Channel assignment for digital networks: a bound and an algorithm

Jeannette Janssen
Mark MacIsaac
Kyle Schmeisser

Technical Report CS-2002-03

May, 2002

Faculty of Computer Science
6050 University Ave., Halifax, Nova Scotia, B3H 1W5, Canada

Channel assignment for digital broadcasting: a bound and an algorithm

Jeannette Janssen Mark MacIsaac Kyle Schmeisser

May 27, 2002

Abstract

The problem of channel assignment in digital networks can be formulated as follows. Services to be broadcast at a transmitter must be packed into blocks, and each block must be assigned a frequency channel. This assignment must satisfy bandwidth and interference constraints. The objective is to minimize spectrum use.

Mathematically, the problem corresponds to a generalized graph colouring problem with a flavour of bin packing. In this paper, we give a new lower bound on spectrum use for this problem, derived from intersecting cliques. We also give a near-optimal, efficient algorithm to assign frequency channels to blocks for the case where the interference graph is a cycle.

1 Introduction

A new generation of audio and video broadcasting networks, using digital technology, is currently being implemented in many countries. Like other wireless networks, digital networks are subject to severe restrictions on the amount of spectrum available for transmission. Consequently, it is important to find assignments of frequency channels to transmitted services which minimize the total amount of spectrum used. Mathematically, this amounts to an optimization problem which incorporates aspects of graph colouring and bin packing.

The graph theoretic aspect of the assignment problem arises due to the possibility of interference between transmitters. Two transmitters in areas close enough to cause interference cannot use the same frequency channel (an exception can be made when two identical signals are transmitted, as will be explained later). The interference constraints can be modelled by an *interference graph*. The vertices of the interference graph correspond to the transmitters, and two vertices are adjacent precisely when the corresponding transmitters can interfere.

The problem as described thus far falls into the class of frequency assignment problems as seen in other wireless networks (for example, cellular telephone networks). In such networks, a frequency assignment corresponds to a colouring or multicolouring of the interference graph. However, the technology used in digital broadcasting networks has two features which cause the frequency assignment problem to be different.

One distinguishing feature of digital broadcast networks is that a channel can be used to transmit more than one signal. For example, in a European network soon to become operational, up to six radio stations can be transmitted on one channel. This has opened the possibility to transmit additional data services, such as stock market information or weather reports, over the same channel. Such services all may have different bandwidth, and hence occupy a different portion of the bandwidth of the channel. This adds an aspect of bin packing to the channel assignment problem.

The other distinguishing feature of digital broadcasting is the use of so-called *single frequency networks*. If the *same* set of services is transmitted at two interfering transmitters, then the same channel may be used at both transmitters. This is not true in traditional networks: the same FM radio station must have different frequencies in adjacent regions, for example. In digital networks, a set of services be allotted the same frequency channel in a cluster of adjacent areas; the term 'single frequency network' refers to such a cluster. Note that optimal packing of services and optimal use of single frequency networks may be contradictory goals: if many services are packed together to be transmitted on one channel, then this same set may not occur in many other areas, so the use of single frequency networks will be restricted.

The first mathematical formulation of the channel assignment problem for digital networks was given by Gräf in [4]. This paper also gives some initial bounds and heuristics for the problem. An account of more sophisticated heuristics and experiments can be found in [5] and [7]. The related problem of channel assignment in cellular networks has been well studied. An overview of recent work can be found in [8].

In this paper, we will first give a new lower bound on the minimum number of frequency channels needed for a given digital broadcasting (DB) problem. The lower bound is based on a subgraph of the interference graph consisting of a number of cliques with a common intersection. This bound will be discussed in Section 2.

In Section 3, we will focus on the special case of the DB channel assignment problem where each service needs a bandwidth equal to the bandwidth of a channel. Since the objective in this case is to assign "colour" (frequency channels) to the services at the vertices of a graph, we refer to this as the *service colouring* problem. The reasons for our focus on this problem are twofold. Firstly, the service colouring problem is equivalent to the problem of finding a frequency assignment for a given block assignment. Most known DB channel assignment algorithms take a two step approach. First a block assignment is found. In the next step frequencies are assigned to the blocks, in other words, a service colouring is found. Secondly, a focus on service colouring gives insight in the general problem since it enables us to isolate which difficulties of the DB channel assignment problem are due to the bin packing aspect, and which to the graph colouring aspect and the possibility to use single frequency networks.

