Algorithmica (1998) 20: 277–293

Algorithmica

On Coloring Unit Disk Graphs[1]

A. Gräf,[2] M. Stumpf,[2] and G. Weißenfels[2]

Abstract. In this paper the coloring problem for unit disk (UD) graphs is considered. UD graphs are the intersection graphs of equal-sized disks in the plane. Colorings of UD graphs arise in the study of channel assignment problems in broadcast networks. Improving on a result of Clark et al. [2] it is shown that the coloring problem for UD graphs remains NP-complete for *any* fixed number of colors $k \geq 3$. Furthermore, a new 3-approximation algorithm for the problem is presented which is based on network flow and matching techniques.

Key Words. Channel assignment problem, Graph coloring, Unit disk graphs, Approximation algorithm.

1. Introduction. *Unit disk* (*UD*) *graphs* are the intersection graphs of equal-sized disks in the plane [2]. They can also be described in terms of "distance" or "proximity" models, which consist of a value $d \geq 0$ and an embedding of the vertices in the plane such that vw is an edge iff $d(v, w) \leq d$, where $d(v, w)$ denotes the Euclidean distance of v and w in the specified embedding. UD graphs arise in a variety of different problems related to broadcast networks, see [2]. In particular, *colorings* of UD graphs play an important role in the *channel assignment problem* [10]. In this context the vertices of the graph G represent transmitters of the same power in a broadcast network, and two transmitters may interfere if they have a distance of at most d, for some given $d \geq 0$. In the simplest setting, interfering transmitters should be given different channels. Since the spectrum available to broadcast services is a limited resource, we would also like to keep the number of channels used in a valid channel assignment of a given network as small as possible. Obviously, this task can be formulated as a graph coloring problem on the underlying UD graph.

It is well known that the general graph coloring problem is NP-complete and that even the problem of *approximating* the chromatic number within any constant ratio is NP-hard [12]. Clark et al. proved in [2] that the coloring problem remains NP-complete for UD graphs. However, their proof left open the possibility that the problem might become easier when a fixed number of colors $k > 3$ is considered. Employing a generalization and combination of techniques in [2] and [5], in this paper we improve on this result by showing that the UD graph coloring problem remains NP-complete for *any* fixed number of colors $k \geq 3$. As with the result of [2], we can show that the problem remains NP-complete when the graph is given *with its model*. This is an important point since

---

[1] This research was supported by the Ministry of Commerce, Transport, Agriculture, and Viticulture of Rhineland-Palatinate under the Programme of the Association for Research in Media Technology in South-West Germany (FMS).

[2] Johannes Gutenberg-Universität, Musikwissenschaftliches Institut, Abteilung Musikinformatik, P.O. Box 3980, 55099 Mainz, Germany. {ag,martin,wf}@muwiinfa.geschichte.uni-mainz.de.

Received February 12, 1996; revised October 9, 1996. Communicated by K. Mehlhorn.

the UD graph *recognition* problem (given a graph, decide whether it is a UD graph and construct a corresponding distance model) is NP-hard [1].

We also consider the problem of *approximating* the chromatic number of UD graphs. It is easy to see that the vertex degrees in a UD graph G containing at least one edge are always bounded from above by $6\omega(G) - 7$, where $\omega(G)$ denotes the maximum clique size of G. Hence G can be colored using at most $6\omega(G) - 6$ colors using any variation of the "sequential" coloring algorithm (consider the vertices in a given order and always assign the "least" color which is allowed at a given vertex). In fact, Peeters has observed that the sequential coloring algorithm, when applied to a certain "lexicographic" vertex ordering, colors each UD graph G with at most $3\omega(G) - 2$ colors [19].

In this paper we present a new approach to the UD graph coloring problem which is based on the idea of partitioning a UD graph into a collection of special subgraphs called "stripes." This approach makes extensive use of the geometric structure of the UD graph to be colored, and hence requires the graph to be given with its model. Given an appropriate choice of parameters, the stripes turn out to be cocomparability graphs which can be colored optimally using Möhring's algorithm [18]. The stripe colorings are then combined using matching techniques as described in [7]. We prove that this algorithm achieves a worst-case performance ratio of 3. As our test results indicate, our algorithm is an alternative worth considering, depending on the structure of the problem instances for the application at hand.

The paper is organized as follows. In Section 2 we introduce the basic concepts used in this paper. In Section 3 we prove that the UD graph coloring problem is NP-complete for all fixed numbers of colors $k \geq 3$. In Section 4 we discuss our approximation algorithm for UD graph coloring. Section 5 summarizes results and points out open problems. For proofs, test results, and other technical details omitted here the reader is referred to [8] and [9].

2. Preliminaries. All graphs in this paper are finite and undirected and do not have loops or multiple edges, except if explicitly indicated. The sets of vertices and edges of a graph G are denoted $V(G)$ and $E(G)$, respectively. The subgraph *induced* by $V \subseteq V(G)$ is denoted G_V, and we write $\overline{G}$ for the *complement* of G (where $vw \in E(\overline{G}) \Leftrightarrow vw \notin E(G)$ for all $v, w \in V(G) = V(\overline{G})$, $v \neq w$). A graph G is called *complete* iff $vw \in E(G)$ for all $v, w \in V(G)$, $v \neq w$. A subset V of $V(G)$ is called a *clique* of G iff G_V is complete and an *independent set* iff $vw \notin E(G)$, $\forall v, w \in V(G)$. A *partition* of G is a set $\mathcal{V}$ consisting of mutually disjoint subsets of $V(G)$ such that $V(G) = \bigcup_{V \in \mathcal{V}} V$. A *k-coloring* is a mapping $f\colon V(G) \to I$ with $|I| \leq k$ such that $f(v) \neq f(w)$, $\forall vw \in E(G)$. (We also interpret a k-coloring as a partition of G into at most k independent sets.) The *chromatic number* of G, $\chi(G)$, is the minimum k for which G has a k-coloring, and the $\chi(G)$-colorings of G are called *optimal*. The maximum size of a clique of G (the *clique number*) and the maximum vertex degree of G are denoted $\omega(G)$ and $\Delta(G)$, respectively.

Clearly, $\omega(G) \leq \chi(G)$ for each graph G. A graph G is called *perfect* iff $\chi(G_V) = \omega(G_V)$ for each $V \subseteq V(G)$. It is well known (see, e.g., [6]) that the class of perfect graphs is closed under taking complements. An important class of perfect graphs we consider in this paper are the *cocomparability graphs* which are the complements of *comparability* (or *transitively orientable*) *graphs*. We call a graph G a comparability graph if we can orient the edges of G in such a manner that the set R of oriented edges is a transitive relation on $V(G)$. We then say that R is a *transitive orientation* of G.

