Models and Approximation Algorithms for Channel Assignment in Radio Networks

Sven O. Krumke [a,*] Madhav V. Marathe [b,**] S. S. Ravi [c,***]

[a] *Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustr. 7, 14195 Berlin, Germany. Email:* `krumke@zib.de`.
[b] *P.O. Box 1663, MS M997, Los Alamos National Laboratory, Los Alamos, NM 87545. Email:* `marathe@lanl.gov`.
[c] *University at Albany – State University of New York, Albany, NY 12222. Email:* `ravi@cs.albany.edu`.

We consider the frequency assignment (broadcast scheduling) problem for packet radio networks. Such networks are naturally modeled by graphs with a certain geometric structure. The problem of broadcast scheduling can be cast as a variant of the vertex coloring problem (called the *distance-2 coloring* problem) on the graph that models a given packet radio network. We present efficient approximation algorithms for the distance-2 coloring problem for various geometric graphs including those that naturally model a large class of packet radio networks. The class of graphs considered include (r, s)-civilized graphs, planar graphs, graphs with bounded genus, etc.

Keywords: Channel Assignment, Radio Networks, Approximation Algorithms, Geometric Graphs.

AMS Subject classification: 68Q25

1. Introduction

1.1. Radio Networks and the Channel Assignment Problem

A *radio network* consists of a group of *transceivers* (also known as *stations*) sharing a common radio channel and communicating with each other using this channel. Examples of such networks include packet radio networks, cellular phone networks and satellite networks. Each transceiver has a range (a geographical region) within which it can communicate with other transceivers. In a multi-hop radio network, messages are transmitted via a series of intermediate transceivers. This allows radio networks to reuse frequencies in transmitting data. Specifically, in a static transmission protocol, an a priori assignment of communication resources such as time, frequency or code is made to the transceivers in a manner that avoids contention. In a dynamic protocol, resources are allocated on-the-fly. Examples of protocols that use static assignment of resources include Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA). Examples of dynamic resource allocation protocols include ALOHA and CSMA. Static protocols are commonly used to handle steady streams of messages while the dynamic protocols are used to handle situations with bursty message traffic. In each of the scenarios, the eventual goal of resource assignment is twofold: (a) messages must be transmitted without collision (or conflict) and (b) the resources must be used efficiently.

In the case of FDMA (TDMA), spatial locations of the transceivers allow more than one transceiver to communicate using the same frequency (during the same time slot). Two transceivers may use the same frequency (or the same time slot) only if the generated signals do not *collide*. Depending on the signaling mechanism, transmissions may collide in two ways, referred to as *primary* and *secondary* interferences. Primary interference occurs when a station is involved in more than one communication task at the same time (sending and receiving, receiving from two different transmitters, etc.). Secondary interference occurs when a receiver R is tuned to a particular transmitter T but also unwantedly receives signals from another transmitter. As a result, the intended message from T to R may get garbled. The *channel assignment problem* involves the assignment of channels (i.e., time slots or frequencies) to transmitters or the connecting links so that all the transmitters (links) assigned the same channel may transmit in a collision-free fashion. In this paper, we consider the channel assignment problem for transmitters. The transmission of a station T is intended for and must be received collision-free by *all* the neighbors

[*] Research supported by the German Science Foundation (DFG, grant Gr 883/5-3)

[**] This work was supported by the Department of Energy under Contract W-7405-ENG-36.

[***] Research supported by NSF Grant CCR-97-34936.

of T (i.e., the receivers which are within the range of T). As pointed out in [25], only primary interference can arise in this situation. Thus, two stations cannot be assigned the same time slot if one station is within the range of the other or the two stations have a common neighbor (i.e., another station located in the intersection of the ranges of the two stations). The objective of the channel assignment problem is to allow the transceivers to communicate with each other without interference while using a minimum number of channels.

As observed in [25,27], the above channel assignment problem corresponds to the *distance-2 vertex coloring* problem in the graph that models the packet radio network. This undirected graph has one node for each transceiver; for each pair of transceivers such that one transceiver is in the range of the other, there is an edge between the corresponding nodes. In this graph, a valid distance-2 coloring must assign different colors to any pair of nodes between which there is a path of length at most 2. Since each color corresponds to a channel, it is important to produce a distance-2 coloring that uses a minimum number of colors. If G denotes the graph modeling the network, then the distance-2 coloring problem on G is equivalent to the standard minimum vertex coloring problem on G^2, where G^2 has the same vertex set as G and there is an edge between two vertices of G^2 if and only if there is a path of length at most 2 between the vertices in G.

The distance-2 coloring problem also arises in approximating the Hessian matrices of certain nonlinear functions using a minimum number of gradient evaluations [16,18]. McCormick [18] has shown that the distance-d coloring problem is NP-complete for any fixed $d \geq 2$. He has also shown that the simple greedy algorithm (i.e., consider the vertices in any order; the color assigned to vertex v_i is the smallest color that has not been used by any node which is within a distance of at most 2 from v_i) provides a performance guarantee of $O(\sqrt{n})$ for the distance-2 coloring problem for any graph G with n vertices. An approximation algorithm with a performance guarantee of $O(\theta)$, where θ is the *thickness*[1] of the graph, appears in [25]. Reference [24] presents a framework that captures a variety of channel assignment problems and a general approximation algorithm that is applicable to all such problems.

1.2. Summary of Results

Here we consider various coloring problems motivated by packet radio networks when instances are restricted to special classes of graphs[2]. The results obtained in this paper include the following.

