Journal of Algorithms **41,** 404–416 (2001)
doi:10.1006/jagm.2001.1192, available online at http://www.idealibrary.com on IDEAL®

Approximation Algorithms for the Achromatic Number

Amitabh Chaudhary

Department of Computer Science, Johns Hopkins University, Baltimore, Maryland 21218
E-mail: amic@cs.jhu.edu

and

Sundar Vishwanathan

Department of Computer Science and Engineering, Indian Institute of Technology, Mumbai 400076, India
E-mail: sundar@cse.iitb.ernet.in

Received April 27, 2000

The achromatic number for a graph $G = \langle V, E\rangle$ is the largest integer m such that there is a partition of V into disjoint independent sets $\{V_1, \ldots, V_m\}$ such that for each pair of distinct sets V_i, V_j, $V_i \cup V_j$ is not an independent set in G. Yannakakis and Gavril (1980, *SIAM J. Appl. Math.* **38**, 364–372) proved that determining this value for general graphs is NP-complete. For n-vertex graphs we present the first $o(n)$ approximation algorithm for this problem. We also present an $O(n^{5/12})$ approximation algorithm for graphs with girth at least 5 and a constant approximation algorithm for trees.

Key Words: achromatic number; approximation algorithms; NP-completeness; graph algorithms.

1. INTRODUCTION

A *complete coloring* of a graph $G = \langle V, E\rangle$ is a partition $P = \{V_1, \ldots, V_m\}$ of the vertices V such that each induced subgraph $\langle V_i\rangle$, $V_i \in P$, is an independent set, and, for each pair of distinct sets $V_i, V_j \in P$, the induced subgraph $\langle V_i \cup V_j\rangle$ is not an independent set. The largest integer m for which G has a complete coloring is called the *achromatic number* of the graph and is denoted by $\Psi(G)$.

0196-6774/01 $35.00

The achromatic number was defined and studied by Harary et al. [7] and Harary and Hedetniemi [6]. Computing the achromatic number for a general graph was proved NP-complete by Yannakakis and Gavril [11]. A simple proof of this fact appears in [5]. Bodlaender [1] proved, further, that the problem remains NP-complete even when we limit ourselves to connected graphs that are both interval graphs and co-graphs. The NP-completeness of the achromatic number for trees was established only recently [9]. For graphs that are complements of trees this problem can be solved in polynomial time [11].

An *approximation algorithm* for a problem, loosely speaking, is an algorithm that runs in polynomial time and produces an "approximate solution" to the problem. We say that an algorithm is an α-approximation algorithm for a maximization problem if it always delivers a solution whose value is at least a factor $\frac{1}{\alpha}$ of the optimum. α is called the approximation ratio. See [4] and [10] for more details.

In the next section we present polynomial time algorithms for approximating the achromatic number. Our main result is an approximation algorithm for general graphs. It achieves an approximation ratio of $O(n/\sqrt{\log n})$, where n is the number of vertices. This is the first $o(n)$ approximation algorithm known for this problem. We then give algorithms for special classes of graphs. The algorithm for trees achieves an approximation ratio of 7. For graphs with girth at least 5 we present an algorithm with an approximation ratio of $O(n^{5/12})$. The last two algorithms involve ideas completely different from those used for general graphs.

1.1. *Preliminary Definitions and Results*

Let $G = \langle V, E\rangle$ be a graph. Let $u, w \in V$, and $U, W \subseteq V$. We use $N(u)$ to denote the *neighborhood* of the vertex u and $N(U)$ to denote the neighborhood of the vertex set U, i.e., $N(u) = \{w : (u, w) \in E\}$ and $N(U) = \{w : w \notin U, \exists u \in U : (u, w) \in E\}$. U and W will be termed *adjacent* if $N(U) \cap W \neq \emptyset$ or $U \cap N(W) \neq \emptyset$.