We now discuss the "main character" of this paper. Given two points $v_1 = (x_1, y_1)$ and $v_2 = (x_2, y_2)$ in the plane, let $d(v_1, v_2)$ denote the *Euclidean distance* between v_1 and v_2. For a set V of points in the plane and a *threshold* value $d \geq 0$ let $G(V, d)$ be the graph with vertex set V and edges $vw \in E(G(V, d)) \Leftrightarrow v \neq w \wedge d(v, w) \leq d$. A graph G is called a *unit disk* (*UD*) *graph* iff $G \cong G(V, d)$ for some suitable set of points V and threshold value d; we then call (V, d) a *distance model* of G. Equivalently, we may think of G as the intersection graph of pairwise distinct, closed equal-sized disks D_v, $v \in V(G)$, where $vw \in E(G)$ iff $v \neq w$ and $D_v \cap D_w \neq \emptyset$. In this case, $\{D_v | v \in V(G)\}$ is called an *intersection model* of G.

In contrast to unit *interval* graphs, which are the counterpart of UD graphs in one-dimensional space (see, e.g., [6]), UD graphs are *not* necessarily perfect. For instance, C_5, the chordless cycle with five vertices, can easily be seen to be a UD graph, but it is not perfect as $\chi(C_5) = 3 > 2 = \omega(C_5)$. It is also worth noting that while the coloring problem remains NP-complete on UD graphs, Clark et al. give an $O(n^{4.5})$ algorithm for computing the clique number of a UD graph [2].

Another fact which deserves mentioning is that it is always possible to "adjust" the threshold value d and the points v in a distance model of a UD graph within certain bounds. More precisely, for each model (V, d) there is an $\varepsilon > 0$ such that $G(V, d') \cong G(V, d)$, $\forall d' \in [d, d + \varepsilon]$ and $G(V \backslash \{v\} \cup \{v'\}, d) \cong G(V, d)$, $\forall v \in V, v' \notin V$: $d(v, v') \leq \varepsilon$. This implies, in particular, that each UD graph has a distance model (V, d) in which d and the coordinates of the points $v \in V$ are all integers. Furthermore, we can replace each vertex in a UD graph by a clique of arbitrary size, an operation which is frequently used in Section 3.

3. The UD *k*-Colorability Problem.

As Clark et al. have shown in [2] the k-colorability problem for UD graphs is NP-complete, even for fixed $k = 3$. The reduction employed in [2] is from planar graph 3-colorability, and so it remains an open question whether the problem is still NP-complete when restricted to some fixed $k > 3$ (note that the k-colorability of planar graphs is NP-complete for $k = 3$, but not for $k > 3$ since planar graphs are all 4-colorable). In this section we use a different reduction to generalize the cited result as follows:

THEOREM 3.1. *The UD k-colorability problem is NP-complete for any fixed $k \geq 3$.*

Theorem 3.1 also holds when the UD graphs are given *with their models*. This is an important observation since the UD graph recognition problem is NP-hard [1].

Our reduction is from the general k-colorability problem. Given any graph G, we will show how to construct a corresponding UD graph $\widehat{G} = (\widehat{V}, \widehat{E})$ which is k-colorable if and only if G is. Our construction is somewhat similar to [2] in that we use an embedding of G into the plane which allows us to replace the edges of G with suitable UD graph chains in a simple and systematic fashion, while preserving the k-colorability property. However, two additional problems arise. First, since G is not necessarily planar we

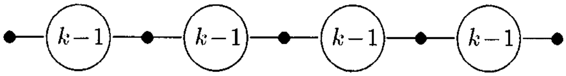

Fig. 1. The k-wire W_k^4.

have to deal with crossing edges in our embedding. Secondly, we have to cope with large vertex degrees in relation to the clique number since any UD graph H satisfies the relation $\Delta(H) \leq 6\omega(H) - 7$ (see Section 4.1). Of course this condition may be violated in a general graph. Both problems are solved with two special types of auxiliary UD graphs which we replace for edge crossings and high degree vertices, respectively.

All "auxiliary" graphs we introduce have some distinguished vertices which are used to connect the graphs to each other; in what follows we refer to these vertices as the *output vertices* of the graph. Readers may convince themselves that all auxiliary graphs are UD graphs. Most of the additional properties stated below follow immediately from the construction. As already noted in Section 2, our construction makes frequent use of cliques joined to single vertices. The cliques are represented by circles labeled with the size of the clique. An edge between a vertex v and a clique C means that v is adjacent to all members of C.

First the UD graphs needed to replace the edges in an embedding of G are introduced. A k-wire of length l, denoted by W_k^l, is shown in Figure 1 (with $l = 4$). The two vertices at both ends are the output vertices of the k-wire. Obviously a k-wire is k-colorable and each k-coloring assigns the same color to both output vertices.

A k-chain of length l, denoted by K_k^l, is shown in Figure 2 (with $l = 3$). Again the output vertices are the two vertices at both ends. Obviously each k-coloring assigns different (but freely choosable) colors to both output vertices.

Now the graphs that will replace the high degree vertices of G are introduced. A k-clone of size $l \geq 2$, denoted by C_k^l, is shown in Figure 3 (with $l = 3$). The output vertices are the vertices $o_0 \cdots o_{l-1}$. The main feature of this construction is that in each k-coloring the same color is assigned to all output vertices.

Finally the graphs needed for replacing the edge crossings in an embedding of G are introduced. This construction is a generalization of a graph employed by Fisher, which is used in [5] to show the NP-completeness of the 3-colorability problem for planar graphs. Such a k-crossing, denoted by H_k, $k \geq 3$, is shown in Figure 4. The vertices $v_0, \ldots, v_3$ are the output vertices of the crossing. A k-crossing is k-colorable and each k-coloring f satisfies $f(v_0) = f(v_2)$ and $f(v_1) = f(v_3)$. Furthermore, there exist colorings f_1 and f_2 which satisfy $f_1(v_0) = f_1(v_2) = f_1(v_1) = f_1(v_3)$ and $f_2(v_0) = f_2(v_2) \neq f_2(v_1) = f_2(v_3)$.

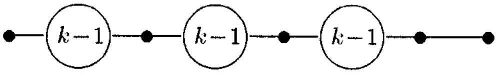

Fig. 2. The k-chain K_k^3.

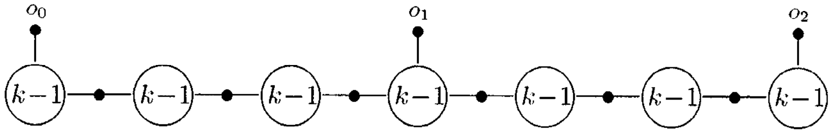

Fig. 3. The k-clone C_k^3.