Service colouring is a special case of graph colouring. Any service colouring of a graph with services assigned to its vertices can be transformed into a standard vertex colouring of a larger graph where vertices represent vertex-service pairs. However, nothing is to be gained from this process, since graph colouring for graphs in general is hard, while the transformation will obscure any special structure that the original graph may have. This is especially relevant since interference graphs of broadcasting networks typically do fall into special graph classes such as planar graphs or unit disk graphs.

We present algorithms to solve the service colouring problem for the cases where the interference graph is a tree or a cycle. It follows easily that a greedy algorithm is optimal for trees. For cycles, we give a more complicated, quadratic algorithm which has performance ratio at most $1 +$

$\frac{2}{n-1}$ (i.e. which uses at most $(1+\frac{2}{n-1})$ times the optimal number of colours), where n is the length of the cycle.

2 Problem definition and lower bounds

Before we can describe our lower bound, we must first define the problem formally.

Definition 2.1 *A* DB channel assignment problem $(G, \mathcal{R}, \mu)$ *consists of the following:*

- *An interference graph* $G = (V, E)$*,*
- *A collection* $\mathcal{R} = \{R_v \mid v \in V\}$*, where* R_v *is the set of services required at* v*,*
- *A function* $\mu : \bigcup_{v \in V} R_v \to [0, 1]$ *giving the bandwidth, relative to the bandwidth of a channel, required for each service.*

The notation $S_{\mathcal{R}}$ *will be used to denote the total set of services, so* $S_{\mathcal{R}} = \bigcup_{v \in V} R_v$*. For any set* $W \subseteq V$ *of vertices, the set of services required on any vertex of* W *will be denoted as* $R_W = \bigcup_{v \in W} R_v$*. For a service* s*,* $\mu(s)$ *will be referred to as the* size *of* s*.*

A channel assignment *of* $(G, \mathcal{R}, \mu)$ *is a pair* $(\mathcal{B}, f)$*, where* $\mathcal{B}$ *is a block assignment and* f *is a frequency assignment.*

A block assignment *is a collection of sets* $\mathcal{B} = \{\mathcal{B}_v \mid v \in V\}$*, so that for each* $v \in V$*,* $\mathcal{B}_v \subseteq 2^{S_{\mathcal{R}}}$ *and the following properties hold:*

(i) $R_v \subseteq \bigcup_{B \in \mathcal{B}_v} B$*, and*

(ii) for all $B \in \mathcal{B}_v$*,* $\sum_{s \in B} \mu(s) \leq 1$*.*

Any set $B \in \mathcal{B}_v$ *(for some vertex* v*) is called a* block*.*

A frequency assignment *is a function* $f : \{(v, B) \mid v \in V,\, B \in \mathcal{B}_v\} \to \mathbb{N}$ *so that for all vertices* $v, w \in V$ *so that* $v = w$ *or* v *is adjacent to* w*, and* *for all blocks $B \in \mathcal{B}_v$, $B' \in \mathcal{B}_w$, if $B \neq B'$ then $f(v, B) \neq f(w, B')$. The value of $f(v, B)$ denotes the frequency channel used to transmit all services in B at the transmitter corresponding to v.*

Given a channel assignment $(\mathcal{B}, f)$ for $(G, \mathcal{R}, \mu)$, the number of frequency channels used, namely $|\{f(v, B) \mid v \in V,\ B \in \mathcal{B}_v\}|$, will be denoted by $|(\mathcal{B}, f)|$.

The minimum number of frequency channels needed for any channel assignment of $(G, \mathcal{R}, \mu)$ will be denoted by $\chi_{db}(G, \mathcal{R}, \mu)$. So

$$\chi_{db}(G, \mathcal{R}, \mu) = \min\{|(\mathcal{B}, f)| : (\mathcal{B}, f) \textit{ is a channel assignment for } (G, \mathcal{R}, \mu)\}.$$

If $(G, \mathcal{R}, \mu)$ is a DB channel assignment problem, and C is a clique in G (a clique is a set of mutually adjacent vertices), then no frequencies can be reused on C, so in any channel assignment, all blocks assigned to the vertices of C must receive different frequencies. Therefore, the number of frequency channels needed is at least the minimum number of blocks needed to pack the services on all vertices of C. This leads to a lower bound on $\chi_{db}(G, \mathcal{R}, \mu)$, which was first formulated in [4].