A *coloring* of a graph is a partition of the vertex set into independent sets. Each such set is called a color class. A coloring is called *complete* or *irreducible* if every pair of distinct color classes C_i, C_j is adjacent. A complete coloring is called *maximum* when the number of color classes is the maximum possible, i.e., it is equal to the achromatic number of the graph. A *partial complete coloring* is a coloring in which only some of the vertices have been colored, but each color class is adjacent to each of the other color classes.

The *distance* between two vertices of a graph is the number of edges in the shortest path between the vertices. The distance between a vertex and an edge is the minimum of the distances between the vertex and an end-

vertex of the edge. The distance between two edges is the minimum of the distances between an end-vertex of the one and an end-vertex of the other edge. A pair of edges is *adjacent* when the distance between them is 0.

The *girth* of a graph is the length of the shortest cycle in the graph.

Given a maximum complete coloring for a graph $G = \langle V, E \rangle$, we can, for every pair of color classes, choose an edge that makes the two color classes adjacent. Such a set of edges is called an *essential* set.

We record some easy results.

Fact 1.1. Any partial complete coloring can be extended to a complete coloring of the entire graph.

Fact 1.2. For a graph $G = \langle V, E \rangle$ the size of any essential set is at most the size of the edge set, i.e., $\binom{\Psi(G)}{2} \leq |E|$. Consequently, $\Psi(G) = O(\sqrt{|E|})$. In particular, if Δ is the maximum degree of a vertex in V, $\Psi(G) \leq \sqrt{|V|\,\Delta} + 1$.

Fact 1.3. Let graph $G = \langle V, E \rangle$ have a maximum complete coloring, and let $U \subseteq V$ be a subset of vertices such that at most c of the color classes intersect U. Then a partial complete coloring of size $\Psi(G) - c$ can be assigned to the induced subgraph $\langle V \backslash U \rangle$. Thus $\Psi(\langle V \backslash U \rangle) \geq \Psi(G) - c$. In particular, we always have $\Psi(\langle V \backslash U \rangle) \geq \Psi(G) - |U|$.

2. APPROXIMATION ALGORITHMS FOR THE ACHROMATIC NUMBER

2.1. *An Algorithm for General Graphs*

Motivation

One approach to finding a complete coloring is to repeatedly remove maximal independent sets. Let $G_0 = G$. At the ith step, find a maximal independent set I_i in G_{i-1}. Set $G_i = G_{i-1} \backslash I_i$. It is easy to see that the independent sets so found make up a complete coloring. The crucial question is how to select the maximal independent sets. For instance, consider the complete bipartite graph minus a perfect matching. If the algorithm picks the wrong independent set initially (i.e., it picks one of the two partite classes), it outputs only two color classes in the complete coloring. But it is possible to choose independent sets, each set being the two endpoints of one of the missing matching edges, such that we get $n/2$ color classes, where n is the number of vertices.

A good criterion, perhaps, for picking I_i is that it should be small in size. The smallest such independent set cannot be found in polynomial time since

this is the *Minimum Maximal Independent Set* problem, which is known to be NP-complete. Indeed, it cannot be approximated in polynomial time to within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$, where n is the number of vertices in the graph [8].

Our strategy is to follow a semi-greedy approach. We keep picking vertices that are adjacent to or *cover* a large number of as yet uncovered vertices and stop if this cannot be done. This guarantees that we have picked a small set of vertices. The problem is that these vertices may not cover the entire graph. A naive approach is to discard the portion of the graph not covered and continue. This may not be efficient since the achromatic number might drop drastically. We get around this by using the uncovered portion of the graph in a judicious manner. Details follow.

Rough Description

We begin with a rough description of the algorithm. The algorithm proceeds in rounds. Each iteration selects an independent set as the next color class. All the iterations (barring the initial $0th$ iteration) have the same format. At the beginning of the ith iteration i color classes have been constructed. Let the color classes be $C_0, \ldots, C_{i-1}$. Consider $G'_i = G\backslash(C_0 \cup \cdots \cup C_{i-1})$. During the course of the algorithm we discard some vertices from G'_i and are left with G_i. (We hide details in this rough description.) The set of vertices in the intersection of the neighborhoods in G_i of the above color classes is called A (for *active*). The remaining vertices in G_i form the set P (for *passive*).