Given any graph G we now outline how to compute a distance model of a UD graph $\widehat{G}$ which is k-colorable if and only if G is k-colorable. First we show how to embed the given graph in a suitable way. After that $\widehat{G}$ is constructed in several steps in which the vertices, edges, and edge crossings of the original graph are replaced by the auxiliary graphs introduced above.

To solve the problem caused by high degree vertices each such vertex will be replaced by a k-clone of size n, where n is the degree of this vertex. Each neighbor of such a vertex is connected to one output vertex of the k-clone. This obviously preserves the k-colorability property. For technical reasons each vertex having degree $n \geq 2$ is replaced. To prepare these replacements, we first construct G' by replacing each vertex v in G with an independent set $M(v)$ of n vertices, where n is the degree of v. Each vertex of this set is connected with one neighbor of v. Obviously each vertex in G' has degree one.

Now G' is embedded into the plane. For this purpose the vertices are placed on the x-axis at equidistant positions (where the vertices in each set $M(v)$ are placed consecutively). Each edge is embedded by one horizontal and two vertical line segments, where the horizontal line segments have pairwise distinct y-coordinates. An example for this construction is given in Figure 5.

We can finally construct an embedding of $\widehat{G}$ by some simple replacements. First the vertices contained in those sets $M(v)$ with $|M(v)| = n \geq 2$ are replaced by the output

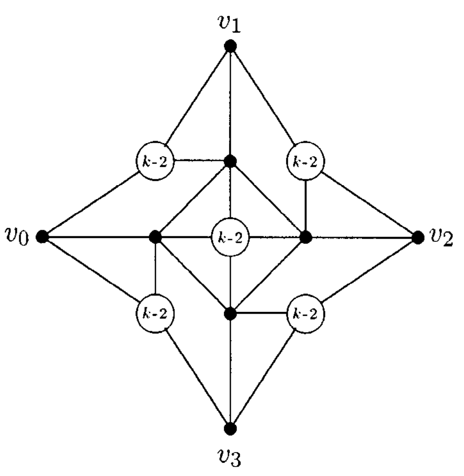

Fig. 4. The k-crossing H_k.

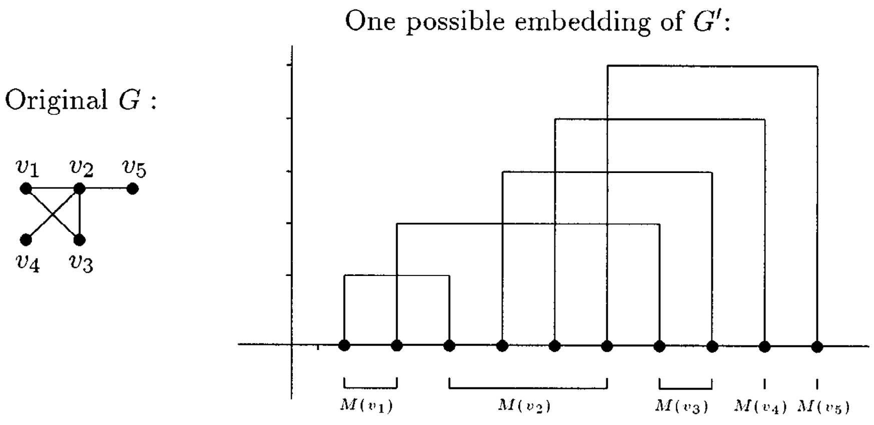

Fig. 5. Example for the described embedding.

vertices of a k-clone of size n where the remaining vertices of the k-clone are placed below the x-axis. Secondly the edge crossings in the embedding of G' are replaced by k-crossings. Finally the line segments which may have been subdivided by k-crossings in the preceding step are replaced by k-chains and k-wires. If an edge is not crossed we simply replace it by a k-chain of suitable length. If an edge uv is crossed by others, then its line segments are subdivided into the following parts: the part that is connected to u, the parts which connect the k-crossings (if several are present), and the part that is connected to v. The latter is replaced by a k-chain of suitable length. The remaining parts are replaced by k-wires of suitable length. This is illustrated in Figures 6 and 7, where a clique is represented by a circle without a dot inside.

The resulting graph $\widehat{G}$ is obviously a UD graph consisting of the four auxiliary graphs. We are now ready to prove the main theorem of this section which says that the UD k-colorability problem is NP-complete for any fixed $k \geq 3$.

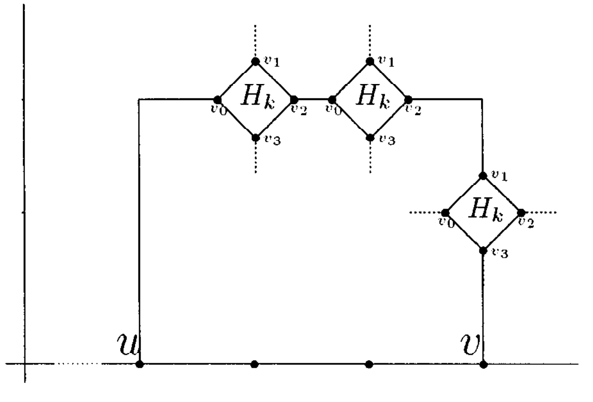

Fig. 6. Before the replacements.

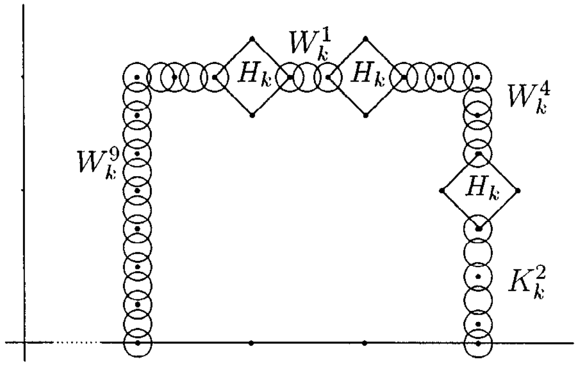

Fig. 7. The described replacements.

PROOF OF THEOREM 3.1. Let $G = (V, E)$ be any graph and $k \geq 3$. The construction of the corresponding UD graph $\widehat{G} = (\widehat{V}, \widehat{E})$ and its embedding can obviously be done in polynomial time. We have to show the following: G is k-colorable $\Leftrightarrow$ $\widehat{G}$ is k-colorable.

Each vertex $v \in V$ is replaced by a set of vertices $M(v)$ in the first step. These vertices are replaced by the output vertices of a k-clone if $|M(v)| \geq 2$. The set of these output vertices is denoted $I(v)$.