1. An important class of graphs that can model a large class of packet radio networks is that of (r, s)-civilized graphs (see Section 3). For this class of graphs, we give a 2-approximation algorithm for the distance-2 coloring problem. We note that the performance guarantee of our algorithm is *independent* of the values of r and s. The class of (r, s)-civilized graphs includes intersection graphs of circles whenever there is a (fixed) minimum separation between the centers of any pair of circles. An approximation algorithm with a performance guarantee of 14 for intersection graphs of circles is given in [28]. It has been shown recently [19] that the performance guarantee of 14 can also be achieved using the greedy algorithm. Reference [28] also presents an approximation algorithm with a performance guarantee of 7 when all the circles have the same radius; such graphs are known as *unit disk graphs* [8]. (Approximation algorithms for the usual vertex coloring problem for unit disk graphs have been presented in [11,17].)
2. The distance-2 coloring problem is known to be NP-complete even for planar graphs of bounded degree [23]. An approximation algorithm with a performance guarantee of at most 9 for the distance-2 coloring problem for all planar graphs was given in [25]. For planar graphs of bounded degree, we show that there is a 2-approximation algorithm for the distance-2 coloring problem.
3. We observe that the approximation algorithm for the distance-2 coloring problem for planar graphs given in [25] is applicable to a more general class of graphs called q-inductive graphs and leads to a performance guarantee of at most $2q - 1$ for such graphs. A number of graph classes including bounded genus graphs, treewidth bounded graphs and intersection graphs of d-dimensional k-ply-neighborhood systems [20] are q-inductive for appropriate values of q.

We also observe that for graphs that are simultaneously treewidth bounded and degree bounded, the distance-2 col-

---

[1] The *thickness* of a graph $G(V, E)$ is the minimum number of subsets into which the edge set E must be partitioned so that each subset in the partition forms a planar graph on V.

[2] Definitions of the special classes of graphs considered in this paper appear in subsequent sections of this paper.

oring problem can be solved in polynomial time. An interesting aspect of our work is that the approximation algorithms for (r, s)-civilized graphs and bounded degree planar graphs are based on the same technique.

The remainder of this paper is organized as follows. Section 2 presents the necessary definitions. Section 3 contains a discussion of how the class of (r, s)-civilized graphs provides a reasonable model for broadcast networks in practice. Section 4 presents the main ideas behind the approximation algorithms developed in this paper. Sections 5, 6 and 7 present approximation algorithms for (r, s)-civilized graphs, bounded degree planar graphs and q-inductive graphs respectively. Some concluding remarks are presented in Section 8.

2. Definitions and Preliminaries

In the remainder of this paper we assume without loss of generality that all given graphs are connected. Otherwise, each connected component of the graph can be colored separately.

Definition 2.1. Let $G(V, E)$ be an undirected graph. Then the *square of* G, denoted by $G^2 = (V, E^2)$, contains the edge (u, v) if and only if there is a path consisting of at most two edges between u and v in G.

Definition 2.2. Let $G(V, E)$ be an undirected graph. A *vertex coloring* of G is a mapping $f \colon V \to \{1, \ldots, k\}$ for some integer $k \in \mathbb{N}$ such that for all $(u, v) \in E$ we have $f(u) \neq f(v)$. The number k is referred to as the *number of colors* in the coloring f.

A *D2-vertex coloring* of a graph G is a mapping $f \colon V \to \{1, \ldots, k\}$ such that $f(u) \neq f(v)$ whenever there is a path consisting of at most two edges between u and v in G.

As noted in Section 1.1, a D2-vertex coloring of G can be viewed equivalently as a vertex coloring of the square graph G^2. We are now ready to formulate the two optimization problems that are the main subject of this paper.

Definition 2.3. An instance of MINCOLOR (MIND2COLOR) is given by an undirected graph $G(V, E)$. The problem is find a vertex coloring (D2-vertex coloring) of G with the minimum number of colors.

The notion of *treewidth* for a graph (see for example [2–4,26]) is standard in the literature on graph algorithms.

Definition 2.4. For any $k \in \mathbb{N}$ the class of *k-trees* is defined recursively as follows.

(i) A clique of size $k + 1$ is a k-tree.

(ii) A k-tree with $n + 1$ vertices is obtained from a k-tree with n vertices by adding a new vertex and edges from the new vertex to a set of k completely connected vertices.

Any subgraph of a k-tree is called a *partial k-tree*. The minimum value of k for which a graph is a partial k-tree is called the *treewidth* of the graph.

A graph family is *treewidth bounded* if there is a constant k such that each graph in the family has treewidth at most k. An alternative approach to defining treewidth is via *tree decompositions* [26]. This leads to an equivalent definition which is easier to use in deriving some of our results.

Definition 2.5. A *tree decomposition* of a graph $G(V, E)$ is a pair (X, T) where $X = \{ X_i : i \in I \}$ is a collection of subsets of V and $T = (I, F)$ is a tree such that all the following conditions hold:

(i) $\bigcup_{i \in I} X_i = V$.

(ii) For every edge $e = (u, v) \in E$, there is a subset X_j such that both u and v are in X_j.

(iii) For all $v \in V$, the set $\{ i \in I : v \in X_i \}$ induces a (connected) subtree of T.

Condition (iii) of the above definition can be replaced by the following equivalent condition: for all $i, j, k \in I$, if j lies on the path in T from i to k, then $X_i \cap X_k \subseteq X_j$ [26].

Given a tree decomposition (X, T) for graph G, the *width* of the decomposition is $\max_{i \in I}(|X_i| - 1)$. The treewidth of G is the minimum treewidth over all tree decompositions. As noted above, this definition of treewidth is equivalent to the definition based on k-trees (see for example [2]).

Many optimization problems which are NP-hard on general graphs become efficiently solvable for the class of treewidth bounded graphs (see for example [2]). In particular, we use the following result of Bodlaender [6]:

Theorem 2.6. For each fixed $k \geq 0$, the MINCOLOR problem for graphs with n nodes and treewidth at most k can be solved in $O(n)$ time. □

We recall some well known facts about vertex coloring.

Fact 2.7. The vertices of any graph with n nodes and maximum node degree Δ can be colored using at most $\Delta + 1$ colors in $O(n^2)$ time . □

Fact 2.8. Given a graph G of maximum node degree Δ, every proper D2-coloring of G requires at least $\Delta + 1$ colors. □

The focus of this paper is on polynomial time approximation algorithms (also referred to as *heuristics*) with provably good performance guarantees. The definition of an approximation algorithm is given below.

Definition 2.9. A polynomial-time algorithm, A, is said to be a *ρ-approximation algorithm* for a minimization problem Π, if for every problem instance I with optimal solution value $\mathsf{opt}(I)$, the solution $A(I)$ returned by the algorithm satisfies $A(I) \leq \rho \cdot \mathsf{opt}(I)$. The factor ρ is called the *performance guarantee* of the algorithm.

