of $\{D_v : v \in X_W(t)\}$, where each daughter D_v is a s_{i+1}-subset of $N(v) \cap W$. The father of D_v is v, for each $v \in X_W(t)$. Set $t_W = 1$.

This completes the description of the algorithm A. To show that the algorithm is well defined, we need the following Lemma.

LEMMA 2.2: *Eve is immortal.*

Proof: Suppose that Z is mortal. Let c be a proper $(k+1)$-coloring of G. First consider any mortal $(i-1)$-witness set M, with $i \in [k]$. Since M is mortal, M has a litter L such that every daughter $D \in L$ is mortal. When L is spawned, $\chi(\{F(D) : D \in L\}) = i$.

Thus $|\{c(F(D)) : D \in L\}| \geq i$. It follows that, setting $W_0 = Z$, we can find a collection $\{W_i : i \in [k]\}$ such that W_i is a mortal daughter of W_{i-1} and $|\{c(F(W_i)) : i \in [k]\}| = k$. Every father in the set $\{F(W_i) : i \in [k]\}$ is an ancestor of W_k and so is adjacent to every vertex in W_k. Thus c must color every vertex in W_k with the same color. It follows that c restricted to $N(W_k)$ is a proper k-coloring. So W_k must be immortal, which is a contradiction. ∎

Clearly A produces a proper coloring of G. It remains to show that A uses at most $(2C+1)n^{1-\epsilon(1-k^2/\log n)/(k+1+k\epsilon)}$ colors. This will follow easily from the next Lemma.

LEMMA 2.3: *Let $Q = \{X_W(j) \in X : X_W(j)$ is small$\}$. Then $|Q| \leq 2^{k^2}(n^\alpha/\delta)^k$.*

Proof: We first show that any i-witness set M has less than $2/\delta_i$ litters and $2n^\alpha/\delta_i$ daughters. We may assume that M is alive since after M dies, M will have no more litters. Then each litter of M contains a live $(i+1)$-witness set. Suppose W and U are two daughters of M from distinct litters. Then there exist distinct j and j' such that $F(W) \in X_M(j)$ and $F(U) \in X_M(j')$. Say $j < j'$.

Then at the stage that $F(U)$ is processed, all the vertices in W have already been processed. Thus $|W \cap U| < s_{i+2}$, since otherwise $F(U)$ would be put in S_W instead of S_M. Thus by Lemma 1.9, M has less than $2/\delta_i$ litters. Since each litter has at most n^α daughters, we have proved the claim. Note also that a k-witness set spawns no litters.

For any fixed i-witness set U, $l_i = |\{X_U(j) \in P_U : X_U(j) \text{ is small}\}|$ is at most one more than the number of litters of W. Thus $l_k \leq 1$ and $l_i \leq 2/\delta_i$, for all $i < k$. Let w_i be the number of i-witness sets. Then $w_0 = 1$ and $w_{i+1} \leq 2^{i+1} w_i n^\alpha/\delta$. It follows that $w_i \leq 2^{i(i+1)/2}(n^\alpha/\delta)^i$. So $|Q| \leq \sum_{0 \leq i \leq k} l_i w_i \leq 2w_k \leq 2^{k^2}(n^\alpha/\delta)^k$. ∎

The algorithm A partitions V into at most $2^{k^2}(n^\alpha/\delta)^k$ small pieces and at most $n^{1-\alpha}$ large pieces. Each piece is colored with at most $Cn^{\alpha(1-\epsilon)}$ colors, except that Z is colored with at most δn colors. Thus, the algorithm uses at most

$$\delta n + Cn^{\alpha(1-\epsilon)}(2^k n^\alpha/\delta)^k + Cn^{1-\alpha}n^{\alpha(1-\epsilon)} = \delta n + C2^{k^2}n^{\alpha(k+1-\epsilon)}\delta^{-k} + Cn^{1-\alpha\epsilon}$$

colors. Setting $\delta = n^{-1/k+\alpha+\alpha/k+k/\log n}$ and $\alpha = (1-k^2/\log n)/(k+1+k\epsilon)$ we have

$$\begin{aligned} &\delta n + C2^{k^2}n^{\alpha(k+1-\epsilon)}\delta^{-k} + Cn^{1-\alpha\epsilon} \\ &\leq n^{1-1/k+\alpha(k+1)/k+k/\log n} + Cn^{1-\alpha\epsilon} + Cn^{1-\alpha\epsilon} \\ &\leq (2C+1)n^{1-\epsilon(1-k^2/\log n)/(k+1+k\epsilon)}. \end{aligned}$$ ∎