($\Rightarrow$) Let f: $V \rightarrow \{1, \ldots, k\}$ be a k-coloring of G. We are going to construct a k-coloring g: $\widehat{V} \rightarrow \{1, \ldots, k\}$ of $\widehat{G}$. First we define, $\forall v \in V$,

$$g(x) := f(v), \qquad \forall x \in I(v),$$

where the k-clones are k-colorable under this condition.

The remaining vertices of $\widehat{G}$ are the vertices of the k-chains, k-wires, and k-crossings which connect the output vertices of different k-clones. Let $x \in I(u)$ be any output vertex of any k-clone and let $y \in I(v)$ be the one which is connected to x. If x and y are connected by a single k-chain, then $uv \in E$ is guaranteed by construction and therefore $f(u) \neq f(v)$ and hence $g(x) \neq g(y)$ (remember that a k-chain is k-colorable under this condition). If x and y are connected by one or more k-wires, one or more k-crossings, and a single k-chain we analogously obtain $g(x) \neq g(y)$. Now let $x = a_1, a_2, a_3, \ldots, a_{2n-1}, a_{2n} = y$ be the output vertices belonging to this connection, numbered from "left to right" (Figure 8 shows an example). If we define

$$g(x) = g(a_1) =: g(a_2) =: \cdots =: g(a_{2n-1}) \neq g(a_{2n}) = g(y),$$

the auxiliary graphs are k-colorable under these conditions. In this way we obtain a k-coloring of $\widehat{G}$.

($\Leftarrow$) This can be shown in a similar manner by considering the properties of the auxiliary graphs. $\square$

4. An Approximation Algorithm.

In the previous section we have shown the apparent intractability of the UD graph coloring problem. Being confronted with such a

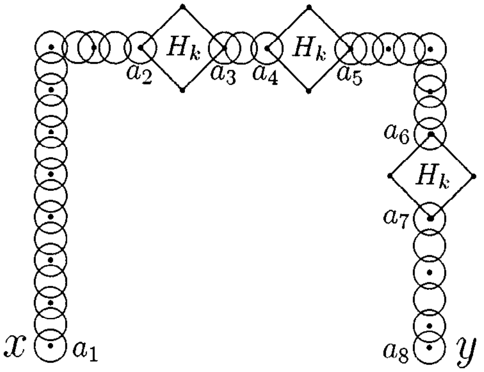

Fig. 8. The output vertices of a crossed edge.

discouraging result, one usually starts the search for special cases of the problem which can be solved efficiently, and for heuristics which produce suboptimal but acceptable results at least "most of the time." Following this line of research, in this section we consider the problem of *approximating* the chromatic number of a UD graph.

Unlike other graph coloring heuristics which are usually quite simple (but nevertheless often difficult to analyze), our algorithm involves a number of different concepts and algorithmic techniques which we develop in a bottom-up fashion. In Section 4.1 we first review the sequential coloring algorithm and discuss how it is applied to the UD graph case. In Section 4.2 we discuss a technique based on bipartite matching which allows us to combine an arbitrary number of subgraph colorings. In Section 4.3 we review Möhring's algorithm for coloring cocomparability graphs, which is the central subroutine in our algorithm. Finally, in Section 4.4 we describe our STRIPE algorithm for coloring UD graphs.

4.1. *The Sequential Coloring Algorithm.* As its name indicates, the sequential coloring algorithm considers the vertices of the graph G to be colored in some linear order, where (starting with 1) the least possible color is assigned to each vertex. Without loss of generality, we assume that the vertex ordering is fixed before the algorithm starts (we can always determine the vertex ordering in a separate pass before the coloring procedure). It is easy to see that the algorithm will actually compute an optimal coloring if it is started with a suitable vertex ordering. (For example, given an optimal coloring $f\colon V(G) \to \{1, \ldots, \chi(G)\}$, order the vertices by their colors.) Another elementary fact is that the sequential coloring algorithm will color each graph G with at most $\Delta(G) + 1$ colors, regardless of the chosen vertex ordering. A better bound for the performance of this algorithm can be given in terms of a parameter introduced by Matula [17], which we call the *span* of an ordering in the following. For a given ordering $<$ of the vertices of G, the span of $<$ is defined as $\max_{v \in V(G)} |\{vw \in E(G) | w < v\}|$. We also call an ordering of span $\leq k$ a *k-span ordering* and we define $\mathrm{sp}(G)$, the span of G, as the minimum span of an ordering of the vertices of G. A minimum span ordering can be determined in time $O(|V(G)| + |E(G)|)$ using a greedy algorithm, see [16]. (More precisely, the algorithm in [16] determines a so-called *smallest-last* ordering which is given by a sequence of vertices $v_1, \ldots, v_n$ such that v_i is a vertex of smallest degree in $G_{\{v_1,\ldots,v_i\}}$.) It is easy to see that the sequential coloring algorithm, when applied to a k-span ordering, colors G using at most $k+1$ colors, and hence $\chi(G) \leq \mathrm{sp}(G)+1$ for each graph G.

For the case of UD graphs, it has been shown by Peeters [19] that each *lexicographic* ordering of the vertices of a UD graph (i.e., order the vertices first by their x- and then by their y-coordinates) achieves a span of at most $3\omega(G)-3$. Thus, each UD graph G can be colored with at most $3\omega(G)-2$ colors in linear time (with respect to the number of vertices and edges). We remark that a straightforward extension of the Peeters construction can be used to prove that for each UD graph G with $\omega(G) \geq 2$ we have that $\Delta(G) \leq 6\omega(G)-7$, see [9] for details. A similar but slightly weaker result has also been obtained in [14].

The original algorithm given in [19] has the disadvantage that it requires lexicographic ordering and thus the graph must be given with its UD model. It is worth noting that the same performance bound can be achieved if we simply apply the sequential coloring algorithm to a minimum span ordering instead. To our knowledge, this is the only known 3-approximation algorithm for the UD graph coloring problem which does not require the graph to be given with its model.

4.2. *Permutative Colorings.* In this section we briefly sketch a technique for combining a given collection of subgraph colorings to a global coloring of a graph (see [7] for further details). The basic idea is to *permute* the subgraph colorings to make them fit together. This "permutative coloring" technique allows us to solve the coloring problem for a special class of cocomparability graphs, namely, the complements of bipartite graphs. In this sense it is a specialization of Möhring's algorithm, to be discussed in Section 4.3. However, the technique is also useful as a general heuristic for combining subgraph colorings, and as such it will be applied in the following.