Since the services all have different sizes, the minimum amount of blocks needed to pack all services is non-trivial to calculate. The calculation corresponds to a bin packing problem, which is known to be NP-hard [3]. However, good lower bounds and approximation algorithms exist (see for example [1, 2]). Moreover, the number of different service sizes encountered in typical channel assignment problems may be small enough to make an exhaustive computation of the optimal bin packing possible. We will give all bounds in this section relative to the optimal bin packing.

Some notation is needed. A *bin packing* problem can be characterized by a pair (A, μ), where A is a set, and $\mu : A \to [0, 1]$ is a function which assigns a size to each element of A. The *optimal bin packing number* $p(A, \mu)$ is the minimal number of unit-sized bins needed to pack all items of A. Obviously, $p(A, \mu) \geq \lceil \sum_{s \in A} \mu(s) \rceil$. Using this notation, we can state the clique bound discussed above.

Proposition 2.2 *(from [4]) For any DB channel assignment problem $(G, \mathcal{R}, \mu)$,*

$$\chi_{db}(G, \mathcal{R}, \mu) \geq \max\{p(R_C, \mu) \mid C \textit{ is a clique of } G\}.$$

We will refer to this bound as the *clique bound* of $(G, \mathcal{R}, \mu)$. A more complicated bound can be derived from a collection of intersecting cliques.

Theorem 2.3 *Let $(G, \mathcal{R}, \mu)$ be a DB channel assignment problem, and let $W \subseteq V(G)$ be a collection of k intersecting cliques $C_1, \ldots, C_k$. Let $A = C_1 \cap \ldots \cap C_k$. Then,*

$$\chi_{db}(G, \mathcal{R}, \mu) \geq (1 - \frac{1}{k})p(R_A, \mu) + \frac{1}{k}p(R_W, \mu).$$

Proof. Let $(G, \mathcal{R}, \mu)$ and $C_1, \ldots, C_k$, A be as in the statement of the theorem. Let $(\mathcal{B}, f)$ be a DB channel assignment for $(G, \mathcal{R}, \mu)$. Let $\mathcal{B}_A = \bigcup_{v \in A} \mathcal{B}_v$ be the collection of blocks assigned to any vertex in A. This collection must cover all services required on A, so $R_A \subseteq \bigcup_{B \in \mathcal{B}_A} B$. This implies that there must be at least as many blocks as minimally needed to pack all services in R_A, so $|\mathcal{B}_A| \geq p(R_A, \mu)$.

Similarly, let $\mathcal{B}_{W-A} = \bigcup_{v \in W-A} \mathcal{B}_v$ be the set of blocks assigned to vertices of $W - A$. Obviously, $\mathcal{B}_A$ and $\mathcal{B}_{W-A}$ are disjoint, and together they form a packing of all services in R_W. So $|\mathcal{B}_A| + |\mathcal{B}_{W-A}| \geq p(R_W, \mu)$.

Any frequency channel assigned by f to any block in $\mathcal{B}_A$ can only be used once, because vertices in A are adjacent to all vertices in W. On the other hand, one frequency channel may be assigned to up to k blocks from $\mathcal{B}_{W-A}$. Namely, it is possible to choose k mutually non-adjacent vertices from the cliques $C_1, \ldots, C_k$, respectively, so blocks assigned to these vertices may all share the same channel.

Therefore, the minimum number of frequency channels required to accommodate all services on W is $|\mathcal{B}_A| + (1/k)|\mathcal{B}_{W-A}| \geq (1 - \frac{1}{k})p(R_A, \mu) + \frac{1}{k}p(R_W, \mu)$. $\square$

Consider the following example. Let $(G, \mathcal{R}, \mu)$ be a DB channel assignment problem which contains a set W which is the intersection of k cliques $C_1, \ldots, C_k$, as described in Theorem 2.3. Suppose R_A consists of n services, each service s of size $\mu(s) = 1/2 - \epsilon$, for some ϵ so that $0 < \epsilon < 1/2$. For each clique C_i, let $O_i = R_{C_i} - A$. For each i, suppose O_i consists of n services of size $1/2 + \epsilon$. Thus, the clique bound indicates that $\chi_{db}(G, \mathcal{R}, \mu) \geq n$. The bound given in Theorem 2.3 gives a higher bound. Note that $p(R_A, \mu) = \frac{n}{2}$ and $p(R_W, \mu) = kn$. The bound then states:

$$\chi_{db}(G, \mathcal{R}, \mu) \geq (1 - \frac{1}{k})\frac{n}{2} + \frac{1}{k}(kn) = n + \frac{1}{2}(1 - \frac{1}{k})n.$$

An optimal DB channel assignment for $(G, \mathcal{R}, \mu)$ will pack $\frac{n}{k}$ services from each set O_i together with services from R_A in blocks containing two services each, and pack all remaining services in blocks containing only one service.