The algorithm actually finds a strong achromatic coloring; i.e., it finds a coloring such that, for every i, there is a vertex in C_i with neighbors in $C_0, \ldots, C_{i-1}$. Our aim is to pick a small set of independent vertices C_i that covers a large portion of A. That C_i is small would ensure, by Fact 1.3, that the achromatic number of the rest of the graph does not drop much when C_i is removed. Also, covering most of A would ensure that we can continue finding a strong achromatic coloring.

For the next color class C_i we pick some vertices from P and at least one vertex from A. This ensures that C_i is adjacent to C_j for $j < i$. Furthermore, we ensure that the neighborhood of C_i in $A\backslash C_i$ is large.

The construction of C_i is divided into two parts. In the first part, we repeatedly choose from P a vertex u that is not adjacent to any of the vertices in the partially constructed independent set C_i and is adjacent to at least α_i as yet uncovered vertices in A; α_i is a parameter that we will fix later. We add u to C_i.

In the second part we choose vertices from A in a similar manner. We choose from A a vertex u that is non-adjacent to the partially constructed C_i

and covers at least α_i new vertices in A. We add u to C_i and repeat until there is no such vertex left. This completes the construction of C_i.

The vertices in the neighborhood of C_i in A form the active set for the next iteration. There are two options for forming the next passive set. So the algorithm splits into two branches. In one branch we form the next passive set using the vertices remaining in P and discard the remaining (uncovered) vertices in A. In the other branch we discard some vertices from those remaining in A, as well as from those in P, and combine the rest to form the passive set for the next iteration.

A branch terminates when the active set for the next iteration is empty. For each such "leaf" of the "execution tree" we get a partial complete coloring of size one more than the depth of the leaf. (Note that the depth of the leaf is the same as the iteration in which the branch terminates.) We choose a partial complete coloring with the maximum size. We shall show that, when $\Psi(G) \geq n/\sqrt{\log n}$, there is at least one leaf at depth $\Omega(\sqrt{\log n})$. This leaf yields the required large partial complete coloring. If we terminate the algorithm at this depth, the total number of leaves (and hence the time taken by the algorithm) is polynomial. This completes the rough description of the algorithm.

We now describe the algorithm formally.

Algorithm Achromatic-Partition

Iteration 0:

We initialize the color class C_0 and the set A to null sets and the set P to the entire vertex set of the graph.

While there is a vertex u in P with at least α_0 neighbors in P we remove this vertex from P and add it to C_0. We also remove its neighbors and add them to A. α_0 is a parameter (an increasing function of n) that we will determine later.

The vertices in A form the active set for the next iteration, and the vertices remaining in P form the next passive set. We set $\delta_1 = \alpha_0$.

Iteration i, $(i = 1, \ldots, \sqrt{\log n}/3)$:

Comment: $C_0, \ldots, C_{i-1}$ are the color classes formed by the previous iterations. The active set A and the passive set P have been constructed in the previous iteration. We also have a number of temporary sets which are initialized to null at the beginning of this iteration. Some of these sets are not necessary for the algorithm but are needed for the analysis. We now briefly describe these temporary sets. C_P and C_A consist of vertices from P and A, respectively, that are used to form the color class C_i. P_P consists of the neighbors of C_P in P, and P_A consists of those neighbors of C_A in P that are not already in P_P. Similarly, A_P consists of the neighbors of C_P in A and A_A of those neighbors of C_A in A that are not already in A_P. P_I

consists of those vertices from the passive set that cannot be used in this iteration and are ignored. P_R is the set of passive vertices left at the end of the iteration. Similarly, A_R is the set of active vertices left at the end of the iteration and A_D is the set of those active vertices that are discarded.

Step 1. While there is a vertex u in P that does not cover A entirely but has at least $\alpha_i = n^{\epsilon}\delta_i$ neighbors in A, we transfer u to C_P, its neighbors in P to P_P, and its neighbors in A to A_P. After all such vertices are removed, we transfer the vertices remaining in P that cover all vertices in A to P_I.