Let G be a graph and let f and g be colorings of G. We say that g is a *permutation* of f if there is a bijection π from the range of f to the range of g such that $g = \pi \circ f$. Now consider a partition $\{V, W\}$ of G and let g and h be colorings of G_V and G_W, respectively. By I and J we denote the ranges of g and h, respectively. A coloring f of G is called a *permutative coloring* with respect to g and h iff f_V is a permutation of g and f_W is a permutation of h. (Here and in the following, f_V denotes the restriction of f to $V \subseteq V(G)$.) Such a coloring is called *optimal* if it uses as few colors as possible (with respect to the given colorings g and h). It is easy to see that an optimal permutative coloring f with $f_W = h$ can be obtained by *matching* as many colors i in I against corresponding colors j in J as possible, in such a manner that each pair of matched colors (i, j) is contained in the set M of all pairs (i, j) such that $vw \notin E(G)$ for all vertices $v \in V, w \in W$ with $g(v) = i$ and $h(w) = j$. The maximum matching determines a bijective mapping π' between a subset I' of I and a corresponding subset J' of J. We then extend π' to a bijective mapping π by assigning a new color $\notin J$ to each unmatched color $i \in I \backslash I'$. Combining the resulting coloring $\pi \circ g$ with h then yields a coloring of G using $|I| + |J| - \nu(M)$ colors, where $\nu(M)$ denotes the size of a maximum matching contained in M, and this is optimal. This procedure can obviously be carried out in time $O((|I| + |J|)^{2.5} + |V(G)| + |E(G)|)$, using Hopcroft and Karp's well-known bipartite matching algorithm [11].

The notion of permutative colorings can be generalized to partitions $\mathcal{V}$ of arbitrary sizes in the obvious way. However, determining an optimal permutative coloring for *three* subgraph colorings instead of two is already an NP-hard problem (see [7]). Therefore we have to settle for heuristic approaches when combining more than two subgraph colorings. One such heuristic, the *sequential permutative coloring* (SPC) algorithm, is introduced below.

ALGORITHM 4.1 (SPC Algorithm)

Input: Graph G, partition $\mathcal{V}$ of G, and colorings g_V of G_V, $V \in \mathcal{V}$.
Output: A coloring f of G.

Method:
1. We consider the members of $\mathcal{V}$ in some given order $V_1, \ldots, V_r$. Let $f_1 = g_{V_1}$.
2. For each i, $2 \leq i \leq r$, determine an optimal permutative coloring f_i with respect to f_{i-1} and g_{V_i}.
3. Return $f = f_r$.

A simple bound for the worst-case performance of Algorithm 4.1 can be given in terms of the span of the so-called *adjacency graph* of the chosen ordering of the partition $\mathcal{V}$. The adjacency graph is defined as follows:

DEFINITION 4.1. Let G be a graph and let $\mathcal{V}$ be a partition of G. Then the *adjacency graph* $G^{\mathcal{V}}$ is defined by $V(G^{\mathcal{V}}) = \mathcal{V}$ and

$$E(G^{\mathcal{V}}) = \{VW | V, W \in \mathcal{V}, V \neq W \text{ and } vw \in E(G) \text{ for some } v \in V, w \in W\}.$$

PROPOSITION 4.1. *Let G be a graph, let $\mathcal{V}$ be a partition of G, and let g_V be a coloring of G_V for each $V \in \mathcal{V}$. Furthermore, let N be the maximum number of colors in the range of g_V over all $V \in \mathcal{V}$ (i.e., $N = \max_{V \in \mathcal{V}} |\{g_V(v) | v \in V\}|$). Then Algorithm* 4.1, *when applied to a K-span ordering of $G^{\mathcal{V}}$, will color G using at most $(K+1)N$ colors.*

The running time of the SPC algorithm depends on the span of the chosen ordering and the number of colors used in each element of the partition. We cite the following result (see [7]).

PROPOSITION 4.2. *Let G be a graph, let $\mathcal{V}$ be a partition of G, and let g_V be a coloring of G_V for each $V \in \mathcal{V}$. Let $n = |V(G)|$, $m = |E(G)|$, $r = |\mathcal{V}|$, and let N denote the maximum number of colors in the range of g_V over all $V \in \mathcal{V}$. Furthermore, assume that Algorithm* 4.1 *is applied to a K-span ordering of $G^{\mathcal{V}}$. Then Algorithm* 4.1 *runs in time $O(n + m + rK^{2.5}N^{2.5})$.*

4.3. *Möhring's Algorithm*. One of the main parts of the STRIPE algorithm is the coloring of large subgraphs called "stripes" which are cocomparability graphs. For the purpose of coloring these subgraphs we employ a general cocomparability graph coloring algorithm due to Möhring [18] which is sketched in the following. Some details specific to the case of stripes in UD graphs are covered in Section 4.4.

The basic idea behind Möhring's algorithm is to associate with each cocomparability graph G a corresponding network N with lower bounds on the edge capacities such that each minimum flow in N corresponds to an optimal coloring of G and vice versa. Möhring's definition of the network N is as follows. Let G be a nonempty cocomparability graph, let $H = \overline{G}$ be its complement, and let R be a transitive orientation of H. N is a directed graph defined by the following rules:

- For each $v \in V(G)$ pick two distinct vertices v' and v'' connected by an edge $v'v''$ in N. We refer to v' and v'' as the *in-* and *out-vertices* for v, respectively.
- For each edge $vw \in R$, let the corresponding out-vertex v'' and the in-vertex w' be connected by an edge $v''w'$ in N.
- Furthermore, N contains two additional vertices, s (the source) and t (the sink), with edges $sv',\ w''t$ for all in-vertices v' and out-vertices w'' in N, respectively.[3]

Given Möhring's network N for a cocomparability graph G, we assign a lower bound $b(v'v'') = 1$ to each edge $v'v''$ connecting a pair of corresponding in- and out-vertices, and zero bounds to all other edges of N. We refer to these bounds as the *standard bounds* for N. The notion of a *flow* φ through the network N is defined as usual, and we say that φ is an *admissible k-flow* iff in addition $\varphi(vw) \geq b(vw)$, $\forall vw \in E(G)$ and the *value* $\varphi(N)$ of the flow (i.e., the total flow emanating from the source s which equals the total flow into the sink t) is at most k. What Möhring has proved is the following:[4]

THEOREM 4.1 [18]. *Let N be the network for a cocomparability graph G with standard bounds b. Then G has a k-coloring f iff N has an admissible k-flow φ.*

Using Möhring's construction, we can actually find an optimal coloring of a cocomparability graph G by determining an admissible flow for the corresponding network N such that $\varphi(N)$ is minimum. This can be done by standard flow techniques. The actual construction of a k-coloring from the corresponding admissible k-flow is fairly straightforward as there is a simple correspondence between admissible flows in the network and color classes in the graph. The algorithm can easily be derived from Algorithm 1.27 of [18].