Then, n channels are assigned to the blocks containing services from R_A, while $(1 - \frac{1}{k})n$ channels suffice to colour the remaining blocks, since each of these channels can be reused in every clique. The total number of channels used equals $n + (1 - \frac{1}{k})n$, exceeds our bound. It is easy to see that no assignment using less channels is possible. This example thus shows that the bound from Theorem 2.3 is not optimal, but can be better than the clique bound.

3 Service colouring

In this section we discuss the DB channel assignment problem in the special case where each service s has size $\mu(s) = 1$. In other words, channels can be used only to transmit exactly one service. As discussed in the introduction, this problem corresponds to the problem of finding a frequency assignment for a given block assignment.

Since μ is constant, we can define a service colouring problem by a pair $(G, \mathcal{R})$, where $G = (V, E)$ is a graph, and $\mathcal{R}$ is an assignment of sets of services $\mathcal{R} = \{R_v \,|\, v \in V\}$ to the vertices of G. A pair $(G, \mathcal{R})$ will be referred to as a *service graph*. A *service colouring* f of $(G, \mathcal{S})$ is an assignment of a colour $f(v, s)$ to each pair (v, s) where $v \in V$ and $s \in R_v$, so that $f(v, s) \neq f(w, t)$ whenever $v = w$ or v is adjacent to w, and $s \neq t$. The objective of the service colouring problem is to use the minimum number of colours.

As noted in the introduction, a service colouring of $(G, \mathcal{R})$ corresponds to a standard vertex colouring of a graph whose vertices are vertex-service pairs of $(G, \mathcal{R})$. This graph will be called $G_{\mathcal{R}}$: the vertices of $G_{\mathcal{R}}$ consist of all pairs (v, s) where $v \in V$ and $s \in R_v$, and two pairs (v, s) and (w, t) are adjacent precisely when $s \neq t$ and $v = w$ or v is adjacent to w. This section presents efficient algorithms for service colouring if G has special structure; this structure may not be apparent if we only consider $G_{\mathcal{R}}$.

Proposition 3.1 *Given a tree T and a service assignment $\mathcal{R}$ for T, an optimal service assignment for $(T, \mathcal{R})$ can be found greedily in linear time.*

Proof. Let T and $\mathcal{R}$ be as stated. An optimal colouring f of $(T, \mathcal{R})$ can be found as follows. Order the vertices of T in a manner consistent with their distance from a root vertex v_0. Starting at v_0 and following this ordering, assign colours to services in a greedy manner. Note that each vertex v, when considered, will have at most one neighbour w whose services are already coloured. For each service $s \in R_w \cap R_v$, let $f(v, s) = f(w, s)$. Each service in $R_v - R_w$ receives the lowest indexed colour not used for any pair (w, s), $s \in R_w$. Clearly, a colouring thus obtained never uses more than $\max_{v \sim w} |R_v \cap R_w|$ colours, which, by Proposition 2.2, is optimal.

Assuming the service sets are ordered, colouring the services at a vertex v with coloured neighbour w will take $\mathcal{O}(|R_v| + |R_w|)$ operations. The colouring thus takes $\mathcal{O}(N)$ operations, where $N = \sum_v |R_v|$, the size of the input. □

Note that the lower bound $\max_{v \sim w} |R_v \cap R_w|$ equals the clique number of $G_{\mathcal{R}}$. This proposition thus translates into the following statement about the expanded graph $G_{\mathcal{R}}$.

Corollary 3.2 *If G is a tree, and $\mathcal{R}$ is a service assignment for T, then $G_{\mathcal{R}}$ is a perfect graph.*

This corollary also follows from the fact (easily verified) that $G_{\mathcal{R}}$ is weakly chordal if G is a tree. The perfection of weakly chordal graphs was established in [6].

For the remainder of this section, we consider the case where G is a cycle. A service graph $(G, \mathcal{R})$ where G is a cycle will be referred to as a *service cycle*. In the following, n will be used exclusively to denote the length of the cycle, and the vertices of the cycle are given as $v_1, \ldots v_n$, according to their placement around the cycle. Addition of the indices is assumed to be modulo n; for example, if $i = n$ then $v_{i+1} = v_1$. The range of an index is also considered modulo n. For example, if $i_0 = n-1$ then $\{i \,|\, i_0 \le i \le i_0 + 3\} = \{n-1, n, 1, 2\}$. Finally, we will use R_i instead of R_{v_i} to denote the service set of v_i.