Comment: The value of δ_i is computed in the previous iteration. We shall prove later that a vertex in P does not have more than δ_i neighbors in P. ϵ is a positive valued function of n which, with foresight, can be taken to be $1/\sqrt{\log n}$. The reason we cannot add any vertices from P_I to C_i is that we want to include at least one vertex from A in C_i.

Step 2. Next, while there is a vertex u in A that has at least α_i neighbors in A, we transfer u to C_A, its neighbors in P to P_A, and its neighbors in A to A_A. Suppose no vertex satisfies this condition, we transfer any one vertex from A to C_A and its neighbors in P and A to P_A and A_A, respectively.

Step 3. The vertices in C_P and C_A form the color class C_i. The vertices remaining in P form the set P_R. If the number of vertices remaining in A is at most $n^{1-\epsilon}$ we transfer all of them to A_D, to be discarded. Otherwise we transfer only those vertices that have more than $n^{\epsilon}\alpha_i$ neighbors in $P_A \cup P_R$ to A_D; the rest of the vertices form the set A_R. The vertices in P_P and A_D are discarded.

Step 4. If both A_P and A_A are empty this branch terminates, and we output the color classes $C_0, \ldots, C_i$. Otherwise the vertices in A_P and A_A form the next active set. For the next passive set, in one branch of the algorithm, we combine the vertices in P_A, P_R, and P_I. In the other branch, we combine vertices in P_A, P_R, and A_R to form the next passive set.

We set $\delta_{i+1} = 2n^{2\epsilon}\delta_i$.
End Iteration i.

Among the sets of color classes generated by the various branches, we choose one with the largest size. Note that the restriction of keeping the running time polynomial allows us to have $\Theta(\log n)$ iterations. But the only guarantee we have (see the proof of Theorem 2.1) is that some branch would be alive for $\sqrt{\log n}/3$ iterations, and so we stop after that for the sake of analysis.

Proofs and Analysis

The Algorithm Achromatic-Partition, as we shall see, results in a large partial complete coloring as long as for some set of choices, which are made in Step 4, the active set A does not shrink too fast. We will use Fact 1.3 to lower bound the value of $\Psi(\langle A \cup P \rangle)$ and Fact 1.2 to upper bound the value of $\Psi(\langle P \rangle)$. A further application of Fact 1.3 will then allow us to lower bound the size of A. We begin the analysis with some easy observations followed by a lemma.

Observation 2.1. The Algorithm Achromatic-Partition outputs a partial complete coloring. This is because at least one vertex in C_i is picked from A and all vertices in A are adjacent to at least one vertex in each color class C_j, $j < i$.

Observation 2.2. For the ith iteration the size of C_i is at most $\lceil n/\alpha_i \rceil$ since each vertex added to C_i, except possibly one, covers at least α_i new vertices.

LEMMA 2.1. *Let d_i denote the maximum number of neighbors a vertex in the passive set has within the passive set during the ith iteration. Then*

$$d_i \leq \delta_i = 2^{i-1} n^{2(i-1)\epsilon} \alpha_0.$$

Proof. It is easy to see by induction on i that $\delta_i = 2^{i-1} n^{2(i-1)\epsilon} \alpha_0$. So we shall just prove that for all i, $d_i \leq \delta_i$. We again proceed by induction on i. Note that $d_1 \leq \delta_1 = \alpha_0$. We now need to prove that $d_{i+1} \leq 2n^{2\epsilon} \delta_i$. Consider the end of iteration i. If the next passive set is formed by vertices of P_A, P_R, and P_I then $d_{i+1} \leq d_i \leq \delta_i$. Now consider the case when P_A, P_R, and A_R form the next passive set. A vertex in $P_A \cup P_R$ has at most δ_i neighbors in $P_A \cup P_R$. It also has less than $\alpha_i = n^\epsilon \delta_i$ neighbors in A_R, for otherwise this vertex is a candidate to be put in C_P in Step 1. Therefore such a vertex has less than $(1 + n^\epsilon)\delta_i$ neighbors in the new passive set. Now consider a vertex in A_R. It has fewer than than α_i neighbors in A_R. It also has at most $n^\epsilon \alpha_i$ neighbors in $P_A \cup P_R$, for otherwise this vertex would end up in A_D while executing Step 3. Thus the number of its neighbors in the new passive set is less than

$$(1 + n^\epsilon)\alpha_i = (1 + n^\epsilon) n^\epsilon \delta_i \leq 2n^{2\epsilon} \delta_i.$$