Conversely, we can also derive an admissible flow φ from each coloring of the original cocomparability graph G. This is useful for computing an initial admissible flow whose value can be reduced along "augmenting paths" as usual, see [3]. (This is in contrast to the standard flow *maximization* problem in which we can simply start off with a zero flow. For our purposes, we have to provide an initial flow function which meets the nonzero bound requirements on the edges between corresponding in- and out-vertices.)

[3] Möhring's original definition actually states that s is only connected to the in-vertices of "minimal" vertices v with respect to R, and t only to "maximal" out-vertices. Since we introduce techniques to reduce the number of edges in the network later, we omit this detail here, in order to simplify matters.

[4] Möhring actually considers the *weighted coloring problem* for cocomparability graphs. The ordinary coloring problem is a special form of this generalized coloring problem in which all vertices have unit weight, see [18] for details.

4.4. *The STRIPE Algorithm*. We are now ready to present our STRIPE algorithm for coloring UD graphs. Let x_v and y_v denote the x- and y-coordinates of a point v in the plane. Given a set of points V, we define $\min_x V$ and $\max_x V$ as the minimal and maximal x-coordinates of the points in V. The values $\min_y V$ and $\max_y V$ are defined analogously. The horizontal and vertical *intervals* covered by V are given by $I_x(V) = [\min_x V, \max_x V]$ and $I_y(V) = [\min_y V, \max_y V]$. Finally, the *length* or *size* of an arbitrary closed interval I is denoted $|I| = \max I - \min I$, and the *width* and *height* of V are defined as $\mathrm{wd}(V) = |I_x(V)|$ and $\mathrm{ht}(V) = |I_y(V)|$.

DEFINITION 4.2. A UD graph G is called a c-*stripe* (for some $c \geq 0$) iff it has a distance model $(V, 1)$ such that V has width at most c.

The basis of the STRIPE algorithm is the following observation:

LEMMA 4.1 (Stripe Lemma). *$\sqrt{3}/2$-stripes are cocomparability graphs.*

PROOF. Let $G = G(V, 1)$ such that $\mathrm{wd}(V) \leq c{:=}\sqrt{3}/2$, and let $H = \overline{G}$. Define an orientation R of the edges of H by

$$R = \{vw \in E(H) | y_v < y_w\}.$$

Note that R is well defined since $y_v = y_w$ implies that $vw \notin E(H)$ because $|x_v - x_w| \leq c < 1$. In order to prove that G is a cocomparability graph we show that R is a transitive relation on V.

Assume that $uv, vw \in R$ and let $a{:=}y_v - y_u,\ b{:=}y_w - y_v$. Since $d(u, v), d(v, w) > 1$ and $\mathrm{wd}(V) \leq c$ we have that $a^2, b^2 > 1 - c^2$. Thus $d(u, w)^2 \geq (a + b)^2 > 4 - 4c^2$. Since $4 - 4c^2 \geq 1 \Leftrightarrow c \leq \sqrt{3}/2$, the assertion follows. □

Note that the proof of Lemma 4.1 shows that a transitive orientation of the complement of a $\sqrt{3}/2$-stripe can actually be obtained by ordering the vertices with respect to y-coordinates. We also remark that the construction in the proof of the Stripe Lemma may fail for each $c > \sqrt{3}/2$. In this sense, the value $c = \sqrt{3}/2$ is optimal.

Lemma 4.1 implies that $\sqrt{3}/2$-stripes are perfect and therefore $\chi(G) = \omega(G)$ for each $\sqrt{3}/2$-stripe G. Moreover, $\sqrt{3}/2$-stripes can be colored optimally using Möhring's algorithm, in time $O(|V(G)|^3)$ (see [18]). In the case of $\sqrt{3}/2$-stripes, the running time can actually be improved to $O(|V(G)|\omega(G)^2)$. This is achieved by applying two simple kinds of optimizations:

- Start from a good initial flow, obtained from an $O(\omega(G))$-coloring of G which can be constructed using the sequential algorithm. This reduces the number of augmentation steps in the flow optimization phase of Möhring's algorithm to $O(\omega(G))$.
- Reduce the number of edges in the network N associated with the $\sqrt{3}/2$-stripe G. By introducing an additional "pipeline" into the network, it is possible to get around with only $O(|V(G)|\omega(G))$ edges. This reduces the complexity of each augmentation step (which essentially is a connectivity problem in a directed graph derived from the network) to $O(|V(G)|\omega(G))$.

We describe each of these in turn. Note that the bound $O(|V(G)|\omega(G)^2)$ is usually an actual improvement over $O(|V(G)|^3)$, since the $\omega(G)$ term only depends on the chromatic number we want to compute, but is otherwise independent of the actual size of the stripe.

We first discuss how to obtain a good initial coloring of a $\sqrt{3}/2$-stripe G. Since G in particular is a UD graph, we know that we can color G with at most $3\omega(G) - 2$ colors using the sequential algorithm, applied to a suitable vertex ordering which can be computed in linear time. In fact, we can show that it is possible to reduce the constant factor to 2:

LEMMA 4.2. *For each $\sqrt{3}/2$-stripe G*, $\mathrm{sp}(G) \le 2\omega(G) - 2$.

PROOF. Let $G = G(V, 1)$ with $\mathrm{wd}(V) \le \sqrt{3}/2$. Consider the lexicographic ordering $<$ on V which orders the members of V first by y- and then by x-coordinates.

Let $v \in V$. Then all neighbors $w < v$ of v, as well as v itself, are contained in the half-circle $C = \{u | y_u \le y_v \wedge d(v, u) \le 1\}$. We can cover $C \cap V$ by two rectangles R_1 and R_2 of width $\sqrt{3}/2$ and height $\frac{1}{2}$. Note that the maximum distance of any two points in R_i is 1 ($i = 1, 2$), and hence $R_1 \cap V$ and $R_2 \cap V$ are cliques of G. In fact, $(R_i \cap C \cap V) \cup \{v\}$ is a clique of G for $i = 1, 2$, and thus $|C \cap V \setminus \{v\}| \le 2\omega(G) - 2$. This holds for each $v \in V$, and hence $<$ has span $\le 2\omega(G) - 2$. $\square$

Next we discuss how to reduce the number of edges in the network associated with a $\sqrt{3}/2$-stripe G. Consider a stripe $G = G(V, 1)$ with $\mathrm{wd}(V) \le \sqrt{3}/2$. We partition G into subgraphs $G_{V_1}, \dots, G_{V_r}$ as follows. Let $y_{\min} = \min_y V$, $y_{\max} = \max_y V$, and $r = \lceil 2(y_{\max} - y_{\min}) \rceil$. We define V_i as

$$V_i = \{v \in V | i = \lfloor 2(y_v - y_{\min}) \rfloor + 1\}.$$

In other words, G is partitioned along the y-axis into "buckets" of height $\le \frac{1}{2}$. Since $(\sqrt{3}/2)^2 + (\frac{1}{2})^2 = 1$, we have that each V_i is a clique of G or, equivalently, an independent set of $\overline{G}$. Also, we have that $d(v_i, v_j) > 1$ for all $v_i \in V_i,\ v_j \in V_j$ with $|i - j| > 2$. Note that by the proof of Lemma 4.1 a transitive orientation R of $\overline{G}$ is given by $R = \{v_i v_j | v_i \in V_i, v_j \in V_j, i < j\}$. We directly construct the network N from the point set V. The basic idea is to introduce an additional *pipeline* into the network which enables us to eliminate all the "trivial" edges vw in $\overline{G},\ v \in V_i,\ w \in V_j,\ |i - j| > 2$. The network is defined as follows. To distinguish it from Möhring's network, we call it the *stripe network*.

