An improved approximation of the achromatic number on bipartite graphs *

Guy Kortsarz[†] Sunil Shende[‡]

May 18, 2006

Abstract

The achromatic number of a graph $G = (V, E)$ with $\mid V \mid= n$ vertices is the largest number k with the following property: the vertices of G can be partitioned into k independent subsets $\{V_i\}_{1 \leq i \leq k}$ such that for every distinct pair of subsets V_i, V_j in the partition, there is at least one edge in E that connects these subsets. We describe a greedy algorithm that computes the achromatic number of a bipartite graph within a factor of $O(n^{4/5})$ of the optimal. Prior to our work, the best-known approximation factor for this problem was $n \log\log n / \log n$ [KK01].

1 Introduction

Consider a connected, undirected graph $G = (V, E)$ with $\mid V \mid= n$ vertices and $\mid E \mid= m$ edges. An *achromatic* coloring is a proper coloring of the graph such that for every pair of distinct colors, there is at least one edge in the graph whose endpoints are assigned those colors. Equivalently, if all the vertices in each color class of an achromatic coloring are *contracted* to a single vertex, the resulting induced graph is a clique. The *achromatic number* of G, denoted as $\psi(G)$, is the *largest* number k, $1 \leq k \leq n$, such that G admits an achromatic coloring with k colors.

The achromatic number problem is to determine $\psi(G)$ for any given graph G. This problem has been studied extensively; for instance, see the surveys of Edwards [Edw97] and of Hughes and MacGillivray [HM97]. We focus on the algorithmic aspects of the problem. Yannakakis and Gavril [YG80] proved that the achromatic number problem is NP-hard. Farber et al. [MFM86]

---

*A preliminary version of the paper appeared in European Symposium on Algorithms (ESA) 2003, pages 385–396.

†Dept. of Computer Science, Rutgers University, Camden. **Email:** `guyk@camden.rutgers.edu`

‡Dept. of Computer Science, Rutgers University, Camden. **Email:** `shende@camden.rutgers.edu`

showed that the problem remains NP-hard for bipartite graphs. Bodlaender [Bod89] established that the problem is NP-hard on graphs that are simultaneously co-graphs and interval graphs. Cairnie and Edwards [CE97] showed that the problem is NP-hard even on trees.

Since an exact solution to the problem appears to be intractable, there has been an interest in approximating the achromatic number. An *approximation algorithm with ratio* $\alpha \geq 1$ for the achromatic number problem is an algorithm that runs in polynomial time and finds, for an input graph G, a number $p \geq \psi(G)/\alpha$ such that G admits an achromatic coloring with p colors.

1.1 Previous work

One might use the following *greedy* approach for finding an achromatic coloring with a large number of colors. For any such coloring, every set of monochromatic vertices in the graph (called a *color class*) is clearly an independent set; to maximize the number of colors, it seems natural to look for small color classes. Hence, iteratively remove from the graph maximal independent sets of small size. However, the problem of finding a *minimum maximal independent set* can also not be approximated within a ratio of $n^{1-\epsilon}$ for any $\epsilon > 0$, unless P=NP [Hal93].

However, using a semi-greedy approach to extracting small independent sets, Chaudhary and Vishwanathan [CV01] gave the first sublinear approximation algorithm for the achromatic number problem with an approximation ratio of $O(n/\sqrt{\log n})$ for any graph with n vertices. They conjectured that the achromatic number can be approximated within a ratio of $O(\sqrt{\psi(G)})$ for any graph G. In support of their conjecture, they gave an algorithm that returns a $O(\sqrt{\psi(G)}) = O(n^{7/20})$ ratio approximation for graphs G with *girth* (*i.e.* length of the shortest simple cycle) at least 7. For graphs G with girth at least 6, Krysta and Loryś [KL99] described an algorithm with approximation ratio $O(\sqrt{\psi(G)}) = O(n^{3/8})$; this ratio was improved slightly to $O(n \log\log n/\log n)$ by Kortsarz and Krauthgamer [KK01]. This latter paper also showed that the Chaudhary-Vishwanathan conjecture holds for graphs of girth 5 and demonstrated an algorithm that approximates the achromatic number within a ratio of $O(n^{1/3})$.

To summarize the upper bounds on approximating the achromatic number for general or bipartite graphs, the best-known approximation ratio guarantees are just barely sub-linear in the number of vertices. We do know that graphs with large girth (at least 5) admit algorithms with relatively low approximation ratio for the achromatic number. This result hinges on the observation that $\psi(G) \leq m/n$ for graphs G with n vertices, m edges and girth at least 5 [KK01]. But, considering the complete bipartite graph, we encounter a graph with girth 4 and achromatic number equal to two, that satisfies $2 << m/n = \Omega(n)$.

On the negative side, the first hardness of approximation result for general graphs was given by Kortsarz and Krauthgamer [KK01]. They showed that the problem admits no $2-\epsilon$ ratio approximation algorithm, unless $P = NP$. In the preliminary conference version of the present paper [KS03], we stated (without a complete proof) the first non-constant lower bound for the problem. The result was that unless NP admits a randomized quasi-polynomial-time algorithm, it is impossible to approximate achromatic number on n-vertex bipartite graphs within an approximation ratio of $(\ln n)^{1/4-\epsilon}$. The methods used for proving the hardness result built upon the combination of one-round two-provers techniques and zero-knowledge techniques as suggested in Feige et.al. [UFS02]. In Halldórsson et.al. [MMHS05], the lower bound is improved to $\sqrt{\log n}$ with details to appear in the forthcoming journal version of the paper.