It should be noted that the performance guarantees provided by most of our approximation algorithms are constants, independent of the size of the input graph.

Using Facts 2.7 and 2.8 one easily obtains the following.

Observation 2.10. Let Δ be an integer constant. For any graph with n nodes and maximum node degree Δ, there are $O(n^2)$ time approximation algorithms with a performance guarantee of at most Δ for both MINCOLOR and MIND2COLOR problems. □

3. Graph Theoretic Models and Justification

References [25,27] considered restricted classes of graphs that can model radio networks in practice. Specifically, Sen and Huson [27] argue that circle intersection graphs[3] are a more realistic model for packet radio networks. While their assertion is valid for packet radio networks, intersection graphs of other geometric objects (e.g. regular polygons) also seem to be appropriate for modeling cellular networks and satellite networks. Here, we propose general graph theoretic models for channel assignment in packet radio networks. Our models are intersection graphs of *k-ply-neighborhood systems* (see for example [10,20,21,29,30]) and *(r, s)-civilized graphs*.

Following [30], a *neighborhood* of a point $p \in \mathbb{R}^d$ is a closed ball of positive radius centered at p. The point p itself is called the *center* of the neighborhood. A *neighborhood system* $\mathcal{N} = \{B_1, B_2, \ldots B_n\}$ is a finite collection of neighborhoods. For integers $k, d > 0$, we say that $\mathcal{N}$ is a *k-ply-neighborhood system in d-dimensions* if no point of $\mathbb{R}^d$ is strictly interior to more than k of the balls.

Definition 3.1. The *intersection graph of a k-ply-neighborhood system* is a graph in which each vertex corresponds to a neighborhood and there is an edge between two vertices if and only if the corresponding neighborhoods have a nonempty intersection.

An important subclass of the above intersection graphs are the *(r, s)-civilized graphs* [30].

Definition 3.2. For each fixed pair of real values $r > 0$ and $s > 0$, a graph G can be drawn in $\mathbb{R}^d$ in an *(r, s)-civilized manner* if its vertices can be mapped to points in $\mathbb{R}^d$ so that the length of each edge is $\leq r$ and the distance between any two points is $\geq s$.

A *planar (r, s)-civilized graph* is an (r, s)-civilized graph whose vertices can be embedded in the Euclidean plane ($\mathbb{R}^2$) such that no two edges intersect (except at their endpoints).

A civilized layout of a graph that can be drawn in a civilized manner in $\mathbb{R}^d$ consists of the coordinates of the vertices in $\mathbb{R}^d$ and the set of edges in the graph. We assume throughout this paper that the dimension (d) of the Euclidean space considered is at least 2. Graphs drawn in a civilized manner have been studied in the context of random walks by Doyle and Snell [9] and in the context of finite element analysis by Vavasis [31].

Both of the above classes are reasonable models for several classes of packet radio networks when considering the channel assignment problem. To see this, consider packet radio networks in which the range of any transmitter can be considered as a circular region with the transmitter at the center of the circle. Let r denote the radius of the region corresponding to a transmitter with the maximum range. Further, it is natural to assume a minimum separation s between any pair of transmitters since the equipment carrying the transmitters cannot be colocated. Clearly, the graphs that model such packet radio networks belong to the class of (r, s)-civilized graphs. In many other realistic situations, the ratio of maximum to the minimum transmitter range is not

---

[3] Circle intersection graphs are intersection graphs of circles (of arbitrary radii) in which we include an edge between a pair of vertices only when the corresponding circles intersect.

fixed; in such cases intersection graphs of k-ply neighborhood systems are more realistic.

Intersection graphs of k-ply-neighborhood systems are a strict generalization of (r, s)-civilized graphs, planar graphs and λ-precision unit disk graphs[4][13,20]. On the other hand, it is easy to see that for any fixed k, intersection graphs of k-ply neighborhood systems are a strict subclass of circle intersection graphs. We were motivated to study these classes of graphs for the following reasons:

1. Both (r, s)-civilized graphs and intersection graphs of k-ply-neighborhood systems are powerful enough to model realistic packet radio networks.
2. Problems of interest in the context of channel assignment (such as variants of coloring) can be efficiently solved (either exactly or approximately) when restricted to these graph classes.
3. Finally, intersection graphs of k-ply-neighborhood systems provide a parametric family of increasingly powerful models (with circle intersection graphs being the limiting case) for packet radio networks. Our results yield a family of "slowly degrading" approximation algorithms with increasing value of k, that is, with increasing power of the models.

4. Overall Idea and Basic results

In this section we outline the overall idea behind our approximation algorithms. We also prove some basic results that will be used to develop the approximation results in the subsequent sections of this paper.

The basic idea behind all our coloring heuristics is simple. Given a graph, we suitably partition the vertices into *levels*. For planar and related classes of graphs, this is done by choosing an arbitrary vertex v and constructing a breadth-first spanning (BFS) tree[5] T rooted at v. Each vertex u now can be assigned a unique level number, which is the number of nodes in the unique path in T from the root to u, including the end points. (Figure 1 (a) gives an illustration).

This method of assigning levels to nodes has the property that for any i, no node in level i is adjacent to a node in level $i + 2$ or greater. Therefore, for the MINCOLOR problem, the colors used for nodes in level i can be reused while coloring vertices in levels $i + 2$ or greater. Similarly, for the MIND2COLOR problem, the colors used for nodes in level i can be reused while coloring vertices in levels $i + 3$ or greater.

For a geometric graph G, the levels are obtained by dividing the plane $\mathbb{R}^2$ into horizontal strips of equal width. (This procedure is explained in Section 5.1; an illustration is shown in Figure 1 (b).) Assume that the strips are numbered using successive integers starting with 1. Each node in the ith horizontal strip is assigned the level i. The width of each strip, which depends on the parameters of the geometric graph G, is so chosen that for any i, no node in level i is adjacent to a node in level $i + 2$ or higher.

The following lemma, which relates the treewidths of G and G^2, is used in establishing the performance guarantees of our approximation algorithms.

Lemma 4.1. Let G be a graph with treewidth k. Let the maximum degree of a node in G be Δ. Then the treewidth of G^2 is at most $(k + 1)\Delta - 1$.