- For each $v \in V$, pick in- and out-vertices v' and v''. Each pair of corresponding in- and out-vertices v' and v'' is connected by an edge $v'v''$.
- Furthermore, N contains $r + 3$ distinct *pipeline vertices* $w_1, \dots, w_{r+3}$, with edges $w_i w_{i+1}$ $(1 \le i < r + 3)$ connecting each pair of consecutive vertices.
- The out-vertex v'' of each $v \in V_i$ is connected by an edge $v''w'$ to all in-vertices $w',\ w \in V_j$ with $0 < j - i \le 2$ and $d(v, w) > 1$.
- Each pipeline vertex w_i is connected by an edge $w_i v'$ to all in-vertices v' for $v \in$

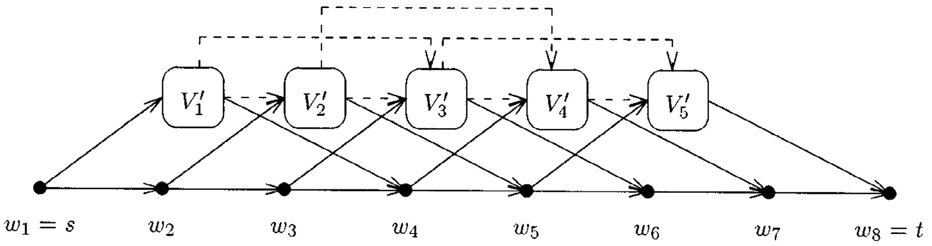

Fig. 9. Stripe network.

V_i $(1 \leq i \leq r)$. Furthermore, there is an edge $v''w_{i+3}$ between each out-vertex v'' of $v \in V_i$ and pipeline vertex w_{i+3} $(1 \leq i \leq r)$.

- The source of N is $s = w_1$, and $t = w_{r+3}$ is its sink.

A sketch of the construction for the case of five buckets is shown in Figure 9. By V_i' we denote the set of in- and out-vertices of bucket V_i. Solid arrows are used to indicate the edges between pipeline vertices and connections between pipeline vertices and in- and out-vertices of the corresponding buckets, whereas dashed arrows represent the "internal" edges between the in- and out-vertices of three consecutive buckets.

Note that for each $s-t$ path in Möhring's network there is a corresponding path in the stripe network which goes through the same edges between pairs of corresponding in- and out-vertices, and vice versa. Thus, if we assign bounds $b(v'v'') = 1$ to edges between pairs of corresponding in- and out-vertices and zero bounds to the remaining edges, the stripe network has an admissible k-flow iff Möhring's network has one. Summarizing the results obtained so far, we have the following:[5]

THEOREM 4.2. *$\sqrt{3}/2$-stripes G can be colored optimally in time $O(|V(G)|\omega(G)^2)$.*

We still have to discuss how to partition a UD graph G into a collection of $\sqrt{3}/2$-stripes, and how to combine the individual stripe colorings obtained with Möhring's algorithm with the SPC algorithm.

DEFINITION 4.3 (Stripe Partition). Let $G = G(V, d)$ be a UD graph. A *c-stripe partition* is a partition $\mathcal{V}$ of G such that each $V \in \mathcal{V}$ has width $\leq cd$.

One way to determine a suitable stripe partition of a given UD graph is the following "bucket sort" method. It can easily be implemented in linear time (with respect to the

[5] Note that in the construction of the stripe network we carry out a constant number of arithmetic operations involving model coordinates for each vertex of G. The total time required by these operations is linear with respect to the size of the (encoding of the) model. However, in the following we simplify matters by assuming that model coordinates, etc., can be represented in constant space and thus each arithmetic operation is carried out in constant time.

size of the encoding of the UD model). Let the model (V, d) and $c > 0$ be given. We construct the *canonical c-stripe partition* $\mathcal{V}$ of $G = G(V, d)$ as follows:

1. Let $x_{\min} = \min_x V,\ x_{\max} = \max_x V$, and $r = \lceil (x_{\max} - x_{\min})/cd \rceil$.
2. Let $V_i = \{v \in V | i = \lfloor (x_v - x_{\min})/cd \rfloor + 1\},\ i = 1, \ldots, r$.
3. Let $\mathcal{V} = \{V_1, \ldots, V_r\}$.

Note that each member V_i of the canonical c-stripe partition $\mathcal{V}$ indeed induces a c-stripe on G. Using the ordering $V_1 < V_2 < \cdots < V_r$ we obtain that $G^{\mathcal{V}}$ has span $\leq K$, where $K{:=}\lceil 1/c \rceil$ because $V_i V_j \notin E(G^{\mathcal{V}}),\ \forall |i - j| > \lceil 1/c \rceil$. In particular, for $\frac{1}{2} \leq c \leq \sqrt{3}/2$ we have that $K = 2$ and hence G can be $3\omega(G)$-colored. The STRIPE algorithm can now be stated as follows:

ALGORITHM 4.2 (STRIPE Algorithm)

Input: A UD graph G, given by its model (V, d).
Output: A coloring f of G.

Method:
1. Construct the canonical c-stripe partition $\mathcal{V}$ of G, $\frac{1}{2} \leq c \leq \sqrt{3}/2$.
2. For each $V \in \mathcal{V}$, use Möhring's algorithm to compute an optimal coloring g_V of G_V:
   (a) Compute a $2\omega(G_V)$-coloring h_V of G_V using the sequential coloring algorithm.
   (b) Compute the stripe network N_V for G_V.
   (c) From h_V compute an initial admissible $2\omega(G_V)$-flow φ_V^0 for N_V.
   (d) Reduce the value of φ_V^0 using standard flow techniques, yielding an optimal $(\chi(G_V)$-) flow φ_V for N_V.
   (e) Compute g_V from φ_V.
3. Combine the stripe colorings $g_V,\ V \in \mathcal{V}$, obtained in step 1 to a coloring f of G by applying Algorithm 4.1 to $G,\ \mathcal{V},\ g_V,\ V \in \mathcal{V}$, ordering the members of $\mathcal{V}$ from "left to right."