1.2 Our Contribution

For graphs with n vertices, all previous results for the achromatic number problem had been unable to obtain approximations better than a factor of $\tilde{O}(n)$ where $\tilde{O}(n)$ is the class of functions that are essentially $O(n)$ ignoring logarithmic factors, *i.e.* functions of the form $O(n\log^k n)$ for some constant k. In this paper, we give a combinatorial greedy approximation algorithm for the problem on bipartite graphs that lowers the $\tilde{O}(n)$ barrier on the approximation ratio. Specifically, the algorithm achieves a ratio of $O(n^{4/5})$ for approximating the achromatic number on every bipartite graph.

2 Preliminaries

Consider a graph $G = (V, E)$. Following standard terminology, we use $d_G(u)$ and $N_G(u)$ to denote, respectively, the *degree* and the set of *adjacent neighbors* of any vertex u in the graph. Wherever possible, we will simplify notation by omitting G from subscripts when the graph G is clear from the context. For any subset $U \subseteq V$ of the vertices, the subgraph of G induced by U is denoted as $G[U]$. If $G[U]$ has no induced edges, then U is said to be an *independent set* in G.

Given disjoint subsets of vertices U, W in the graph, we say that they are *adjacent* if there exist adjacent vertices $u \in U$ and $v \in W$. The set U *covers* W if every vertex in W is adjacent to some vertex in U.

A *proper k-coloring* of the graph is a mapping that assigns to every vertex, a corresponding color in the range $[1, k]$ such that adjacent vertices receive distinct colors. Thus, any proper k-coloring of a graph partitions its vertex set into k independent sets - one per color - called its *color classes*. An *achromatic k-coloring* is a proper coloring where every distinct pair of color classes are adjacent. The partition formed by the color classes is called an *achromatic partition*; henceforth, we will use the terms achromatic coloring and achromatic partition interchangeably.

The *achromatic number problem* is to determine for any given graph G, the *largest* number k such that G has an achromatic k-coloring. Note that, in contrast, the *chromatic number problem* is to determine for graph G, the *smallest* number k such that G has a proper k-coloring (which, by minimality of k, is also an achromatic coloring).

The chromatic and achromatic numbers of a graph G are denoted by $\chi(G)$ and $\psi(G)$ respectively. Clearly, $\psi(G) \geq \chi(G)$ and indeed, the problem of finding the achromatic number, being a maximization problem, is fundamentally different from that of finding the chromatic number, a minimization problem. For instance, when $\psi(G) = O(1)$, a complete coloring with $\psi(G)$ colors can be found in polynomial time by guessing $\binom{\psi(G)}{2}$ *critical* edges (see [MFM86] for a more efficient algorithm). In contrast, even when $\chi(G) = 3$, it is NP-hard to find a 3-coloring of G. However, the general cases for both problems and the bipartite case for the achromatic number problem are known to be NP-hard as mentioned in the introduction.

3 Colorings Obtained From Matchings

We first establish a series of facts that will be used in the subsequent development and analysis of our algorithm that approximates the achromatic number in any given bipartite graph. The following sequence of lemmas is well known [Mát81, Edw97, CV01, KL99].

Lemma 1 *Let U be a subset of vertices of graph G. Then an achromatic coloring of the subgraph, $G[U]$, can be extended greedily to an achromatic coloring of G.*

Lemma 2 *Consider v, an arbitrary vertex in G, and let $G \setminus v$ denote the graph resulting from removing v and its incident edges from G. Then, $\psi(G) - 1 \leq \psi(G \setminus v) \leq \psi(G)$.*

Note that if v is an isolated vertex in G, then its removal does not affect the achromatic number, *i.e.* $\psi(G \setminus v) = \psi(G)$. Hence, Lemma 2 above can be stated more generally as follows. Let U be any *ordered* subset of vertices of G. Suppose that we remove vertices in U from the graph (in the order determined by U) one by one. Let $U_c \subseteq U$ be the subset of vertices v such that v is *not isolated* in the subgraph of G that exists just prior to v's removal. Then, $\psi(G \setminus U$ is bounded above by $\psi(G)$ and below by $\psi(G) - \mid U_c \mid$.

A subset of edges of the graph G is called a *matching* if no two distinct edges in the subset share a common endpoint. Let $M = \{(u_1, v_1), \ldots, (u_k, v_k)\}$ be a matching with the sets of endpoints $X = \{u_1, \ldots, u_k\}$ and $Y = \{v_1, \ldots, v_k\}$. Then:

- M is said to be *independent* if it is the subgraph $G[X \cup Y]$. matching.
- M is said to be *semi-independent* if X and Y are independent sets, and the edges in M, *ordered as above*, respect the following additional property: for all $j > i \geq 1$, it holds that u_i is not adjacent to v_j.

Note that in a semi-independent matching, x_i may well be adjacent to y_j for $1 \leq j < i$. Hence, not every semi-independent matching is independent (although the converse is trivially true). A semi-independent matching can be used to obtain a complete coloring of the induced subgraph of its vertices as stated in the lemma below; a weaker version of this result, based on using an independent matching, is used in [CV01].

Lemma 3 [Mát81] *Let M be a semi-independent matching of size $\binom{t}{2}$ in G, and let $V(M)$ be the set of vertices in M. Then, an achromatic t-coloring of the subgraph $G[V(M)]$ can be computed efficiently.*

We now focus exclusively on bipartite graphs for the remainder of the paper. For independent sets of vertices U and V, we denote by $G(U, V, E)$ the bipartite graph G with bipartition (U, V) and edge set $E \subseteq U \times V$. For simplicity, we henceforth omit the original set of edges, E, from our notation since all our bipartite graphs are derived as induced subgraphs of the input graph G. For $U' \subseteq U$ and $V' \subseteq V$, we use the alternative notation $G[U', V']$ for the induced subgraph $G[U' \cup V']$ to make explicit the subsets of the original bipartition that induce the subgraph.