Our service colouring algorithm for cycles is based on the construction, in each iteration, of a set of services which can all receive the same colour. Such a set will be called an *independent service set* (ISS). An ISS is a pair (I, σ) where $I \subseteq \{1, 2, \ldots, n\}$ and $\sigma : I \to \bigcup_i R_i$ is a function so that $\sigma(i) \in R_i$ for all $i \in I$, and if $\{i, i+1\} \subseteq I$ then $\sigma(i) = \sigma(i+1)$.

Ideally, in each step of the algorithm an ISS is found so that, when the services in the ISS are removed, the value of the clique bound for the remaining service graph is reduced by one. Such an ISS cannot always be found, but we will show later that steps where the clique bound is not reduced do not occur too often. The special type of ISS constructed by the algorithm is defined below.

Definition 3.3 *Given a service cycle $(G, \mathcal{R})$ and an index i_0 so that $1 \le i_0 \le n$, A* Reducing Independent Service Set (RISS) starting at i_0 *is a pair (I, σ) where $I \subseteq \{1, 2, \ldots, n\}$, and $\sigma : I \to \bigcup_i R_i$ is a function, which satisfy the following properties:*

1. *$i_0 \notin I$.*
2. *For each i so that $i_0 < i \le i_0 + n - 1$:*
   - (a) *if $i - 1 \notin I$ then $i \in I$ and, if $R_i \not\subseteq R_{i-1}$ then $\sigma_i \in R_i - R_{i-1}$, otherwise $\sigma(i) \in R_i$,*
   - (b) *if $i - 1 \in I$ and $\sigma(i-1) \notin R_i$ then $i \notin I$,*
   - (c) *if $i - 1 \in I$ and $\sigma(i-1) \in R_i$ then $i \in I$ and $\sigma(i) = \sigma(i-1)$.*

Note that properties 2 (a), (b) and (c) guarantee that an RISS is indeed an ISS.

Some more definitions are introduced to facilitate the reading of the algorithm and its analysis. Assume a service cycle $(G, \mathcal{R})$ is given. For any index i ($1 \le i \le n$) the *edge value* of i equals $|R_i \cup R_{i+1}|$, and is denoted by $\nu(i)$. If $U = (I, \sigma)$ is an ISS, then $\mathcal{R} - U$ is the service assignment for G obtained by removing the services in U from the service sets: $\mathcal{R} - U = \{R'_i \,|\, 1 \leq i \leq n\}$, where $R'_i = R_i - \{\sigma(i)\}$ if $i \in I$, and $R'_i = R_i$ otherwise. The edge value of i is said to be *reduced* by removing U if $|R'_i \cup R'_{i+1}| < |R_i \cup R_{i+1}|$.

We now describe the Service Colouring for Cycles (SCS) algorithm.

SCS Algorithm

INPUT: A cycle G, and a service assignment $\mathcal{R}$ for G.
OUTPUT: A service colouring f for $(G, \mathcal{R})$.

1. Set $C = 1$, $B = \max_i |R_i \cup R_{i+1}|$.
2. For each service $s \in \bigcap_{i=1}^{n} R_i$:
   (a) For all i, $1 \leq i \leq n$, set $f(v_i, s) = C$,
   (b) Remove s from each of the R_i,
   (c) Set $C := C + 1$, $B := B - 1$.
3. If there exists an index i, $1 \leq i \leq n$, so that $\nu(i) = B$ and $\nu(i-1) < B$ then set $i_0 = i$, else let i_0 be such that $R_{i_0+1} \not\subseteq R_{i_0}$.
4. Construct an RISS U starting at i_0.
5. Set $f(v_i, \sigma(i)) = C$ for all $i \in I$, and set $\mathcal{R} := \mathcal{R} - U$, $C := C + 1$.
6. If $\nu(i_0 - 1) < B$ then set $B := B - 1$.
7. If $R_{i_1} = \emptyset$ for some index i_1 then proceed to Step 8, otherwise return to Step 3.
8. Colour $(G, \mathcal{R})$ greedily, starting at vertex v_{i_1+1}.

The following lemmas will establish the performance ratio and the complexity of the SCS algorithm.