The lemma follows. ∎

We continue the analysis by introducing two parameters. Let k_i denote the achromatic number of the induced subgraph $\langle A \cup P \rangle$ at the beginning of the ith iteration. Similarly, let l_i denote the achromatic number of the induced subgraph $\langle P \rangle$ at the beginning of the ith iteration.

We now prove a bound on the value of k_1.

LEMMA 2.2. $k_1 \geq \Psi(G) - n/\alpha_0$.

Proof. Consider the beginning of the first iteration. We have $A \cup P = V(G)\backslash C_0$. From Fact 1.3, $k_1 \geq \Psi(G) - |C_0|$. The lemma follows as C_0 is at most n/α_0 in size. ∎

LEMMA 2.3. *At the end of iteration i, for at least one set of choices made in Step* 4,

$$k_{i+1} \geq \frac{1}{2^i}\left(\Psi(G) - \frac{n}{\alpha_0}\right) - 3n^{1-\epsilon},$$

and, furthermore, the value of l_{i+1} is at most $\sqrt{n2^i n^{2i\epsilon}\alpha_0} + 1$.

Proof. We shall prove that at the end of iteration i, for at least one branch,

$$k_{i+1} \geq \tfrac{1}{2}\left(k_i - 3n^{1-\epsilon}\right).$$

The first part of the lemma will then follow by induction on i, the base case being Lemma 2.2.

Let H_i denote the induced subgraph $\langle A \cup P\rangle$ at the beginning of iteration i. Consider the state at the end of iteration i. Let H' denote the induced subgraph $\langle V(H_i)\backslash C_i \backslash P_P \backslash A_D\rangle$. From Fact 1.3,

$$\Psi(H') \geq k_i - |C_i| - |P_P| - |A_D|. \tag{1}$$

We now bound from above the sizes of C_i, P_P, and A_D.

Recall that $\alpha_i = n^\epsilon \delta_i$ and $\delta_i = 2^{i-1}n^{2(i-1)\epsilon}\alpha_0$. From Observation 2.2 it follows that

$$|C_i| \leq \left\lceil \frac{n}{n^\epsilon 2^{i-1} n^{2(i-1)\epsilon}\alpha_0} \right\rceil \leq n^{1-\epsilon}. \tag{2}$$

Each vertex in C_P can have at most d_i neighbors in P_P. By Lemma 2.1 $d_i \leq \delta_i$. Therefore

$$|P_P| \leq |C_P|\delta_i \leq (|C_i| - 1)\delta_i \leq \frac{n}{\alpha_i}\delta_i = n^{1-\epsilon}. \tag{3}$$

We next prove that the size of A_D is also at most $n^{1-\epsilon}$. The non-trivial case is when we transfer vertices from A to A_D that have more than $n^\epsilon\alpha_i$ neighbors in $P_A \cup P_R$. Note that a vertex in $P_A \cup P_R$ has less than α_i neighbors remaining in A; otherwise the vertex would have been transferred to C_P in Step 1. Therefore, it follows that the number of vertices in A that satisfy the required condition are at most

$$\frac{|P_A \cup P_R|\,\alpha_i}{n^\epsilon \alpha_i} \leq n^{1-\epsilon}. \tag{4}$$