For any vertex $v \in V$ in the bipartite graph $G(U, V, E)$, the induced subgraph consisting of v and its neighbors, $N_G(v) \subseteq U$, is called the *star centered at* v in the graph. Now, suppose that U does not contain any isolated vertices. A simple iterative procedure that we will call the *star removal algorithm* can be used to compute an achromatic coloring of G as follows. In iteration $i \geq 1$ of the algorithm, we choose an arbitrary surviving vertex $v_i \in V$ of non-zero degree in the current graph. The star centered at v_i in the current graph is removed in the iteration along with all the other edges incident on the star's vertices. The resulting graph is used for the next iteration. Note that the surviving portion of U in this resulting graph after an iteration, contains isolated vertices if and only if there are no further edges left. When all the edges of G have been eliminated, we process the sequence of stars removed in successive iterations. If an arbitrary edge (u_i, v_i) is chosen from the ith star, it is not difficult to see that the the resulting sequence of edges, $M = \{(u_1, v_1), \ldots, (u_k, v_k)\}$, forms a semi-independent matching.

The number of deleted stars, k, equals the number of iterations before the residual graph is devoid of any edges. Letting $\Delta_G(V)$ denote the largest degree of any vertex in bipartition V of G, it follows that k must be at least $|U|/\Delta_G(V)$. In conjunction with Lemmas 1 and 3, we get the following result.

Lemma 4 *Let $G(U, V)$ be a bipartite graph with no isolated vertices in U. Then, the star removal algorithm produces an achromatic partition of size at least $\Omega(\sqrt{|U|/\Delta_G(V)})$.*

4 The Reducing Congruence and the Reduced Graph

Hell and Miller [HM76] define a very natural equivalence relation on the vertex set of any graph G. The relation, also called the *reducing congruence* of G [Edw97, HM97], is defined as follows: any pair of vertices of G are *equivalent* if and only if they have exactly the same set of neighbors in the graph.

We denote by $S_G(v)$, the equivalence class of vertex v under the reducing congruence for G (the subscript being dropped when G is clear from the context). Let q be the number of distinct equivalence classes under the reducing congruence for G. Assume that the vertices of G are indexed so that $S(v_1), \ldots, S(v_q)$ denote the distinct equivalence classes. Each member of the equivalence class $S(v_i)$ is called a *copy* of v_i. Note that, by definition, two equivalent vertices cannot be adjacent to each other in G, hence $S(v_i)$ is an independent set in G. The *reduced graph* of G, denoted G^*, is the graph induced by one copy from each of the equivalence classes. Equivalently, G^* is just the subgraph $G[\{v_i : 1 \leq i \leq q\}]$. Note that if v_i, v_j are adjacent in G^*, then the subgraph $G[S(v_i), S(v_j)]$ is a complete bipartite graph. The following two results can be shown.

Lemma 5 [KK01] *Let $S(v_1)$, $S(v_2)$, $\ldots$, $S(v_q)$ be the equivalence classes under the reducing congruence for graph G. Then, given any subset $U \subseteq \{v_i : 1 \leq i \leq q\}$, we can extend an achromatic k-coloring of $G^*[U]$ to an achromatic k-coloring of $G[U]$ in polynomial time. Hence, $\psi(G) \geq \psi(G^*)$.*

Theorem 1 [KK01] *Let G be a bipartite graph whose reducing congruence has q equivalence classes. Then, there is an efficient algorithm to compute a complete coloring of G with at least*

$$\min\{\psi(G)/q, \ \sqrt{\psi(G)}\}$$

colors. Hence, the achromatic number of a bipartite graph can be approximated to within a ratio of $O(\max\{q, \sqrt{\psi(G)}\})$.

We introduce a useful definition next. The *reduced degree* ${d^*}_G(v)$ of any vertex v in graph G is the degree of its copy in the reduced graph G^*. In other words, given the reducing congruence for G, the reduced degree is the maximum number of pairwise non-equivalent neighbors of a vertex in the graph G.

Lemma 6 *Let v, w be a pair of vertices of G such that $S_G(v) \neq S_G(w)$ and $d^*_G(w) \geq d^*_G(v)$. Then there is a vertex z adjacent to w but not to v.*

Proof. Assume the contrary, which implies that $N_G(w) \subseteq N_G(v)$. However, as $d^*_G(w) \geq d^*_G(v)$ it follows that $N_G(w) = N_G(v)$ and hence that $S_G(w) = S_G(v)$, a contradiction. □

5 Intuitive description of the algorithm

Our goal is to show that for any bipartite graph $G(U, V, E)$ with n vertices, we can find an achromatic partition of size at least $\psi(G)/O(n^{4/5})$. Let ψ^* be an upper bound on the true value of $\psi(G)$. Our approximation algorithm uses $\psi^*(G)$ to obtain an achromatic partition of an induced subgraph of G. The algorithm is guaranteed to produce a large number of colors in the partition when the value of ψ^* equals $\psi(G)$ so it suffices to simply run the algorithm for all possible values of $\psi(G)$ and use the best solution from among all the runs.