COROLLARY 2.4: *For every positive integer k, there exists an on-line algorithm A_k and an integer N such that, for every on-line k-colorable graph $G^{\ll}$ on $n \geq N$ vertices, $\chi_{A_k}(G^{\ll}) \leq n^{1-1/(k+1)!}$.*

Proof: Arguing by induction on k, it follows immediately from Theorem 2.1 that for sufficiently large positive integers n, there exists an algorithm $A_{k,n}$ such that for all $G \in \Gamma(k,n)$, $\chi_{A_{k,n}}(G) < \frac{1}{4}n^{1-1/(k+1)!}$. Thus we are done by Lemma 1.10. ∎

Corollary 2.4 is not quite as strong as Theorem 1.5. To obtain the full strength of Theorem 1.5, note that Eve could spawn a new litter every time that we learn that the chromatic number of $X_Z(t_Z)$ is greater than zero, i.e. every time that we learn that $X_Z(t_Z)$ is non-empty. Then the litters of Eve would have size one. This gives an improved estimate on the number of mothers and k-witness sets, which in turn allows one to calculate the tighter bound of Theorem 1.5. This is all done much more carefully in Section 4 for the special cases of $k = 3$ and $k = 4$. However, the final details of the proof of Theorem 1.5 are left to the reader.

3. Graphs that contain neither C_3 nor C_5

In this section we prove Theorem 1.8. We begin by defining an algorithm B that is intended to color any on-line graph $G^{\ll}$ on n vertices that contains neither C_3 nor C_5 with less than $3n^{1/2}$ colors. Initialize the variables by setting $t = 0$ and $W_1 = \emptyset$. Recall that $[0] = \emptyset$. Next consider the input sequence $v_1 \ll v_2 \ll \cdots \ll v_n$ of $G^{\ll}$. At the s-th stage the algorithm processes the vertex v_s as follows.

1. If there exists $i \in [t]$ such that v_s is not adjacent to any vertex colored $2i-1$, then let j be the least such i and color v_s with $2j-1$.
2. Otherwise, if there exists $i \in [t]$ such that $v_s \in N(W_i)$, then let j be the least such i and color v_s with $2j$.
3. Otherwise, if $W_t = \emptyset$, let $W_t = \{v \in N^{\ll}(v_s)\colon$ the color of v is odd$\}$ and color v_s with color $2t$. (Note that $|W_t| \geq t$, since Case 1 does not hold. Also, for all $i < t, W_i \cap W_t = \emptyset$, since Case 2 does not hold.)
4. Otherwise, color v_s with $2t+1$, set $W_{t+1} = \emptyset$, and replace t by $t+1$.

LEMMA 3.1: *The algorithm B produces a proper coloring of G.*

Proof: First consider an odd color $2i-1$. Any vertex colored $2i-1$ is colored by either Case 1 or Case 4 and only the first vertex colored $2i-1$ is colored by Case 4. Since Case 1 does not allow a vertex to be colored $2i-1$ if it is adjacent to any other vertex colored $2i-1$, the set of vertices colored $2i-1$ is independent.

Next consider an even color $2i$. Any vertex colored $2i$ is colored by either Case 2 or Case 3 and only the first vertex v_s colored $2i$ is colored by Case 3. Consider any two vertices v_r and v_q, $s < r < q$, that are colored $2i$ by Case 2. Then there exists w_q, $w_r \in W_i$ such that v_q is adjacent to w_q and v_r is adjacent to w_r. Both w_q and w_r are adjacent to v_s. Since G does not contain C_3, neither v_q nor v_r is adjacent to v_s. Since G contains neither C_3 nor C_5, v_q is not adjacent to v_r. It follows that the set of vertices colored $2i$ is independent. ∎

LEMMA 3.2: *The algorithm B uses less than $3n^{1/2}$ colors to color any on-line graph on n vertices that induces neither C_3 nor C_5.*

Proof: First note that $W_i \cap W_j = \emptyset$ and $|W_i| \geq i$, for all distinct $i, j \in [t]$. Let

$S_i = \{v \in V : v \text{ is colored } i\}$. Then

$$n \geq |S_{2t-1}| + \sum_{1 \leq i < t} (|W_i| + |S_{2i}|) \geq 1 + \sum_{1 \leq i < t} (i+1) \geq t(t-1)/2 + t > t^2/2.$$

Thus $t < (2n)^{1/2}$. So at most $2t < 3n^{1/2}$ colors are used by the algorithm B. ∎

This completes the proof of Theorem 1.8. ∎

4. A refined algorithm

In this section we introduce a general on-line algorithm $R_{k+1,n}$ for coloring $(k+1)$-colorable on-line graphs on n vertices. The algorithm will be defined in terms of several parameters and subroutines. Theorems 1.6 and 1.7 will then be proved by specifying the parameters and subroutines and analyzing the resulting algorithm. But first, we must introduce the concept of (f, g, i)-good on-line algorithms.