Lemma 3.4 *Let $(G, \mathcal{R})$ be a service cycle and i_0 an index so that $R_{i_0-1} \not\subseteq R_{i_0}$. Let U be an RISS starting at i_0. Let I^* be the set of indices for which the edge value is not reduced by removing U. Then $I^* \subseteq \{i_0 + n - 1\} \cup \{j \mid \nu(j-1) > \nu(j)\}$.*

Proof. Let $(G, \mathcal{R})$, i_0, $U = (I, \sigma)$ and I^* be as stated. For all i, $1 \leq i \leq n$, let R'_i denote the service set of v_i in $\mathcal{R} - U$. From the definition of an RISS it follows that, if $i \neq i_0 + n - 1$ and $i \in I$, then $R_i \cup R_{i+1}$ contains $\sigma(i)$ while $R'_i \cup R'_{i+1}$ does not. So the edge value of index i is reduced by removing R. Likewise, if $i \notin I$ and $R_{i+1} \not\subseteq R_i$, then $R_i \cup R_{i+1}$ contains $\sigma(i+1)$ and $R'_i \cup R'_{i+1}$ does not, so the edge value of i is reduced by removing U.

Let j be an index so that $j \in I^*$ and $j \neq n + i_0 - 1$. Then, by the argument given, $j \notin I$ and $R_{j+1} \subseteq R_j$. Therefore, $\nu(j) = |R_j|$. Also, by the definition of i_0 in the statement of the lemma, $j \neq i_0$. Since $j \notin I$ and $j \neq i_0$, by Properties 2(a) and (c) of an RISS, $j - 1 \in I$ and $\sigma(j-1) \in R_{j-1} - R_j$. Therefore, $\nu(j-1) = |R_{j-1} \cup R_j| > |R_j| = \nu(j)$. □

Lemma 3.5 *Let $(G, \mathcal{R})$ be a service cycle, and let $B = \max_i \nu(i)$ be the clique bound of $(G, \mathcal{R})$. Suppose there exists an index i, $1 \leq i \leq n$, so that $\nu(i) < B$, and let i_0 be as defined in Step 3 of the SCS algorithm. Let U be an RISS starting at i_0. Then the clique bound of $(G, \mathcal{R} - U)$ equals $B - 1$.*

Proof. Let $(G, \mathcal{R})$, B, i_0 and U be as stated. For all i, $1 \leq i \leq n$, let $\nu'(i)$ denote the edge value of index i in $(G, \mathcal{R} - U)$. Since there exists an index i so that $\nu(i) < B$, i_0 is chosen by the SCS algorithm to be such that $\nu(i_0) = B$ and $\nu(i_0 - 1) \leq B - 1$. Therefore, $\nu'(i_0 - 1) \leq B - 1$. For each index j with $i_0 \leq j < i_0 + n - 1$ and $\nu(j) = B$, it holds that $\nu(j) \geq \nu(j-1)$, so by Lemma 3.4, the edge value of j is reduced by removing U, so $\nu'(j) = B - 1$. Therefore, the clique bound of $(G, \mathcal{R} - U)$ equals $B - 1$. □

Definition. Given a service cycle $(G, \mathcal{R})$ and an integer k, a *k-path* in $(G, \mathcal{R})$ is a path $v_i \ldots v_{i+t}$ $(0 \leq t \leq n)$ so that, for all j so that $i \leq j < i + t$, $|R_j \cup R_{j+1}| = k$.

Lemma 3.6 *Let $(G, \mathcal{R})$ be a service cycle which consists of exactly one B-path, and one $(B-1)$-path, where the B-path starts at i_0 and has length ℓ, and $1 \leq \ell \leq n - 1$. Let U be an RISS starting at i_0. Then $(G, \mathcal{R} - U)$*

consists of one $(B-2)$ path and one $(B-1)$ path, and the $(B-1)$-path has length at most $\ell+2$.

Proof. Let $(G,\mathcal{R})$, B, i_0, ℓ and U be as stated. Note that the clique bound of $(G,\mathcal{R})$ equals B. By Lemma 3.4, the only indices whose edge values may not be reduced by removing U are i_0+n-1 and $i_0+\ell$. Both these indices have edge value $B-1$, so their edge value after removing U may be $B-1$ or $B-2$. The edge values of all other indices is reduced, so all indices j with $i_0 \leq j < i_0+\ell$ have an edge value of $B-1$, and all j with $i_0+\ell+1 \leq j \leq i_0+n-2$ have an edge value of $B-2$ after U is removed. So $(G,\mathcal{R}-U)$ consists of a $(B-1)$-path and a $(B-2)$-path, and the $(B-1)$-path has length at most $\ell+2$. $\square$