To explain the key intuition behind the algorithm, it is convenient to use terms like *small* and *large* in an informal sense to qualify the relative sizes of various sets. We postpone more precise characterizations of these terms, but observe here that by *small*, we mean of size roughly $O(n^{4/5})$ or n^δ for some $0 < \delta \leq 4/5$, and by *large*, we mean of size roughly $\omega(n^{4/5})$.

We may assume that G has no isolated vertices because such vertices have no effect on the achromatic number of G. Also, $\psi(G)$ may be assumed to be large, for otherwise even the achromatic coloring induced by the initial bipartition (U, V) will achieve a small ratio of approximation.

Next, consider the reducing congruence on G. Since G has no isolated vertices, its equivalence classes under the reducing congruence can be cleanly partitioned into those that are subsets of U (the U-equivalence classes) and those that are subsets of V (the V-equivalence classes). Let q_U (respectively, q_V) be the number of U-equivalence (respectively, V-equivalence) classes under the reducing congruence on G, and let $q = q_U + q_V$ be the total number of equivalence classes. If q were small, then Theorem 1 (via the algorithm of [KK01]) would guarantee a good approximation ratio for $\psi(G)$. Hence, we can assume that both q and $\psi(G)$ are both large, *i.e.* have magnitude $\omega(n^{4/5})$).

Since q is large, either q_U or q_V must be large. In any event, on average, an equivalence class under the reducing congruence has few vertices (roughly $O(n^{1/5})$). We call such classes the *light* equivalence classes. By the Markov inequality, there will only be a few heavy (*i.e.* not light) equivalence classes. The effect of those classes on our algorithm is negligible; for the sake of a simplified description, the maximum size of a heavy equivalence class is not pertinent.

The heart of the approximation algorithm is a subroutine, Ach-Bip, that takes as input a bipartite graph $G[U_0, V_0]$ and a guessed value ψ^* of the achromatic number to iteratively compute a sequence of color classes $A_1, A_2, \ldots A_k$. These color classes form an achromatic partition of $G[\cup_{1\le i\le k} A_i]$. Broadly speaking, in iteration i, $i \ge 1$, Ach-Bip works as follows:

- It starts with a subgraph $G_{i-1} = G[U_{i-1}, V_{i-1}]$
- If G_{i-1} has no light U_{i-1}-equivalence classes, then the subroutine call exits. Otherwise, a set, A_i, of independent vertices in G_{i-1} is computed with the following properties: A_i is relatively small in size, and covers a relatively large set $U_i \subseteq U_{i-1} \setminus A_i$. This ensures that color classes $A_{i+1}, \ldots$ that may be computed by future iterations are adjacent to A_i.
- If the removal of A_i and some related vertices from G_{i-1} would not reduce the (guessed) achromatic number significantly, then the next iteration is initiated on a subgraph $G_i = G[U_i, V_i] \subset G_{i-1}$.

The subroutine, Ach-Bip, is first executed on the graph $G(U, V)$. If q_U, the number of U-equivalence classes, is large then this subroutine call may produce a large enough collection of color classes. However, if the call exits because there are no light U_{i-1}-equivalence classes at the beginning of some iteration i (with i being a relatively small number), then we still have the possibility that q_V, the number of V-equivalence classes, is large (as explained earlier, either q_U or q_V must be large in the beginning).

Hence, the algorithm makes a second call on Ach-Bip. In this call, the input to the subroutine is the graph $G_{i-1} = G[V_{i-1}, U_{i-1}]$ that remains after the first call. Note in particular that the roles of U_{i-1} and V_{i-1} interchanged, *viz.* that the left and right sides of the bipartition are now taken to be V_{i-1} and U_{i-1} respectively. This second call may find a large enough achromatic partition of a subgraph of G_{i-1}. It is however possible that the second application of Ach-Bip also halts within a small number of iterations - small enough that we cannot get a good guarantee on the ratio of the number of color classes found in either of the two calls with respect to the actual achromatic number.

In this case, the residual graph that remains after the second application halts, has a small number of equivalence classes in its reducing congruence but has an achromatic number that, by design, is at most $\psi^*/2$ less than the original graph G. Provided that our guess, ψ^*, is close to the optimal value, Theorem 1 ensures that a large achromatic partition of the residual graph can be found. This coloring can be extended in the usual greedy manner to obtain a large ratio of approximating the achromatic number of G. This completes the informal overview of the algorithm.

6 Formal description of the algorithm

The approximation algorithm, Approx-Bip, is described below. As mentioned earlier, we may think of the algorithm as being executed once for each possible value of the guessed achromatic number, ψ^*. The best solution from all the runs is used. The overall runtime may be improved by a logarithmic factor by deploying binary search over the possible range of values of ψ^*.

Input: $G(U, V)$, a bipartite graph; ψ^*, a positive integer
Output: An achromatic partition of G

1 $\mathcal{A}_1 \leftarrow$ the achromatic partition returned by the call Ach-Bip($G(U, V)$, ψ^*). Let $G^{[1]} = G[U^{[1]}, V^{[1]}]$ be the induced subgraph that remains when the procedure call halts ;

2 $\mathcal{A}_2 \leftarrow$ the achromatic partition returned by the call Ach-Bip($G[V^{[1]}, U^{[1]}]$, ψ^*). Note that the roles of the bipartitions are *interchanged*. Let $G^{[2]} = G[U^{[2]}, V^{[2]}]$ be the induced subgraph that remains when this second application of the procedure halts ;

3 If either of the achromatic partitions $\mathcal{A}_1$ or $\mathcal{A}_2$ is of size at least $\psi^*/(16 \cdot n^{1/5})$, then the corresponding achromatic coloring is extended to a complete achromatic coloring of G which is returned as final output ;

4 Otherwise, apply the algorithm of Theorem 1 on the subgraph $G^{[2]}$. The achromatic coloring thereby obtained can be extended to a complete achromatic coloring of G which is returned as final output.