Proof: Consider any tree decomposition (X, T) of $G(V, E)$ such that the treewidth of the decomposition is k. By Definition 2.5, any subset X_i in the collection X satisfies the condition $|X_i| \leq k + 1$.

We can construct a tree decomposition (X', T') for G^2 as follows. The tree T' is identical to T. The collection X' is obtained by replacing each set X_i by a set X'_i, where

$$X'_i = X_i \cup \{\, v : \text{for some } x \in X_i\text{: } (x, v) \in E \,\}.$$

In other words, X'_i is obtained by adding to X_i the neighbors of every node in X_i. Using Definition 2.5, it can be verified that (X', T') is indeed a tree decomposition for G^2. Since the maximum node degree in G is Δ, it follows that for any $i \in I$, $|X'_i| \leq (k+1)\Delta$. Thus, the treewidth of G^2 is at most $(k + 1)\Delta - 1$. □

An immediate consequence of Lemma 4.1 and Theorem 2.6 is the following.

Proposition 4.2. The problem MIND2COLOR can be solved in $O(n)$ time for n-node graphs for which the treewidth and maximum node degree are both bounded by constants. □

5. Approximation Results for $(\mathbf{r}, \mathbf{s})$-Civilized Graphs

When considering (r, s)-civilized graphs, we assume that the parameters r and s are *fixed*. Although the performance

---

[4] For any fixed $\lambda > 0$, consider a finite set of unit disks in the plane where the centers of any two disks are at least λ apart. A λ-precision unit disk graph $G(V, E)$ corresponding to the above set of unit disks has vertices of G in one-to-one correspondence with the set of unit disks and two vertices are joined by an edge iff the corresponding disks intersect.

[5] Since we are assuming that the graph is connected, such a tree exists.

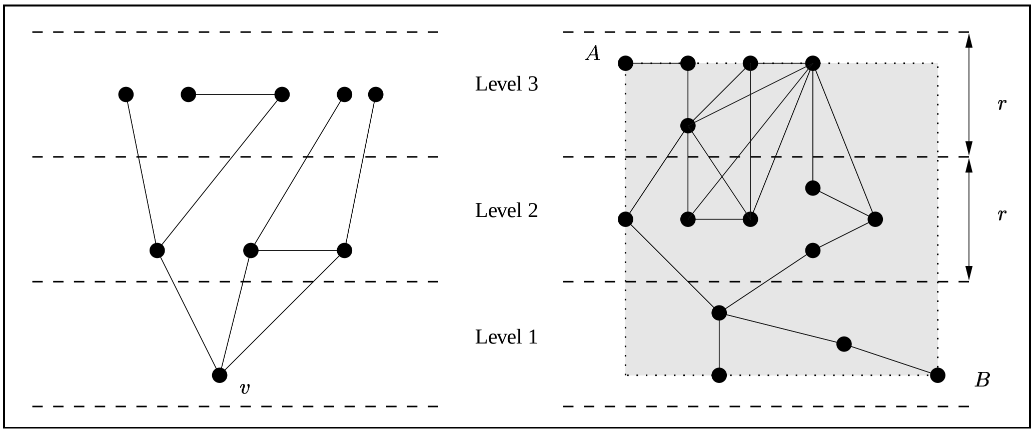

Figure 1. Partition into levels: (a) Using a BFS tree (b) Using horizontal strips of width r

guarantees of our heuristics are independent of the values of r and s, the running times of the heuristics are polynomial only when r and s are fixed.

We discuss our algorithms for *planar* (r, s)-civilized graphs. However, all the algorithms can be extended directly to civilized graphs drawn in higher dimensions (albeit with slightly worse performance guarantees). For the remainder of this section, we use (r, s)-civilized graphs to mean planar (r, s)-civilized graphs.

When r and s are constants, (r, s)-civilized graphs have the following important property which will be used in the approximation algorithm of Section 5.2.

Lemma 5.1. Let r and s be constants and let G be an (r, s)-civilized graph. Then the maximum node degree in G is bounded by a constant $C_{r,s}$ that depends only on r and s.

Proof: Consider a point X that represents a node of G. All the nodes that are neighbors of X in G are located within or on the circle of radius r centered at X; that is, the region containing all the neighbors of X has area at most πr^2. Partition this region into square cells of side $s/\sqrt{2}$. Under this partitioning scheme, each square cell contains at most one neighbor of X, since G is an (r, s)-civilized graph. Thus, the number of neighbors of X is at most $C_{r,s} := \lceil 2\pi r^2/s^2 \rceil$, which is a constant given that r and s are constants. ☐

The MINCOLOR problem for unit disk graphs was shown to be NP-complete in [8]. An examination of this proof shows that the graph resulting from the reduction is a (1,2)-civilized graph. Hence the MINCOLOR problem remains NP-complete for (r, s)-civilized graphs. The MIND2COLOR problem was shown to be NP-complete for a class of graphs called *planar point graphs* in [27]. The reduction used to prove this result actually produces a $(1, 3)$-civilized graph. Therefore, the MIND2COLOR problem is also NP-complete for (r, s)-civilized graphs. These hardness results in conjunction with the usefulness of (r, s)-civilized graphs in modeling packet radio networks motivate the study of approximation algorithms for MINCOLOR and MIND2COLOR problems for (r, s)-civilized graphs. By Lemma 5.1, for constant values of r and s, (r, s)-civilized graphs are degree bounded. Therefore, by Observation 2.10, both MINCOLOR and MIND2COLOR can be approximated to within a constant factor (namely, the maximum node degree) for (r, s)-civilized graphs. The approximation algorithms presented in this section provide a substantially better performance guarantee, namely 2, for both the MINCOLOR and the MIND2COLOR problems. For expository reasons, we first discuss our approximation algorithm for MINCOLOR and then show how a similar idea leads to an approximation algorithm for MIND2COLOR as well.