Lemma 3.7 *In the SCS Algorithm, at the beginning of Step 3, the value of B always represents the clique bound of $(G,\mathcal{R})$*

Proof. In Step 1, B is initialized as $\max_i |R_i \cup R_{i+1}|$, the clique bound of $(G,\mathcal{R})$. It is easy to see that after Step 2, B again represents the clique bound of $(G,\mathcal{R})$. So the statement of the lemma holds for the first time that Step 3 is entered. Suppose then that the statement holds at the beginning of Step 3 in some iteration, and let B and $(G,\mathcal{R})$ be as they are at the beginning of this iteration. We will show that the statement holds again in the next iteration.

If there exists an index so that $\nu(i) < B$, then by Lemma 3.5, and the choice of i_0 in Step 3 and U in Step 4, the clique bound of $(G,\mathcal{R}-U)$ equals $B-1$. In particular, after Step 5, $\nu(i_0+n-1)$ will be smaller than B, so B will be decreased. So after Steps 5 and 6, the statement will hold again.

If $\nu(i) = B$ for every index i, $1 \leq i \leq n$, then by Lemma 3.4, the only index that may not have been reduced is i_0+n-1. So the edge value of i_0+n-1 in $(G,\mathcal{R}-U)$ equals B or $B-1$, and that of all other indices equals $B-1$. So after $\mathcal{R}$ has been replaced by $\mathcal{R}-U$ in Step 5, the clique bound of $(G,\mathcal{R})$ equals the edge value of i_0+n-1. So after Step 6, the statement of the lemma holds again. $\square$

Lemma 3.8 *If, in some iteration of the SCS algorithm, B is not decreased in Step 6, then B is decreased in the next $\lfloor \frac{n}{2} \rfloor$ iterations.*

Proof. Suppose that B is not decreased in some iteration of the SCS algorithm. Let $(G, \mathcal{R})$ be as it is at the beginning of the *next* iteration. From the proof of Lemma 3.7 we know that now $\nu(i) = B - 1$ for all i such that $i_0 \leq i < i_0 + n - 1$, and $\nu(i_0 + n - 1) = B$. So $(G, \mathcal{R})$ consists of a B-path of length 1, and a $(B-1)$-path of length $n-1$. Hence, there is only one choice for i_0 in Step 3: the index of the beginning of the B-path. By Lemma 3.6, and with U as chosen in Step 4, $(G, \mathcal{R} - U)$ will consist of a $(B-1)$-path and a $(B-2)$-path, and the $(B-1)$-path will have length at most three. In Step 6, B will be decreased, so in the following iteration $(G, \mathcal{R})$ will consist of a B-path of length at most three, and a $(B-1)$-path.

Generalizing the argument, it follows that in the $\lfloor \frac{n}{2} \rfloor$ iterations following an iteration where B is not decreased, $(G, \mathcal{R})$ will consist of a B-path and a $(B-1)$-path, and in each iteration the length of the B-path will increase by at most two. Consequently, B will be decreased in each of these iterations. $\square$

If, in Step 7 of the SCS algorithm, $R_{i_1} = \emptyset$ for some index i_1, then the remaining graph is a path, and hence is coloured optimally by the greedy algorithm (see Lemma 3.1). So Step 8 of the SCS algorithm finds a colouring of the remaining graph $(G, \mathcal{R})$ which uses exactly B colours. Hence the following corollary follows directly from the previous lemma.

Corollary 3.9 *The SCS algorithm uses at most* $\left(1 + \frac{1}{\lfloor n/2 \rfloor}\right) \max_i |R_i \cup R_{i+1}|$ *colours for a service colouring of* $(G, \mathcal{R})$.

For an analysis of the complexity of the SCS algorithm, note that the most expensive step in each iteration is Step 4, the construction of an RISS. This step takes $\mathcal{O}(N)$ operations, where $N = \sum_{i=1}^{n} |R_i|$. Hence the complexity of the SCS algorithm applied to a service cycle $(G, \mathcal{R})$ with clique bound B is $\mathcal{O}(B \cdot N) = \mathcal{O}(N^2)$.

This completes the proof of the following theorem.