Algorithm 1: Approx-Bip

We now provide a few notational abbreviations that simplify the formal description of procedure Ach-Bip and the subsequent analysis of the algorithm. A set is called **heavy** if it contains at least $n^{1/5}$ vertices. Otherwise, it is called **light**. In the bipartite graph $G(U, V)$, a vertex $v \in V$ is said to be U**-heavy** if the reduced degree of v is at least $n^{1/5}$.

Definition 1 *Starting with a subgraph G_0 of the graph G, let $G_0 \supset G_1 \ldots \supset G_i$ be a sequence of induced subgraphs of G obtained by successively removing vertices (and their adjacent edges). Let ψ^* be a positive integer. Then, for any $i \geq 1$, the deletion of some set of vertices S_i from G_i is said to be ψ^*-**safe** for G_i if the total number of non-isolated vertices (including those in S_i) removed from the initial subgraph G_0 is at most $\psi^*/4$.*

Definition 1 is critical to the description of the subroutine Ach-Bip that appears next.

Input: A bipartite graph $G_0(U_0, V_0)$, and a *guessed* achromatic number ψ^*
Output: An achromatic partition $\{A_1, A_2, \ldots\}$ of the induced graph $G_0[\cup_i A_i]$

if $\psi^* < 8 \cdot n^{4/5}$ **then**
  return $\mathcal{A} = \{U_0, V_0\}$
end
$\mathcal{A} = \emptyset$;
for $i = 1, 2, \ldots$ **do**
  if *there are no light U_{i-1}-equivalence classes in G_{i-1}* **then** /* Stop Condition 1 */
    return $\mathcal{A}$
  end
  $u \leftarrow$ a vertex in U_{i-1} with minimum reduced degree in G_{i-1} ;
  $U', V', G' \leftarrow U_{i-1} \setminus S_{G_{i-1}}(u),\ V_{i-1} \setminus N_{G_{i-1}}(u),\ G[U', V']$;
  $C_i \leftarrow \emptyset$
  while $(U' \neq \emptyset)$ ***and*** *$\exists$ a U'-heavy vertex in V'* **do**
    $v \leftarrow$ a U'-heavy vertex in V' with maximum reduced degree in G' ;
    Add v to C_i ;
    $U', V', G' \leftarrow U' \setminus N_{G'}(v),\ V' \setminus \{v\},\ G[U', V']$
  end
  $q' \leftarrow$ the number of U'-equivalence classes in G' ;
  if $q' > n^{3/5}$ **then** /* Stop Condition 2 */
    return *the partition obtained by applying the star removal algorithm to G'*
  end
  $D_i \leftarrow \{w \in U' \mid S_{G'}(w)$ is a light equivalence class$\}$;
  for *every heavy U'-equivalence class $S_{G'}(w)$* **do**
    add an arbitrary neighbor of $S_{G'}(w)$ to C_i
  end
  $A_i \leftarrow S_{G_{i-1}}(u) \cup C_i$;
  $L_i \leftarrow$ the set of isolated vertices in the graph $G[U_{i-1} \setminus A_i, V_{i-1} \setminus A_i]$;
  if *it is* ***not*** ***ψ^*-safe*** *to delete* $(A_i \cup D_i \cup L_i)$ **then** /* Stop Condition 3 */
    return $\mathcal{A}$
  end
  add A_i to $\mathcal{A}$;
  $U_i, V_i, G_i \leftarrow U_{i-1} \setminus (A_i \cup D_i \cup L_i),\ V_{i-1} \setminus (A_i \cup D_i \cup L_i),\ G[U_i, V_i]$
end

Procedure Ach-Bip

6.1 The approximation ratio

We now analyze the approximation ratio of Algorithm Approx-Bip. Our goal is to show that the approximation ratio is bounded by $O(n^{4/5})$. The analysis is conducted under the assumption that $\psi(G) \geq 8 \cdot n^{4/5}$. Otherwise, returning an arbitrary achromatic partition (say, the original bipartition of size 2), as done in line 3 of procedure Ach-Bip, trivially gives an $O(n^{4/5})$ ratio.

We start by observing that the execution of the **for** loop starting at line 5 in procedure Ach-Bip, could halt in one of *three mutually exclusive ways* during some iteration $(k+1) \geq 1$:

Stop Condition 1: At the beginning of the iteration,there are no light U_k-equivalence classes in G_k.

Stop Condition 2: The star removal algorithm can be applied during the iteration.

Stop Condition 3: Just prior to the end of the iteration, it is found that the current deletion of $(A_{k+1} \cup D_{k+1} \cup L_{k+1})$ is not ψ^*-safe for G_k.

Note that the induced subgraphs G_i $(i \geq 1)$ form a monotone decreasing chain. If the star removal algorithm (stop condition 2) cannot be applied during any iteration, then eventually one of the other two conditions must hold since we keep removing vertices and edges during each iteration. This guarantees that procedure Ach-Bip *will* eventually halt. It remains to analyze the approximation ratio under each of the three stop conditions.

The schematic shown in Figure 1 depicts the various sets computed during iteration $i \geq 1$ of procedure Ach-Bip. We say that iteration $i \geq 1$ is *successful* if none of the stop conditions are triggered during the iteration, *i.e.* the procedure commences the next iteration with the surviving subgraph G_i. Suppose that the first k iterations are successful and let $(k+1) \geq 1$ be the first unsuccessful iteration of the procedure.