5.1. *Approximation for* MINCOLOR

Recall that an (r, s)-civilized graph G is specified by a set $\mathcal{P}$ of points in the plane, where each point in $\mathcal{P}$ corresponds to a node of G. So, we will use the terms "point" and "node" interchangeably throughout this section. Let $R_{\mathcal{P}}$ be the axis parallel rectangle whose height is determined by the top-most and bottom-most points of $\mathcal{P}$ and whose width is determined by the left-most and right-most points of $\mathcal{P}$. Note that every point of $\mathcal{P}$ is contained within (or is on) $R_{\mathcal{P}}$. We envision dividing $R_{\mathcal{P}}$ into horizontal strips, each of width r (see Figure 1 (b) for an illustration). Now, for each point $X \in \mathcal{P}$, it is easy to determine the strip to which X belongs. (When a point lies on the boundary between two successive strips, the point is assigned to the bottom strip.) Let the successive strips be numbered from bottom to top using integers 1, 2, . . ., t. For each i, let $S_i \subseteq \mathcal{P}$ denote the set of points assigned to strip i, and let G_{S_i} denote the subgraph of G induced by the points in S_i ($1 \le i \le t$). The following result is proven in [13].

Lemma 5.2. Let f be a fixed positive integer. For any i, $1 \le i \le t - f + 1$, let $\sigma_{i,f}$ denote the set $\bigcup_{j=i}^{i+f-1} S_j$ and let $G_{\sigma_{i,f}}$ denote the subgraph of G induced by the vertex set $\sigma_{i,f}$. The treewidth of $G_{\sigma_{i,f}}$ is $O(fr^2/s^2)$. □

The intuition behind the proof of Lemma 5.2 is the following. When we consider a rectangular slice of height rf and width r, the number of vertices of $G_{\sigma_{i,f}}$ within the slice is $O(fr^2/s^2)$, since $G_{\sigma_{i,f}}$ is an (r, s)-civilized graph. Further, the vertices within the rectangular slice form a separator for $G_{\sigma_{i,f}}$; that is, the removal of these vertices splits $G_{\sigma_{i,f}}$ into disjoint subgraphs. This idea is used recursively in [13] to construct a tree decomposition of treewidth $O(fr^2/s^2)$.

Using the above lemma with $f = 1$ and the fact that r and s are fixed, we can conclude that the treewidth of each subgraph G_{S_i} ($1 \le i \le t$) is bounded by a constant. Therefore, we can use Theorem 2.6 to obtain a minimum vertex coloring of each subgraph G_{S_i}. Furthermore, since G is (r, s)-civilized, for any i, no node in S_i is adjacent to a node in S_{i+2}. So, the colors used for points in an odd (even) numbered strip can be reused in any other odd (even) numbered strip. Let N_{odd} and N_{even} denote the maximum number of colors used by the heuristic to color an odd strip and an even strip respectively. Thus, the heuristic uses at most $N_{\text{odd}} + N_{\text{even}}$ colors. However, any optimal coloring of G must use at least $\max\{N_{\text{odd}}, N_{\text{even}}\}$ colors, since the subgraph within each strip was colored optimally by the approximation algorithm. Therefore, the algorithm provides a performance guarantee of 2.

To estimate the running time of the algorithm, note that for an n-node graph, the number of nonempty strips is at most n. The subgraph induced on any strip can be optimally colored in $O(n)$ time by Theorem 2.6. So, the running time of the algorithm is $O(n^2)$. To summarize:

Theorem 5.3. For any positive constants r and s, there is 2-approximation algorithm with a running time of $O(n^2)$ for the MINCOLOR problem for planar (r, s)-civilized graphs. □

5.2. Approximation for MIND2COLOR

In this section, we first show that a direct extension of the idea used to prove Theorem 5.3 leads to an approximation algorithm with a performance guarantee of 3 for the MIND2COLOR problem. We then present a modification to the algorithm that improves performance guarantee to 2.

We begin with a description of our 3-approximation algorithm for the MIND2COLOR problem on (r, s)-civilized graphs. As in the algorithm for MINCOLOR, we first construct the horizontal strips and thus partition the set $\mathcal{P}$ into sets S_1, S_2, . . ., S_t. To approximate the MINCOLOR problem, we divided the strips into two groups (odd and even numbered strips). To obtain a near-optimal solution to the MIND2COLOR problem, we divide the strips into three groups Γ_0, Γ_1 and Γ_2 as follows: strip i is included in group Γ_j, where $j \equiv (i-1) \pmod 3$. (Thus, for $j = 0, 1, 2$, group Γ_j contains every third strip beginning with strip $j + 1$.)

Consider any group Γ_j. Note that the minimum distance between any pair of points from different strips of Γ_j is greater than $2r$. Since G is an (r, s)-civilized graph, no pair of points from different strips of Γ_j can be adjacent in G^2. In other words, the colors used in any distance-2 coloring of a strip in Γ_j can be reused in obtaining a distance-2 coloring of any other strip in Γ_j.

Consider any group Γ_j and any strip i in Γ_j. As before, let S_i denote the set of vertices of G in strip i. Further, let $G^{(2)}_{S_i}$ denote the subgraph of G^2 induced[6] by the subset S_i. We now show how $G^{(2)}_{S_i}$ can be optimally colored in polynomial time. Let $\sigma_i = S_{i-1} \cup S_i \cup S_{i+1}$ and let G_{σ_i} denote the subgraph of G induced by σ_i. Applying Lemma 5.2 with $f = 3$, it follows that G_{σ_i} is treewidth bounded. Furthermore, G_{σ_i} is also degree bounded, since G itself is degree bounded. Using these two facts and Lemma 4.1, it follows that $G^2_{\sigma_i}$ is treewidth bounded. As observed earlier, a vertex in S_i can be adjacent only to vertices in σ_i. Therefore, $G^{(2)}_{S_i}$ must be a subgraph of $G^2_{\sigma_i}$. Since the latter is treewidth bounded, so is the former. Hence, by Theorem 2.6, we can color $G^{(2)}_{S_i}$ optimally in polynomial time.