Theorem 3.10 *The SCS algorithm for finding a service colouring of a service cycle has a performance ratio of at most* $\left(1 + \frac{1}{\lfloor n/2 \rfloor}\right)$, *and a complexity of* $\mathcal{O}(N^2)$, *where* $N = \sum_{i=1}^{n} |R_i|$, *the size of the input.*

Consider the following example. Let G be a cycle of length n, let $A = \{a_1, \ldots, a_n\}$ be a set of services, and $\mathcal{R} = \{R_i \mid 1 \leq i \leq n\}$ be so that $R_i = A - \{a_i, a_{i+1}\}$ for each index i. Then $R_i \cup R_{i+1} = A - \{a_{i+1}\}$, so the clique bound of $(G, \mathcal{R})$ equals $n-1$. Given any choice for i_0, $R_{i_0+1} - R_{i_0} = \{a_{i_0}\}$ So in any RISS $U = (I, \sigma)$ starting at i_0, a_{i_0} is the unique choice for $\sigma(i_0 + 1)$. Moreover, this RISS is itself unique, with $I = \{i_0 + 1, \ldots, i_0 + n - 2\}$, and $\sigma(j) = a_{i_0}$ for all $j \in I$. It can be verified that each RISS constructed in Step 3 of the SCS algorithm applied to $(G, \mathcal{R})$ is unique, and consists of all $n - 2$ occurrences of one particular service a_j. Hence, the SCS algorithm uses $n = (1 + \frac{1}{n-1}) \max_i |R_i \cup R_{i+1}|$ colours.

The above seems to give a lower bound on the performance ratio of the SCS algorithm. This is not the case; the colouring found in the example is, in fact, optimal. Namely, it is easy to check that in this example the maximal size of any ISS equals $n - 2$, so at most $n - 2$ services can receive the same colour. In total, there are $\sum_{i=1}^{n} |R_i| = n(n-2)$ services that occur in $(G, \mathcal{R})$, so at least n colours are needed.

It may be, therefore, that the performance ratio is substantially better than the upper bound given in Theorem 3.10. In fact, we have found no examples where the SCS algorithm asymptotically exceeds the optimal number of colours.

4 Further work

The problem of channel assignment in digital broadcasting networks has emerged only recently, and consequently there are many challenging aspects of this problem yet waiting to be explored. On the theoretical side, a study of the service colouring problem, and of other simplifications of the problem, will give insight into the structure and degree of difficulty of the problem.

For a theoretical exploration of the service colouring problem, we suggest the development of bounds and algorithms for the case where the service graph has a structure that corresponds to that found in a typical real-life network. Suitable graphs would be hexagon graphs (subgraphs of the triangular lattice), planar graphs, unit disk graphs, and circle intersection graphs.

One abstraction of the DB channel assignment problem which is worth exploring is that where all services have size $1/k$ for some integer k. This means that up to k services fit into a channel. The bin packing aspect has now become trivial, but unlike service colouring, this problem cannot be mapped to standard graph colouring. Preliminary work suggests that this problem is quite complex.

On the practical side, efficient heuristics should be developed which can find an acceptable solution to real-life problems, and even take into account additional constraints (such as restrictions on frequency use at certain transmitters).

References

[1] E.G. Coffman, M.R. Garey, and D.S. Johnson. Approximation algorithms for bin packing: a survey. In D. Hochbaum, editor, *Approximation algorithms for NP-hard problems*. PWS publishing, 1997.

[2] S.P. Fekete and J. Schepers. New classes of lower bounds for bin packing problems. *Int. Progr. and Comb. Opt.*, 1412:257–270, 1998.

[3] Michael R. Garey and David S. Johnson. *Computers and Intractability: A guide to the Theory of NP-Completeness*. W.H. Freeman and Co., 1976.

[4] A. Gräf. DAB ensemble planning—problems and techniques. *Telecommunication Systems*, pages 137–154, 2001.

[5] A. Gräf and T. McKenney. Ensemble planning for digital audio broadcasting. In I. Stojmenović, editor, *Handbook of Wireless Networks and Mobile Computing*, pages 267–288. Wiley, 2002.

[6] Ryan B. Hayward. Weakly triangulated graphs. *J. Combin. Th. B*, 39(2):200–208, 1985.

[7] T. McKenney. Eine Anpassung der Tabu Search Methode an das DAB Ensemble-Plannungsproblem. Technical report, Musikinformatik und Medientechnik, Johannes Gutenberg-Universität, Mainz, 2000.

[8] L. Narayanan. Channel assignment and graph multicoloring. In Ivan Stojmenović, editor, *Handbook of Wireless Networks and Mobile Computing*. Wiley, 2002.