Lemma 7 *If procedure Ach-Bip halts during iteration $(k+1)$ under the stop condition 2 above, then the achromatic partition returned has size at least $n^{1/5}$. As $\psi(G) \leq n$, an $O(n^{4/5})$-ratio is derived.*

Proof. During iteration $(k+1)$, consider the graph $G^{'} = G[U^{'}, V^{'}]$ on which the star removal procedure is applied on line 19 of the procedure. Let u be the vertex chosen during the iteration on line 9 of the procedure. By construction, we observe that

$$\begin{aligned} U^{'} &= U_k \setminus (S(u) \cup N(C_{k+1})) \\ V^{'} &= V_k \setminus (N(u) \cup C_{k+1}) \end{aligned}$$

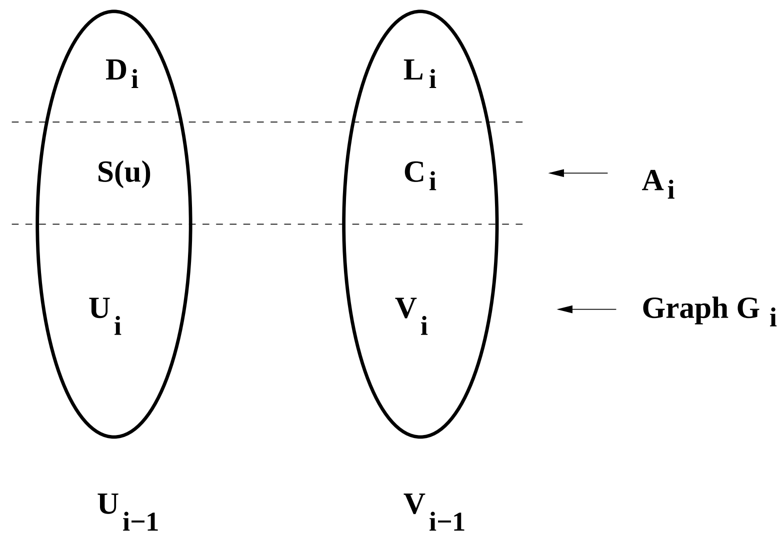

Figure 1: The graph G_{i-1}

where $S(u)$ and $N(u)$ are with respect to the graph G_k that survives at the end of iteration k. Now, C_{k+1} is the collection of large reduced-degree non-neighbors of u collected in the inner loop during the iteration. Hence, for any vertex $w \in U^{'}$, it holds that $S(w) \neq S(u)$. Using Lemma 6 and the fact that w does not belong to $N_{G'}(C_{k+1})$, we conclude that w must have at least one neighbor in $V^{'}$ and hence, that $U^{'}$ contains no isolated vertices in $G^{'}$.

Furthermore, the inner loop condition (line 12) guarantees that every vertex in $V^{'}$ is adjacent to at most $n^{1/5}$ $U^{'}$-equivalence classes in $G^{'}$. When the star removal algorithm is applied, $q^{'}$ (the number of $U^{'}$-equivalence classes) is at least $n^{3/5}$. From the discussion preceding Lemma 4, it is easy to see that the star removal algorithm will produce a collection of at least

$$\sqrt{\frac{n^{3/5}}{n^{1/5}}} = n^{1/5}$$

stars, and hence an achromatic partition of size at least $n^{1/5}$ can be returned as claimed. □

Turning now to stop condition 3, we show that if the procedure halts during iteration $(k+1)$ because a ψ^*-unsafe deletion is flagged, then k, the number of classes in the achromatic partition $\mathcal{A}$ computed thus far, must already be large enough. To this end, we establish a sequence of claims.

Claim 1 *For all i such that $1 \leq i \leq k$, the set A_i is an independent set and is adjacent to A_j for every $j \in [i+1, k]$. In other words, $\mathcal{A}$ is an achromatic partition of the subgraph $G[\cup_{1 \leq i \leq k} A_i]$.*

Proof. We first verify that at the end of a successful iteration i, the set of vertices A_i is an independent set. By construction, $A_i = S_{G_{i-1}}(u)) \cup C_i$ where u is the vertex chosen on line 9. The vertices in $S_{G_{i-1}}(u)$ are mutually non-adjacent by definition. Moreover, $C_i \subseteq V_{i-1} \setminus N_{G_{i-1}}(u)$ and hence C_i is an independent set that is not adjacent to $S_{G_{i-1}}(u)$. Thus, A_i is independent as well.

Now, by construction, the vertices retained in the set U_i at the end of the iteration, are exactly those that are *covered* by some vertex in $C_i \subset A_i$. The set A_j, for $i < j \leq k$, contains at least one vertex in $U_{j-1} \subset U_i$. Hence there is always an edge between A_i and A_j. □

Claim 2 *For all i such that $1 \leq i \leq k$, the size of the set $(A_i \cup D_i)$, just prior to executing the safety check on line 27, is bounded by $4n^{4/5}$.*

Proof. By construction, $A_i = S_{G_{i-1}}(u) \cup C_i$ prior to executing line 27. We know that $S_{G_{i-1}}(u)$ is a light equivalence class and hence, $\mid S_{G_{i-1}}(u) \mid < n^{1/5}$. A vertex $v \in V_{i-1}$ is added to C_i either during the inner loop (line 12) or later, if it happens to be adjacent to a heavy $U^{'}$-equivalence class (line 23).