Our heuristic uses three different sets of colors for the three groups. Within each group, colors used for one strip are reused while coloring another strip. Let N_j denote the maximum number of colors used by the heuristic to distance-2 color a strip in group Γ_j ($j = 0, 1, 2$). Thus, the heuristic uses at most $N_0 + N_1 + N_2$ colors. However, any optimal distance-2 coloring of G must use at least $\max\{N_0, N_1, N_2\}$ colors, since the subgraph within each strip was distance-2 colored optimally by the heuristic. Therefore, the perfor-

---

[6] It should be noted that the square of the subgraph G_{S_i} is, in general, different from the subgraph of G^2 induced on S_i.

mance guarantee provided by the algorithm is 3. The running time of the algorithm can be seen to be $O(n^2)$ using an argument identical to that preceding Theorem 5.3. Thus, we conclude:

Theorem 5.4. For any positive constants r and s, there is a 3-approximation algorithm with a running time of $O(n^2)$ for the MinD2Color problem on planar (r, s)-civilized graphs. □

In order to improve the performance guarantee to 2, we partition the strips into just two groups Γ_1 and Γ_2 as follows. Strip i is included in Γ_1 if $i \equiv 1 \pmod 4$ or $i \equiv 2 \pmod 4$; otherwise, strip i is included in Γ_2. (Thus, Γ_1 is the set of strips $\{1, 2,\ 5, 6,\ 9, 10, \ldots\}$ while Γ_2 is the set of strips $\{3, 4,\ 7, 8,\ 11, 12, \ldots\}$.)

First, consider the strips in Γ_1. As before, let S_i denote the subset of vertices in strip i. It is easy to see that for any i and j such that $j \geq i + 4$, no vertex in $S_i \cup S_{i+1}$ is adjacent in G^2 to a vertex in $S_j \cup S_{j+1}$. Thus, for any i and j such that $j \geq i + 4$, the colors used in any valid coloring of the subgraph of G^2 induced by the set $S_i \cup S_{i+1}$ can be reused in coloring the subgraph of G^2 induced by the set $S_j \cup S_{j+1}$. A similar condition holds for the strips in Γ_2.

Also note that, for any $i \geq 1$, the subgraph of G^2 induced by the set $S_i \cup S_{i+1}$ is both degree and treewidth bounded. This follows from Lemma 5.2 (with $f = 2$), Lemma 4.1 and the fact that G itself is degree bounded. Hence, by Theorem 2.6, we can color the subgraph of G^2 induced by $S_i \cup S_{i+1}$ optimally in polynomial time.

The modified heuristic uses two different sets of colors for the two groups, namely Γ_1 and Γ_2. Within each group, colors used for one consecutive pair of strips are reused while coloring other consecutive pairs of strips. Let N_j denote the maximum number of colors used by the heuristic to distance-2 color a consecutive pair of strips in group Γ_j ($j = 1, 2$). Thus, the heuristic uses at most $N_1 + N_2$ colors. However, any optimal distance-2 coloring of G must use at least $\max\{N_1, N_2\}$ colors, since the subgraph within each consecutive pair of strips was distance-2 colored optimally by the heuristic. Therefore, the algorithm provides a performance guarantee of 2. As before, the running time of the algorithm is $O(n^2)$. Thus, we have:

Theorem 5.5. For any positive constants r and s, there is a 2-approximation algorithm with a running time of $O(n^2)$ for the MinD2Color problem on planar (r, s)-civilized graphs. □

6. Distance-2 Coloring of Bounded Degree Planar Graphs

The MinD2Color problem remains NP-hard even for bounded degree planar graphs [23]. Reference [25] presents an approximation algorithm with a performance guarantee of at most 9 for the MinD2Color problem for all planar graphs. In this section, we show that for planar graphs of bounded degree, the level partitioning approach mentioned in Section 4 leads to a heuristic with a performance guarantee of 2.

Given a bounded degree planar graph G, we partition the vertices into levels using the BFS approach given in Section 4. As before, let S_i denote the set of vertices in level i. As in the 2-approximation algorithm for (r, s)-civilized graphs, we partition the levels into two groups Γ_1 and Γ_2. Thus, Γ_1 is the set of strips $\{1, 2,\ 5, 6,\ 9, 10, \ldots\}$ while Γ_2 is the set of strips $\{3, 4,\ 7, 8,\ 11, 12, \ldots\}$. Consider the strips in Γ_1 and let i be an integer such that both S_i and S_{i+1} are in Γ_1. (A similar argument can be made for any consecutive pair of strips in Γ_2.) Let $\sigma_i = S_i \cup S_{i+1}$ and let $G^{(2)}_{\sigma_i}$ denote the subgraph of G^2 induced by the vertices in σ_i. We now show how $G^{(2)}_{\sigma_i}$ can be colored optimally in polynomial time.

Let G_{σ_i} denote the subgraph of G induced by σ_i. It is well known [5] that G_{σ_i} belongs to the class of 2-outerplanar graphs. Since the treewidth of a t-outerplanar graph is at most $3t - 1$ [7], it follows that G_{σ_i} is treewidth bounded. Furthermore, G_{σ_i} is also degree bounded, since G itself is degree bounded. Using these two facts and Lemma 4.1, it follows that $G^2_{\sigma_i}$ is treewidth bounded. As observed earlier, a vertex in S_i can be adjacent only to vertices in σ_i. Therefore, $G^{(2)}_{\sigma_i}$ must be a subgraph of $G^2_{\sigma_i}$. Since the latter is treewidth bounded, so is the former. Hence, by Theorem 2.6, we can color $G^{(2)}_{S_i}$ optimally in polynomial time.

Now, using the same approach as in the 2-approximation algorithm for (r, s)-civilized graphs, we can obtain an approximation algorithm with a performance guarantee of 2 for bounded degree planar graphs. This result is summarized in the following theorem.

Theorem 6.1. There is an $O(n^2)$ time approximation algorithm with a performance guarantee of 2 for the MinD2Color problem on bounded degree planar graphs. □

7. Distance-2 Coloring of q-Inductive Graphs

We use the definition of q-inductive graphs as given in in [14].

Definition 7.1. Let q be a positive integer. A class of graphs $\mathcal{G}$ is *q-inductive* if for every $G \in \mathcal{G}$, the vertices of G can be assigned distinct integers in such a way that each vertex is adjacent to at most q higher numbered vertices.

Several well known classes of graphs belong the q-inductive class for appropriate values of q. For example, trees are 1-inductive, outerplanar graphs are 2-inductive (see [22]) and planar graphs are 5-inductive. By a simple application of Euler's formula [32], it can be seen that graphs of bounded genus are $O(1)$-inductive. It can also be verified that any chordal graph with maximum clique size ω is $(\omega - 1)$-inductive (any perfect elimination order is an $(\omega - 1)$-inductive order) and that any graph of treewidth t is t-inductive.