For the purpose of analyzing Algorithm 4.2, let G be a UD graph, $n = |V(G)|$, $m = |E(G)|$, and $\omega = \omega(G)$. Step 1 can be done in time $t_1 = O(n)$ (under our assumption that arithmetic operations are carried out in constant time). Over all $V \in \mathcal{V}$, step 2 can be done in time $t_2 = O(n\omega^2)$ by Theorem 4.2. Since $m = O(n\omega)$, by Proposition 4.2 step 3 can be carried out in time $t_3 = O(n\omega + r\omega^{2.5})$, where $r = |\mathcal{V}|$ is the size of the stripe partition. We obtain the following result:

THEOREM 4.3. *Let G be a UD graph, n, m, ω, and r as above. Then Algorithm* 4.2 *colors G with at most* 3ω *colors in time* $O(n\omega^2 + r\omega^{2.5})$.

We remark that the cocomparability graph coloring problem can also be solved using bipartite matching as discussed in Section II.8 of [4]. If we replace our version of Möhring's algorithm with this method, step 2 of Algorithm 4.2 can be done in time $t_2' = O(n^{2.5})$ which is better than the bound $t_2 = O(n\omega^2)$ if $\omega > n^{3/4}$.

5. Concluding Remarks. In this paper we considered the coloring problem for UD graphs. Improving on a result of Clark et al. [2], we have shown that the UD graph coloring problem remains NP-complete for *each* fixed number of colors $k \geq 3$.

We then considered the problem of *approximating* the chromatic number of UD graphs. We first reviewed existing sequential coloring techniques for UD graphs. Then we discussed an alternative 3-approximation algorithm for the UD graph coloring problem, the so-called *STRIPE algorithm*. In contrast to the sequential coloring algorithm the STRIPE algorithm makes extensive use of the UD model which must be given with the input graph. The basic idea behind the STRIPE algorithm is to partition the input graph into a sequence of induced subgraphs called *stripes*. With an appropriate choice of parameters, the stripes become cocomparability graphs which can be colored optimally using Möhring's algorithm [18]. The individual stripe colorings are then combined with the permutative coloring technique [7].

We remark that the STRIPE algorithm can be extended to more general kinds of models (such as “double disk” graphs) and generalized graph coloring problems (such as the “minimum distance” coloring problem); see [9] and [13] for further details.

We implemented both the STRIPE algorithm and Peeters' version of the sequential UD graph coloring algorithm, in order to estimate the practical performance and efficiency of the STRIPE algorithm and to compare both approaches. Our experimental results indicate that both algorithms usually achieve comparable performance ratios, with the STRIPE algorithm slightly outperforming the sequential algorithm when applied to realistic graphs arising in channel assignment problems in which the points are distributed nonuniformly [9].

As we pointed out, the UD model is an essential prerequisite for UD graph algorithms which make extensive use of the geometric structure of the graph. Therefore finding heuristics for the construction of approximate UD graph models is an interesting field for further research. Such techniques would have other useful applications, such as the visualization of graphs which have some apparently geometric structure, but are given without a model.

A major open question is whether the bound $3\omega(G) - 2$ on the chromatic number of a UD graph can be further improved. We do not know of any UD graph where this large gap between clique and chromatic number actually occurs. Because the clique number of a UD graph can be computed in polynomial time, any better bound on the chromatic number in terms of the clique number would enable us to give a tighter approximation of the chromatic number of a UD graph.

Acknowledgments. Thanks are due to Clemens Lautemann for his support, criticism, and suggestions which helped to improve various parts of this paper, and to Achim Quellmalz, Anne Knälmann, and Bernd Müller (our project partners from the Technical Administration of the German Südwestfunk Public Broadcasting Corporation) for fruitful discussions. We would also like to thank Frank Wankmüller whose efforts made this research project possible. Last but not least we wish to thank the anonymous referees for their helpful comments on earlier drafts of this paper, and for pointing out the alternative cocomparability graph coloring algorithm mentioned in Section 4.4.

References

[1] H. Breu and D. G. Kirkpatrick. Unit disk graph recognition is NP-hard. Technical Report 93-27, Department of Computer Science, University of British Columbia, 1993.

[2] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.

[3] S. Even. *Graph Algorithms*. Pitman, London, 1979.

[4] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, 1962.

[5] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[6] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Computer Science and Applied Mathematics. Academic Press, New York, 1980.

[7] A. Gräf. Permutative graph colorings. Musikinformatik und Medientechnik 5/92, Johannes Gutenberg-Universität, Musikwissenschaftliches Institut, Bereich Musikinformatik, Mainz, 1992.

[8] A. Gräf. Coloring and Recognizing Special Graph Classes. Ph.D. thesis, Johannes Gutenberg-Universität, Mainz, 1995. Published as Report No. 20/95 in the series Musikinformatik und Medientechnik.

[9] A. Gräf, M. Stumpf, and G. Weißenfels. On coloring unit disk graphs. Musikinformatik und Medientechnik 17/94, Johannes Gutenberg-Universität, Musikwissenschaftliches Institut, Bereich Musikinformatik, Mainz, 1994.

[10] W. Hale. Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 68:1497–1514, 1980.

[11] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2:225–231, 1973.

[12] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. In *Proc*. 25*th Annual ACM Symposium on the Theory of Computing*, pages 286–293. ACM, New York, 1993.

[13] E. Malesinska, S. Piskorz, and G. Weißenfels. On the approximability of coloring problems for disk graphs. Musikinformatik und Medientechnik 26/96, Johannes Gutenberg-Universität, Musikwissenschaftliches Institut, Bereich Musikinformatik, Mainz, 1996.

[14] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. Unpublished manuscript (see also [15]).

[15] M. V. Marathe, H. B. Hunt III, and S. S. Ravi. Geometry based approximations for intersection graphs. In *Proc*. *Fourth Canadian Conference on Computational Geometry*, pages 244–249, 1992.

[16] D. W. Matula and L. L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the Association of Computing Machinery*, 30(3):417–427, 1983.

[17] D. W. Matula, G. Marble, and J. Isaacson. Graph coloring algorithms. In R. C. Read, editor, *Graph Theory and Computing*, pages 108–122. Academic Press, New York, 1972.

[18] R. H. Möhring. Algorithmic aspects of comparability graphs and interval graphs. In I. Rival, editor, *Graphs and Orders*, pages 41–101. Reidel, Dordrecht, 1985.

[19] R. Peeters. On coloring j-unit sphere graphs. FEW 512, Department of Economics, Tilburg University, 1991.