In the former case, just prior to v being added to C_i, v must have been adjacent to at least $n^{1/5}$ pairwise non-equivalent vertices in $U^{'}$. These vertices (along with their copies) are removed from $U^{'}$ after v is added to C_i and before the next iteration of the inner loop commences. In other words, each vertex of $U^{'}$ eliminated in the inner loop corresponds to exactly one vertex in C_i that causes its elimination. Since the initial size of $U^{'}$ is bounded by n, it follows that no more than $n/n^{1/5} = n^{4/5}$ vertices could have been added to C_i during the execution of the inner loop.

The number of vertices, added to C_i because they are witness to being adjacent to some heavy $U^{'}$-equivalence class (see line 23), is at most the number of heavy $U^{'}$-equivalence classes. This latter quantity is bounded above by the total number of $U^{'}$-equivalence classes. Since $U^{'}$ has less than $n^{3/5}$ classes (otherwise, the star removal algorithm would have been used), it follows that at most $n^{3/5}$ vertices are added to C_i in line 23. Thus, prior to executing line 27, there are at most

$$n^{1/5} + n^{4/5} + n^{3/5} \quad \leq \quad 3 \cdot n^{4/5}$$

vertices in A_i.

$U^{'}$ has at most $n^{3/5}$ light equivalence classes when control reaches line 21. Since the vertices in D_i just prior to executing the safety check are simply those belonging to such light $U^{'}$-equivalence classes, the number of vertices in D_i is at most $n^{1/5} \cdot n^{3/5} = n^{4/5}$. Summing up, we see that $(A_i \cup D_i)$ contains at most $4n^{4/5}$ vertices when it is tested for ψ^*-safety on line 27. □

Claim 3 *If the first k iterations are successful, then the difference, $\psi(G_0) - \psi(G_k)$, is at most $4k \cdot n^{4/5}$.*

Proof. Consider the graph G_i at the end of the i^{th} successful iteration. For clarity, we can view the construction of G_i from G_{i-1} as taking place in two stages. First, the set of vertices $(A_i \cup D_i)$ is removed from G_{i-1} giving us an intermediate graph G_i^-. Then L_i, the set of all isolated vertices in G_i^- (see line 26 in procedure Ach-Bip), are deleted from G_i^- yielding G_i.

Using Lemma 2 and Claim 2 above, it follows that $\psi(G_{i-1}) - \psi(G_i^-) \le 4n^{4/5}$. Since L_i is an isolated set of vertices in G_i^-, their removal from G_i^- has no effect on the achromatic number. Therefore, it holds that $\psi(G_i^-) = \psi(G_i)$ and hence

$$\text{forany} 1 \le i \le k, \quad \psi(G_{i-1}) - \psi(G_i) \le 4n^{4/5}.$$

The telescoping sum of the above k inequalities, one per successful iteration, yields

$$\psi(G_0) - \psi(G_k) \le 4kn^{4/5}$$

as claimed. □

Lemma 8 *If procedure Ach-Bip halts during iteration* $(k+1)$ *under the stop condition 3 above, then the achromatic partition returned has size at least* $\lfloor \psi^*/16n^{4/5} \rfloor$.

Proof. Since the first k iterations were successful, it follows that for each $i \in [1, k]$, it is safe to delete $(A_i \cup D_i \cup L_i)$. However, it is unsafe to delete $(A_{k+1} \cup D_{k+1} \cup L_{k+1})$ and by Definition 1 and Claim 3, this can only happen if

$$4(k+1)n^{4/5} \;>\; \psi^*/4.$$

Hence $\mathcal{A} = \{A_1, A_2, \dots A_k\}$, which is an achromatic partition of the subgraph $G[\cup_{1 \le i \le k} A_i]$ by Claim 1, has size $k \ge \lfloor \psi^*/(16n^{4/5}) \rfloor$. Applying Lemma 1, we conclude that a complete achromatic coloring of G with at least $\lfloor \psi^*/(16n^{4/5}) \rfloor$ colors can be computed. □

We now address the stop condition 1 in procedure Ach-Bip. If the procedure halts on this stop condition in iteration $(k+1)$, then we have two possibilities. If $k \ge \lfloor \psi^*/(16n^{4/5}) \rfloor$, a sufficiently large partition has been found (obvious from the preceding discussion) and we are done. Otherwise, $k < \lfloor \psi^*/(16n^{4/5}) \rfloor$ and we do not necessarily have a good guarantee of an approximation ratio.

However, note that G_k, the graph at the beginning of iteration $(k+1)$, has no light U_k-equivalence classes which is what triggers the stop condition. Hence, U_k has no more than $n^{4/5}$ equivalence classes (each heavy class has at least $n^{1/5}$ vertices and $| U_k | \le n$) that are all heavy.

Claim 4 *Assume that both applications of procedure Ach-Bip on lines 1 and 2 of algorithm Approx-Bip halt on stop condition 1 of procedure Ach-Bip. Let* q_1 *(respectively,* q_2*) be the number of* $U^{[1]}$*-equivalence classes in* $G^{[1]}$ *(respectively, the number of* $U^{[2]}$*-equivalence classes in* $G^{[2]}$*). Then, the* *graph $G^{[2]}$ has achromatic number at least $\psi(G) - \psi^*/2$ and has at most a total of $(q_1 + q_2) \leq 2n^{4/5}$ equivalence classes.*

Proof. Observe that the removal of vertices (along with all their incident edges) from a graph cannot increase the number of equivalence classes: two vertices that were equivalent before the removal, remain equivalent afterwards. Hence, the number of $V^{[2]}$ equivalence classes is at most q_1 (note that the partitions are interchanged before the second application of procedure Ach-Bip on line 2). Thus $G^{[2]}$ has at most a total of $(q_1 + q_2)$ equivalence classes. The discussion preceding the claim shows that $(q_1 + q_2)$ is bounded above by $2n^{4/5}$.