The intersection graphs of *k-ply-neighborhood systems* defined in Section 3 are known to be inductive:

Theorem 7.2 [20]. The intersection graph of a k-ply-neighborhood system in d-dimensions is $3^d k$ inductive. □

Several optimization problems for inductive graphs have been studied in the literature (see for example [15] and the references cited therein). Here, we study the problem of obtaining an approximation for the MinD2Color problem for q-inductive graphs. Our approach is facilitated by a linear relationship between the inductiveness of a graph and its thickness (defined in Section 1.1). This relationship, stated below, is an easy consequence of a result (Lemma 3.1) proved in [1].

Fact 7.3 [1]. If a graph is q-inductive, then its thickness is at most q. □

We now discuss the usefulness of the above relationship in the context of the MinD2Color problem. As mentioned in Section 1.1, Ramanathan and Lloyd [25] presented a heuristic for the MinD2Color problem. They showed that the performance guarantee provided by the heuristic is $O(\theta)$ for all graphs of thickness θ. An outline of this heuristic is as follows. First, the heuristic arranges the vertices of the graph in a certain linear order. Let $\langle v_1, v_2, \ldots, v_n \rangle$ denote the chosen ordering. Then, the heuristic iterates over the vertices in that order assigning a color to vertex v_i during iteration i. The color assigned to vertex v_i is the smallest color not used by any of the previously colored vertices which are within a distance of at most two from v_i. In the remainder of this section, we will refer to this heuristic as FF (first fit).

When FF is applied to a q-inductive graph G, the vertex ordering chosen by FF corresponds to a q-inductive ordering. This observation in conjunction with the performance guarantee of FF and Fact 7.3 implies that for q-inductive graphs, FF provides a performance guarantee of $O(q)$. By a closer examination of the analysis of the performance of FF in [25], we can establish a more precise bound on the performance guarantee provided by the heuristic for q-inductive graphs.

Theorem 7.4. For the MinD2Color problem on q-inductive graphs, Algorithm FF provides a performance guarantee of at most $2q - 1$.

Proof: Following the proof of Theorem 4.1 in [25] for planar graphs, and noting that the proof relies only on the 5-inductiveness of planar graphs, it can be verified that the number of colors used by the heuristic for a q-inductive graph with maximum node degree Δ is at most $q\Delta + (q-1)(\Delta - q) + 1$. This expression is at most $(2q-1)\Delta + 1$, for all $q \geq 1$. Since at least $\Delta + 1$ colors are necessary for any valid D2-coloring (Fact 2.8), the performance guarantee provided by FF on q-inductive graphs is at most $2q - 1$. □

Since bounded genus graphs and graphs of d-dimensional k-ply-neighborhood graphs for fixed values of d and k are $O(1)$-inductive, the following observation is an easy consequence of the above theorem.

Observation 7.5. Algorithm FF provides a performance guarantee of $O(1)$ for the MinD2Color problem for bounded genus graphs and intersection graphs of d-dimensional k-ply-neighborhood systems, where d and k are fixed integers. □

As indicated in [25], the running time of FF on a q-inductive graph with n nodes and maximum degree Δ is $O(nq\Delta)$.

8. Concluding Remarks

In this paper, we proposed the intersection graphs of k-ply-neighborhood systems as a reasonable model for a large class of packet radio networks. The minimum channel assignment problem for such networks was modeled as the distance-2 coloring problem on the underlying undirected graph. Since these problems are NP-hard, we proposed efficient heuristic algorithms with provably good performance guarantees. Our approximation algorithms for (r, s)-civilized graphs and bounded degree planar graphs are based on the same approach; this approach partitions the vertex set of the graph into several levels and the distance-2 coloring is constructed by considering the subgraphs induced by the nodes in a small number of consecutive levels.

We conclude by pointing out some directions for further research. First, it would be useful to examine whether our approach can be used to obtain good approximation algorithms for other versions of the channel assignment problem discussed in [24]. Another direction is to devise other approaches that provide better performance guarantee results for the MIND2COLOR problem. Finally, it is of interest to identify other suitable classes of graphs which can also serve as models for packet radio networks that arise in practice.

Some Recent Results

While this paper was under review, several new results for the MIND2COLOR problem appeared in the literature. Heuvel and McGuinness [12] have shown that for any planar graph with maximum node degree Δ, G^2 can be colored using at most $2\Delta + 25$ colors. Agnarsson and Halldórsson [1] considered the minimum vertex coloring problem for G^k, the k^{th} power of a graph G. They have shown that when G is planar, G^k is $\Theta(\Delta^{\lfloor k/2 \rfloor})$-inductive, where Δ is the maximum node degree of G. They also present a 2-approximation algorithm for the MIND2COLOR problem for planar graphs. Further, they establish that for a general graph G with n nodes, it is computationally difficult to approximate the minimum coloring problem for G^k to within a factor $\Omega(n^{1/2-\epsilon})$ for any $\epsilon > 0$ and $k \geq 2$. Very recently, Zhou, Kanari and Nishizeki [33] have shown that for any graph G of bounded treewidth and any fixed integer $k \geq 1$, the MINCOLOR problem for G^k can be solved in polynomial time. This result in conjunction with our level decomposition approach presented in Section 6 also yields a 2-approximation algorithm for the MIND2COLOR problem for all planar graphs.

Acknowledgements

We thank the referees for a careful reading of the manuscript and helpful suggestions that improved the paper. We thank Professor Magnús M. Halldórsson (University of Iceland and University of Bergen) for his observations which helped us to improve the performance of our approximation algorithms for the (r, s)-civilized and bounded degree planar graphs. We also thank Professor Harry Hunt III (University at Albany - State University of New York), Professor Errol Lloyd (University of Delaware), Professor Arunabha Sen (Arizona State University), Dr. Ewa Malesinska (Technical University of Berlin) and Dr. Ravi Sundaram (Akamai Technologies, Inc.) for several fruitful discussions in the course of writing this paper.

References

[1] G. Agnarsson and M. Halld´orsson, Coloring Powers of Planar Graphs, in *Proc. Eleventh Annual ACM-SIAM Symposium on Algorithms* (SODA'2000), San Francisco, CA, Jan. 2000, pp. 654–662.