Thus from inequalities (1), (2), (3), and (4) we infer that

$$\Psi(H') \geq k_i - 3n^{1-\epsilon}.$$

Let $H'_A = \langle V(H') \backslash P_I \rangle$ and $H'_P = \langle V(H') \backslash A_R \rangle$. Note that one branch has $H_{i+1} = H'_A$ and the other has $H_{i+1} = H'_P$. Every vertex in P_I is adjacent to every vertex in A_R. Therefore for any complete coloring of H', vertices in P_I are assigned colors different from those in A_R. So, either the vertices in P_I or those in A_R are assigned at most $\Psi(H')/2$ colors. This and Fact 1.3 lead to

$$\max\{\Psi(H'_A), \Psi(H'_P)\} \geq \frac{\Psi(H')}{2}.$$

So at least one branch has $k_{i+1} = \Psi(H_{i+1}) \geq (k_i - 3n^{1-\epsilon})/2$, and this proves the first part of the lemma.

Recall from Lemma 2.1 that the maximum degree of a vertex in the subgraph $\langle P \rangle$ during iteration $(i+1)$ is $2^i n^{2i\epsilon} \alpha_0$. The rest of the proof follows from Fact 1.2. ∎

THEOREM 2.1. *The achromatic number of a graph, $\Psi(G)$, can be approximated to within $O(n/\sqrt{\log n})$, where n is the number of vertices.*

Proof. If $\Psi(G) \leq n/\sqrt{\log n}$ we output any complete coloring. If not, we use Algorithm Achromatic-Partition. We note that the algorithm results in a partial complete coloring of size at least $(i+2)$ as long as there is a vertex in A at the beginning of the $(i+1)$th iteration. From Fact 1.3 we have that, at the beginning of the $(i+1)$th iteration,

$$|A| \geq \Psi(\langle A \cup P \rangle) - \Psi(\langle P \rangle) = k_{i+1} - l_{i+1}.$$

Thus from Lemma 2.3 the size of the partial complete coloring is at least $(i+2)$ if

$$\frac{1}{2^i}\left(\Psi(G) - \frac{n}{\alpha_0}\right) - 3n^{1-\epsilon} - \sqrt{n2^i n^{2i\epsilon}\alpha_0} - 1 \geq 1.$$

For $\Psi(G) \geq n/\sqrt{\log n}$, $\alpha_0 = \log n$, $\epsilon = 1/\sqrt{\log n}$, and $i = \sqrt{\log n}/3$ the above inequality is satisfied. (The logarithms are base 2.) ∎

2.2. *Independent Matchings and Unions of Independent Stars*

DEFINITION 2.1. A subset M of the edge set E of a graph $G = \langle V, E \rangle$ is a *matching* in G if no two edges in M have a common vertex; i.e., the distance between any pair of distinct edges in M is at least one. We call a matching M *independent* if there does not exist any edge in $E \backslash M$ that is adjacent to more than one edge in M; i.e., the distance between any pair of distinct edges in M is at least two.

Observation 2.3. Given an independent matching of size $\binom{k}{2}$ for a graph $G = \langle V, E \rangle$, a partial complete coloring of size k can be assigned to the vertices of G. Furthermore, if the maximum degree of any vertex in V is at most Δ, an independent matching of size $O(|E|/\Delta^2)$ can be found by a simple greedy strategy. This, in turn, can be used to obtain a partial complete coloring of size $\Omega(\Psi(G)/\Delta)$. (Remember that $\Psi(G) = O(\sqrt{|E|})$.)

The concept of an independent matching can be extended to that of independent stars.

DEFINITION 2.2. Let $G = \langle V, E \rangle$ be a graph. A set of edges $s = \{e_1, \ldots, e_{|s|}\} \subseteq E$ is called a *star* if each edge $e \in s$ is incident on a common vertex c. c is called the *center* of the star. A set of edges $S = \{e_1, \ldots, e_{|S|}\} \subseteq E$ is called a *union of independent stars* if there exists a partition $P = \{s_1, \ldots, s_p\}$ of S, such that each $s_i \in P$ is a star in G, and for any pair of edges $e \in s_i$, $f \in s_j$, such that $i \neq j$, the distance between e and f is at least two.

We now make the following observation regarding the use of a union of independent stars in assigning a partial complete coloring to the graph.