For functions f and g and a natural number i, call an on-line algorithm A (f, g, i)-**good** if whenever A uses $f(|V|)$ colors to color an on-line graph $G^{\ll} = (V, E, \ll)$, then there exists $W \subset V$, such that $|W| \leq g(|V|)$ and $\chi(G[W]) > i$. For example, the algorithm B from Theorem 1.8 is $(4n^{1/2}, 5, 2)$-good. Trivially First-Fit is $(1, 1, 0)$-good and $(2, 2, 1)$-good. Also, if A is an on-line algorithm that colors any i-colorable on-line graph on n vertices with less than $f(n)$ colors, then A is (f, id, i)-good. Let $\{B_i : i \leq k\}$ be a collection of on-line algorithms such that B_i is (f_i, g_i, i)-good, for all $i \leq k$.

We are now ready to state the on-line algorithm $R = R_{k+1,n}$, in terms of the (f_i, g_i, i)-good subroutines B_i, $i \leq k$, and the parameter δ, using the witness tree defined in Section 2. In fact, R is the same as the algorithm A of Section 2, except that Step 2.3 is omitted and Step 2.4 is replaced by the step 2.4* given below.

2.4* If $f_i(|X_W(t)|)$ colors have been used on $X_W(t)$, then we can find a subset $F \subset X_W(t)$ such that $|F| \leq g_i(|X_W(t)|)$ and $\chi(F) > i$. If $i = k$, then W dies. Otherwise $i < k$ and W spawns a new litter consisting of $\{D_v : v \in F\}$, where each daughter D_v is a s_{i+1}-subset of $N(v) \cap W$. The father of D_v is v, for each $v \in X_W(t)$. Set $t_W = t+1$ and set $X_W(t+1) = \emptyset$.

Just as before, Eve is immortal; so Step 2.2 of the algorithm is well defined. Clearly, R produces a proper coloring. We are left with the problem of determining the number of colors that R uses. Let l_i be one more than the maximum number of litters an i-witness set can have. Let w_i be the number of i-witness sets. Let c_i be the number of colors used to color $\bigcup\{S_W : W \text{ is an } i\text{-witness set}\}$. The following Lemma relates these parameters.

LEMMA 4.1: *Let $i \leq k$, W be an i-witness set, and $j \in [t_W]$. Then*

(1) *$l_i \leq 2/\delta_i$, except that $l_k = 1$,*
(2) *the litter of W corresponding to $X_W(j)$ has size at most $g_i(|X_W(j)|)$,*
(3) *$w_0 = 1$ and $w_i < 2w_{i-1}g_{i-1}/\delta_{i-1}$,*
(4) *$c_i \leq \sum f_i(|X_W(s)|)$, where the sum is over all i-witness sets W and $s \in [t_W]$, and*
(5) *$c_i \leq l_i w_i f_i(n/(l_i w_i))$, provided that f_i is concave down.*

Proof: (1) Let M be an i-witness set. We may assume that M is alive since after M dies, M will have no more litters. Then each litter of M contains a live $(i+1)$-witness set. Suppose W and U are two daughters of M from distinct litters. Then there exist distinct j and j' such that $F(W) \in X_M(j)$ and $F(U) \in X_M(j')$. Say $j < j'$. Then at the stage that $F(U)$ is processed, all the vertices in W have already been processed. Thus $|W \cap U| < s_{i+2}$, since otherwise $F(U)$ would be put in S_W instead of S_M. Thus by Lemma 1.9, M has less than $2/\delta_i$ litters. Note also that a k-witness has no litters. (2–4) These statements follow immediately from the statement of the algorithm. (5) This follows from Jensen's inequality and the observation that $n \geq \sum |X_W(s)|$, where the sum is over all i-witness sets W and $s \in [t_W]$. ∎

Proof of Theorem 1.6: Let $k = 2$ and consider a fixed n. We shall first specify an on-line coloring algorithm $A_{3,n}$ that will color any 3-colorable on-line graph on n vertices with at most $5n^{2/3}\log^{1/3} n$ colors. Let $B_0 = B_1 =$ First-Fit and $B_2 = R_2$, where R_2 is an on-line coloring algorithm that will color any 2-colorable on-line graph on m vertices with less than $2\log m$ colors. By [LST] such an algorithm exists. Then each B_i is (f_i, g_i, i)-good, where the values of the f_i and g_i as well as bounds on l_i, w_i and c_i, calculated from Lemma 4.1, are given in Table 1. Let $\delta = (32\log n/n)^{1/3}$, and set $A_{3,n} = R_{3,n}$, with the above choice of parameters and subroutines. Then $A_{3,n}$ uses at most $\delta n + 2/\delta + 8/\delta^2 + 32\log n/\delta^2 \leq 5n^{2/3}\log^{1/3} n$ colors.