[2] S. Arnborg, J. Lagergren and D. Seese, Easy Problems for Tree-Decomposable Graphs, *J. Algorithms*, 12 (1991), pp. 308–340.

[3] S. Arnborg and A. Proskurowski, Linear-Time Algorithms for NP-hard problems on Graphs Embedded in k-Trees, *Discrete Applied Math.*, 23 (1989), pp. 11–24.

[4] S. Arnborg, Efficient Algorithms for Combinatorial Problems on Graphs with Bounded Decomposability, *BIT*, 25 (1985), pp. 2–23.

[5] B. S. Baker, Approximation Algorithms for NP-complete Problems on Planar Graphs, *J. ACM*, 41 (1994), pp. 308–340.

[6] H. L. Bodlaender, Dynamic programming on graphs of bounded treewidth, in *Proc. 15th International Colloquium on Automata Languages and Programming* (ICALP'88), LNCS Vol. 317, 1988, pp. 105–118.

[7] H. L. Bodlaender, Planar Graphs with Bounded Treewidth, Technical Report RUU-CS-88-14, Department of Computer Science, University of Utrecht, The Netherlands, March 1988.

[8] B. N. Clark, C. J. Colbourn, and D. S. Johnson, Unit Disk Graphs, *Discrete Math.*, 86 (1990), pp. 165–177.

[9] P. G. Doyle and J. L. Snell, *Random Walks and Electric Networks*, The Carus Mathematical Monographs, The Mathematical Association of America, Providence, RI, 1984.

[10] K. Greenbaum, G. L. Miller and S. H. Teng, Moments of Inertia and Graph Separators, in *Proc. 5th Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA'94), Arlington, VA, Jan. 1994, pp. 452–461.

[11] A. Gräf, M. Stumpf and G. Weißenfels, On Coloring Unit Disk Graphs, *Algorithmica*, 20 (1998), pp. 277–293.

[12] J. van den Huevel and S. McGuinness, Coloring the Square of a Planar Graph, Report No. LSE-CDAM-99-06, Center for Discrete and Applicable Mathematics, London School of Economics, London, UK, June 1999.

[13] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz and R. E. Stearns, NC-Approximation Schemes for NP-

and PSPACE-Hard Problems for Geometric Graphs, *J. Algorithms*, 26 (1998), pp. 238–274.

[14] S. Irani, Coloring Inductive Graphs On-line, *Algorithmica*, 11 (1994), pp. 53–72.

[15] K. Jansen, An Approximation Scheme for Bin Packing with Conflicts, in *Proc. Scandinavian Workshop on Algorithmic Theory* (SWAT'98), Stockholm, Sweden, LNCS Vol. 1432, Springer-Verlag, July 1998, pp. 35–46.

[16] D. S. Johnson, The NP-Completeness Column: An Ongoing Guide, *J. Algorithms*, 3 (1982), page 184.

[17] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi and D. J. Rosenkrantz, Simple Heuristics for Unit Disk Graphs, *Networks*, 25 (1995), pp. 59–68.

[18] S. T. McCormick, Optimal Approximation of Sparse Hessians and its Equivalence to a Graph Coloring Problem, *Math. Programming*, 26 (1983), pp. 153–171.

[19] X. Ma and E. L. Lloyd, Personal communication, March 1999.

[20] G. L. Miller, S. H. Teng, W. Thurston and S. A. Vavasis, Separators for Sphere Packings and Nearest Neighbor Graphs, *J. ACM*, 44 (1997), pp. 1–29.

[21] G. L. Miller, S. H. Teng, and S. A. Vavasis, A Unified Geometric Approach to Graph Separators, in *Proc. 32nd IEEE Symposium on Foundations of Computer Science* (FOCS'91), Puerto Rico, Oct. 1991, pp. 538–547.

[22] A. Proskurowski and M. M. Syslo, Efficient Vertex and Edge Coloring of Outerplanar Graphs, *SIAM J. Algebraic and Discrete Methods*, 7 (1986), pp. 131–136.

[23] S. Ramanathan, Scheduling Algorithms for Multi-Hop Radio Networks, Ph.D. thesis, Department of Computer Science, University of Delaware, Newark, 1993.

[24] S. Ramanathan, A Unified Framework and Algorithm for (T/F/C) DMA Channel Assignment in Wireless Networks, in *Proc. IEEE INFOCOM'97*, Kobe, Japan, April 1997.

[25] S. Ramanathan and E. L. Lloyd, Scheduling Algorithms for Multi-Hop Radio Networks, *IEEE/ACM Trans. on Networking*, 1 (1993), pp. 166–172.

[26] N. Robertson and P. Seymour, Graph Minors XIV, Wagner's conjecture, Preprint, 1992.

[27] A. Sen and M. L. Huson, A New Model for Scheduling Packet Radio Networks, *Wireless Networks*, 3 (1997), pp. 71–82.

[28] A. Sen and E. Malesinska, On Approximation Algorithms for Radio Network Scheduling, Tech. Report 96-008, Department of Computer Science and Engineering, Arizona State University, Tempe, AZ, 1996.

[29] D. Spielman and S. H. Teng, Spectral Partitioning Works: Planar Graphs and Finite Element Meshes, in *37th Annual Symposium on Foundations of Computer Science* (FOCS'96), Burlington, VT, Oct. 1996, pp. 96–105.

[30] S. H. Teng, Points, Spheres, and Separators, A Unified Geometric Approach to Graph Separators, Ph.D. thesis, School of Computer Science, Carnegie Mellon University, CMU-CS-91-184, Pittsburgh, August 1991.

[31] S. A. Vavasis, Automatic Domain Partitioning in Three Dimensions, *SIAM J. Sci. Stat. Comp.*, 12 (1991), pp. 950–970.

[32] R. J. Wilson, *Introduction to Graph Theory*, Second Edition, Longman Publishers, Bungay, UK, 1979.

[33] X. Zhou, Y. Kanari and T. Nishizeki, Generalized Vertex Colorings of Partial k-Trees, *IEICE Transaction Fundamentals*, Vol. E-A, No. 4, Apr. 2000, pp. 1–8.