Observation 2.4. Given a union of independent stars S such that the size (or *degree*) of each star is at most Δ_S, a partial complete coloring of size $\sup\{k: \binom{k}{2} + (k-1)(\Delta_S - 1) \leq |S|\}$ can be assigned to the graph.

2.3. *An Algorithm for Trees*

The following algorithm generates a partial complete coloring for a tree.

Algorithm Tree-Partition

Input: A tree $T = \langle V, E \rangle$, and an integer t.

Comment: t should ideally be $\Psi(T)$. We do not know the value of $\Psi(T)$ but run the algorithm for each of the possible n values.

Output: A partial complete coloring for T.

1. Mark any vertex $r \in V$ as the root of the tree T.

2. Group the edges in E into levels $0, 1, \ldots, l$ depending on the distance from r.

3. E_0 is the set of edges at levels $0, 3, \ldots, 3\lfloor \frac{l}{3} \rfloor$. Similarly E_1 and E_2 are the sets of edges at levels that are equivalent to 1 mod 3 and 2 mod 3, respectively.

Comment: It is obvious that E_0, E_1, and E_2 are each unions of independent stars.

4. For every vertex v with more than t neighbors, remove all but t of its neighbors. Let the resultant collection of independent stars be $\overline{E_0}$, $\overline{E_1}$, and $\overline{E_2}$.

Comment: Note that the degree of each $\overline{E_i}$ is at most t.

5. Now use Observation 2.4 to generate a partial complete coloring for each $\overline{E_i}$ in turn and output the coloring with the largest size.

THEOREM 2.2. *Algorithm Tree-Partition assigns a partial complete coloring of size at least* $(\Psi/7)$ *to a given tree, where* Ψ *is the achromatic number of the tree.*

Proof. Consider the execution of the Algorithm Tree-Partition when $t = \Psi$. Let E' be a set of essential edges in some maximum coloring of the given tree. Let $E'_i = E_i \cap E'$ for $0 \le i \le 2$. Note that $E'_0 \cup E'_1 \cup E'_2 = E'$. So for at least one among E'_0, E'_1, and E'_2, say E'_k,

$$|E'_k| \ge \frac{1}{3}|E'| = \frac{1}{3}\binom{\Psi}{2}.$$

Furthermore, for any vertex v, there are at most Ψ essential edges incident on v. Hence $|\overline{E_k}| \ge |E'_k| \ge \frac{1}{3}\binom{\Psi}{2}$. Thus, from Observation 2.4, given $\overline{E_k}$, we can assign a partial complete coloring of size

$$\sup\left\{k : \binom{k}{2} + (k-1)(\Psi - 1) \le \frac{1}{3}\binom{\Psi}{2}\right\} \ge \frac{\Psi}{7}.$$

The theorem follows. ∎

2.4. *An Algorithm for Graphs with Large Girth*

DEFINITION 2.3. Let $G = \langle V, E\rangle$ be a graph. The *extended neighborhood* of an edge $e \in E$, denoted by $N_E(e)$, is the set of edges at a distance at most one from e.

THEOREM 2.3. *If the girth of a graph is at least* 5, *and the achromatic number is* Ψ, *then we can find a complete coloring with at least* $\sqrt{\Psi/3}$ *colors.*

Proof. We construct the color classes incrementally. At the end of the ith step we have constructed the color classes $C_1, \ldots, C_i$, such that the color class C_j, $1 \le j \le i$, consists of a multiset of independent vertices $v_{j,1}, \ldots, v_{j,i}$. Vertices $v_{j,k}$ and $v_{k,j}$, for $k \ne j$, are adjacent.

During the $(i+1)$th step we find, for $1 \le j < i+1$, adjacent vertices $v_{j,i+1}$ and $v_{i+1,j}$. Note that $v_{j,i+1}$ could be an already existing vertex in C_j.

We claim that, as long as $i < \sqrt{\Psi/3}$, we can find such vertices.