By Lemma 1.10, there exists an on-line algorithm A_3 such that for any n and any 3-colorable on-line graph G on n vertices, $\chi_A(G) \leq 20n^{2/3}\log^{1/3} n$. ∎

Table 1

i	$f_i(x)$	$g_i(x)$	l_i	w_i	c_i
0	1	1	$2/\delta$	1	$2/\delta$
1	2	2	$4/\delta$	$2/\delta$	$16/\delta^2$
2	$2\log x$	x	1	$16/\delta^2$	$32\log n/\delta^2$

Proof of Theorem 1.7: Let $k = 3$ and consider a fixed n. We shall first specify an on-line coloring algorithm $A_{4,n}$ that will color any 4-colorable on-line graph on n vertices with at most $30n^{5/6}\log^{1/6} n$ colors. Let $B_0 = B_1 =$ First-Fit, $B_2 = B$, the algorithm from Section 3, and $B_3 = A_3$, the algorithm from the previous proof. Then each B_i is (f_i, g_i, i)-good, where the values of the f_i and g_i as well as bounds on l_i, w_i and c_i, calculated from Lemma 4.3, are given in Table 2. Let $\delta = 173^{1/2}(\log n/n)^{1/6}$, and set $A_{4,n} = R_{4,n}$, with the above choice of parameters and subroutines. Then $A_{4,n}$ uses at most $\delta n + 2/\delta + 8/\delta^2 + 34\delta^{-3/2}n^{1/2} + 173n^{2/3}\log^{1/3} n/\delta \leq 30n^{5/6}\log^{1/6} n$ colors.

By Lemma 1.10, there exists an on-line algorithm A_4 such that for any n and any 3-colorable on-line graph G on n vertices, $\chi_A(G) \leq 120n^{5/6}\log^{1/6} n$. ∎

Table 2

i	$f_i(x)$	$g_i(x)$	l_i	w_i	c_i
0	1	1	$2/\delta$	1	$2/\delta$
1	2	2	$4/\delta$	$2/\delta$	$16/\delta^2$
2	$3x^{1/2}$	5	$8/\delta$	$16/\delta^2$	$34\delta^{-3/2}n^{1/2}$
3	$20x^{2/3}\log^{1/3} x$	x	1	$640/\delta^3$	$173n^{2/3}\log^{1/3} n/\delta$

5. Remarks

All the on-line algorithms presented here clearly run in polynomial time. From this point of view the upper bounds we prove are reasonably strong, since the best off-line polynomial time algorithm for coloring a k-colorable graph uses close to $n^{1-1/k}$ colors. The algorithms A_3 and A_4 are even better, since the best polynomial time algorithms for 3-colorable and 4-colorable graphs use more than $n^{3/14}$ [BK] and $n^{3/5}$ [B] colors, respectively. The on-line polynomial algorithm B actually beats these bounds in the case that G contains neither C_3 nor C_5.

There is still a huge difference between the lower bounds of Theorem 1.2 and our new upper bounds. While the upper bounds probably can be improved, I believe that the lower bounds are very weak. Note that the lower bounds for general graphs are no better than those for perfect graphs. I suggest the following challenge for the reader.

Problem 5.1: Prove that $\Omega(n^\epsilon) = \chi_{ol}\Gamma(3,n) = O(n^{1/2})$.

References

[B] A. Blum, *New approximation algorithms for graph coloring,* Journal of the Association for Computing Machinery **31** (1994), 470–516.

[BK] A. Blum and D. Karger, *An $\tilde{O}(n^{3/14})$-coloring for 3-colorable graphs,* preprint.

[KK] H. Kierstead and K. Kolossa, *On-line coloring of perfect graphs,* Combinatorica, to appear.

[L] L. Lovász, *Combinatorial Problems and Exercises,* Akadémiai Kiadó, Budapest, 1993.

[LST] L. Lovász, M. Saks and W.T. Trotter, *An online graph coloring algorithm with sublinear performance ratio,* Discrete Mathematics **75** (1989), 319–325.

[S] M. Szegedy, private communication (1986).

[V] S. Vishwanathan, *Randomized online graph coloring,* Journal of Algorithms **13** (1992), 657–669.