Since neither application was halted by stop condition 3, the vertices deleted during both applications were ψ^*-safe for deletion. Hence, the net decrease in the achromatic number is at most $2\psi^*/4 = \psi^*/2$. □

Theorem 2 *For at least one value of ψ^*, viz. when $\psi^* = \psi(G)$, Algorithm Approx-Bip achieves an approximation ratio of $O(n^{4/5})$.*

Proof. Lemma 7 shows that if either of the two applications of procedure Ach-Bip halt on the stop condition 2, then we are guaranteed an approximation ratio of $O(n^{4/5})$.

The same ratio is obtained from Lemma 8 if either of the two applications of procedure Ach-Bip halt on the stop condition 3 when $\psi^* = \psi(G)$.

Otherwise, if $\psi^* = \psi(G)$ and both applications of procedure Ach-Bip on lines 1 and 2 of algorithm Approx-Bip halt on stop condition 1, then from Claim 4 we see that the residual graph $G^{[2]}$ as an achromatic number that is at least $\psi(G)/2$. An application of the algorithm of Theorem 1 on graph $G^{[2]}$ (see line 4 of algorithm Approx-Bip) provides an $O(\max\{n^{4/5}, \sqrt{\psi(G)}\}) = O(n^{4/5})$ approximation ratio for the achromatic number of $G^{[2]}$ since the number of equivalence classes of $G^{[2]}$ is $O(n^{4/5})$ by Claim 4. The achromatic coloring of $G^{[2]}$ can be extended greedily to one of G with approximation ratio $O(n^{4/5})$.

□

Acknowledgment

It may hold that an $n^{1-\epsilon}$ ratio approximation algorithm (for any constant $\epsilon > 0$) is impossible for general graphs (unless, say, $P = NP$). After all, an $\Omega(n^{1-\epsilon})$ inapproximability result does exist for the maximum independent set problem [Has99] and the achromatic number problem and the maximum independent set problem are after all closely related.

The first author would like to thank Robert Krauthgamer and Magnus M. Halldorsson for useful discussions. The authors are also grateful to the anonymous referees for suggestions that helped to improve the presentation.

References

[Bod89] H. L. Bodlaender. Achromatic number is NP-complete for cographs and interval graphs. *Information Processing Letters*, 31(3):135–138, 1989.

[CE97] N. Cairnie and K. Edwards. Some results on the achromatic number. *J. Graph Theory*, 26(3):129–136, 1997.

[CV01] A. Chaudhary and S. Vishwanathan. Approximation algorithms for the achromatic number. *J. Algorithms*, 41(2):404–416, 2001.

[Edw97] K. Edwards. The harmonious chromatic number and the achromatic number. In R. A. Bailey, editor, *Surveys in Combinatorics, 1997 (Invited papers for 16th British Combinatorial Conference)*, number 241 in LMS Lecture Notes, pages 13–47. Cambridge University Press, 1997.

[Hal93] M. M. Halldórsson. Approximating the minimum maximal independence number. *Information Processing Letters*, 46(4):169–172, 1993.

[Has99] J. Hastad. Clique is hard to approximate within n to the power $1-\epsilon$. *Acta Mathematica*, 182:105–142, 1999.

[HM76] P. Hell and D. J. Miller. On forbidden quotients and the achromatic number. In C. St. J. A. Nash-Williams & J. Sheehan, editor, *Proc.5th British Combinatorial Conference (1975)*, volume 15 of *Congressus Numerantium*, pages 283–292, 1976.

[HM97] F. Hughes and G. MacGillivray. The achromatic number of graphs: a survey and some new results. *Bulletin of the Institute of Combinatorics and its Applications*, 19:27–56, 1997.

[KK01] G. Kortsarz and R. Krauthgamer. On the approximation of the achromatic number. *SIAM J.of Discrete Mathematics*, 14(3):408–422, 2001.

[KL99] P. Krysta and K. Loryś. Efficient approximation algorithms for the achromatic number. In *Proc.7th European Symposium on Algorithms (ESA)*, number 1643 in LNCS, pages 402–413. Springer, Berlin, 1999.

[KS03] G. Kortsarz and S. Shende. Approximating the achromatic number problem on bipartite graphs. In *Proc.11th European Symposium on Algorithms (ESA)*, number 2832 in LNCS, pages 385–396. Springer, Berlin, 2003.

[Mát81] A. Máté. A lower estimate for the achromatic number of irreducible graph s. *Discrete Mathematics.*, 33(2):171–183, 1981.

[MFM86] P. Hell M. Farber, G. Hahn and D. Miller. Concerning the achromatic number of graphs. *J. Combin. Theory, Series B*, 40(1):21–39, 1986.

[MMHS05] J. Radhakrishnan M. M. Halldórsson, G. Kortsarz and S. Sivasubramanian. Complete partitions of graphs. In *Proc.Symp.on Discrete Algorithms (SODA)*, pages 860–869, 2005.

[UFS02] G. Kortsarz U. Feige, M. Halldórsson and A. Srinivasan. Approximating the domatic number. *SIAM J. Computing*, 32(1):172–195, 2002.

[YG80] M. Yannakakis and F. Gavril. Edge dominating sets in graphs. *SIAM J. Appl. Math*, 38(3):364–372, 1980.