We find these vertices one by one. At the beginning of the $(i+1)$th stage we have at most i distinct vertices in each color class, and we are left with at least $n - i^2$ vertices that have not been put in any color class.

Suppose that we have found such vertices for $j < k < i+1$. Consider the color class C_k. If there is already an edge between some vertex in C_k and a vertex in the class C_{i+1}, which is being constructed, we are done. If not, let R' denote the set of vertices that have not been put in any color class so far. It is easy to see that $|R'| > n - i^2 - 2k > n - 2i^2$. Now, let L denote the set of vertices that have a neighbor in both C_k and C_{i+1}. Since the girth is greater than 4 no two vertices in L can have a common neighbor in both C_k and C_{i+1}. There are at most i vertices in each of these two color classes. Consequently, $|L| \leq i^2$. Let $R = R' \backslash L$. Supposing there is a vertex in R that has an edge to one color class but not the other, we are done; we add this vertex to the color class to which it is not adjacent. We are left with the case that vertices in R do not have a neighbor in either of the two color classes. The achromatic number of R is at least $\Psi - (n - |R|) > \Psi - 3i^2 > 0$. Hence there is at least one edge in R. Place one endpoint in C_k and the other in C_{i+1}. ∎

Furthermore, every n-vertex graph with girth at least g has at most $n\lceil n^{2/(q-2)} \rceil$ edges (see [2, Theorem 3.7(a), p. 126]). Putting this together with the previous theorem and the fact that the achromatic number is at most $O(\sqrt{|E|})$, it follows that the achromatic number of graphs with girth at least 5 can be approximated to $O(n^{5/12})$.

3. OPEN PROBLEMS

There are a number of open problems. Find lower bounds for the approximability for general graphs. We believe that the upper bound for general graphs can be improved to $O(n^{\epsilon})$. We hazard a guess that it should be $\sqrt{\Psi}$. Can one prove good lower bounds, at least for computing the strong achromatic number of a graph? It seems as if one can get better approximation ratios as the graph gets sparser. It would be nice to come up with an algorithm that reflects this phenomenon.

Can geometric methods be used to find good algorithms, like the recent algorithms for the chromatic number? Let us note that this problem seems to behave differently from the chromatic number problem. For instance, if the achromatic number is a constant it can be found in linear time [3].

ACKNOWLEDGMENTS

The authors thank the referees for useful suggestions that led to this improved version.

REFERENCES

1. H. L. Bodlaender, Achromatic number is NP-complete for co-graphs and interval graphs, *Inform. Proces. Lett*. **31** (1989), 135–138.
2. B. Bollobás, "Extremal Graph Theory," Academic Press, London, 1978.
3. M. Farber, G. Hahn, P. Hell, and D. J. Miller, Concerning the achromatic number of graphs, *J. Combin. Theory Ser. B*, **40** (1986), 21–39.
4. M. R. Garey and D. S. Johnson, "Computers and Intractability," Freeman, San Francisco, 1979.
5. N. Goyal and Sundar Vishwanathan, A simple proof of NP-completeness of undirected grundy numbering and related problems, personal communication.
6. F. Harary and S. Hedetniemi, The achromatic number of a graph, *J. Combin. Theory* **8** (1970), 154–161.
7. F. Harary, S. Hedetniemi, and G. Prins, An interpolation theorem for graphical homomorphisms, *Portugal. Math.* **26** (1967), 453–462.
8. M. M. Halldórsson, Approximating the minimum maximal independence number, *Inform. Process. Lett.* **46** (1993), 169–172.
9. D. Manlove and C. McDiarmid, The complexity of harmonious coloring for trees, *Discrete Appl. Math.* **57** (1995), 133–144.
10. D. B. Shmoys, Computing near-optimal solutions to combinatorial optimization problems, *in* "Advances in Combinatorial Optimization" (W. Cook, L. Lovász, and P. Seymour, Eds.), pp. 355–397, Am. Math. Soc., Providence, 1995.
11. M. Yannakakis and F. Gavril, Edge dominating sets in graphs, *SIAM J. Appl. Math.* **38** (1980), 364–372.