Scheduling of a smart antenna: capacitated coloring of unit circular-arc graphs*

Guy Even† Shimon (Moni) Shahar*

July 12, 2006

Abstract

We consider scheduling problems that are motivated by an optimization of the transmission schedule of a smart antenna. In these problems we are given a set of messages and a conflict graph that specifies which messages cannot be transmitted concurrently. In our model the conflict graph is a unit circular-arc graph.

Two variants of the problem are considered: C-MBL and NU-C-MBL. In C-MBL, the messages have unit demands, whereas in NU-C-MBL demands are arbitrary. We present an optimal algorithm for C-MBL and a 3-approximation algorithm for NU-C-MBL.

Keywords: smart antennas, scheduling with conflicts, capacitated coloring, unit circular-arc graphs.

1 Introduction

1.1 Background: scheduling of Smart Antennas

The major issue in wireless networks is how to utilize bandwidth efficiently. This problem is becoming more and more acute as the density and bandwidth requirements of clients increase. The only way to alleviate this bottleneck is to use resources more efficiently. For example, instead of transmitting a message to all direction in the vicinity of an antenna, use a directional beam that concentrates the energy only in the direction of the receiving client. This approach has the following advantages. First, the transmission energy is reduced, so apart from saving energy, health hazards are reduced. Second, interference is reduced as clients do not receive transmissions that are not aimed towards them. This means that reliable communication is possible with weaker signals and more messages can be transmitted simultaneously.

Traditionally, a directional beam is formed by using a reflector. To obtain a pencil shaped beam, a two-dimensional parabolic reflector (i.e. a dish) is used. The beam is directed in different directions by rotating the dish so that it is aimed in the desired direction. This mechanical method of steering the beam is slow and is appropriate for base stations that serve a small and fixed set of clients.

*This work was presented in CAAN 2006 - Third Workshop on Combinatorial and Algorithmic Aspects of Networking, 2 July 2006, Chester, United Kingdom

†School of Electrical Engineering, Tel-Aviv University, {guy, moni}@eng.tau.ac.il

Large and mobile sets of clients are often served by organizing horn shaped antennas side by side. Each horn forms a beam that has the shape of a sector (say, a sector with an angle of 60 degrees). This enables transmission in all directions but has two main drawbacks. First, the gain due to the wide sector is small. Second, separate radio frequency (RF) processing is required for each horn antenna. This incurs an additional cost, especially since high power RF components are expensive.

Advances in technology enable forming directional beams with an array of simple antennas (instead of reflectors). The common signal is multiplied by antenna specific complex weights (i.e., each antenna in the array is associated with a different weight). The multiplication by a complex weight changes the amplitude and the phase of the signal. The signals transmitted from different antennas in the array add in interfering and accumulating patterns to form a directional beam (see left part of Fig. 1). By modifying the complex weights, it is possible to electronically steer the beam from one direction to another. Electronic steering can be performed very quickly since no mechanical movement is required. Moreover, a few directional beams in different directions can be super-positioned (see right part of Fig. 1).

Calculating the required weights needed to form a given beam is a complicated task (see: [15, 13, 8]). One method to overcome this problem is to pre-calculate the coefficients that yields a pencil beam of width 2Δ in any given direction θ, and use superposition to obtain a combination of pencil beams. Due to side-lobes and non-linear effects, the angular difference between super-positioned beams cannot be too small. To simplify the model, we assume that beams may not overlap, namely, the angular difference should be at least Δ. This means that two clients with an angular difference less than Δ may not be served simultaneously.

To summarize, we assume that the antenna is "smart" in the sense that in every time slot, the beam may be the superposition of at most C pencil beams, each of width 2Δ. The angular difference between two pencil beams in the same time slot must be at least Δ.

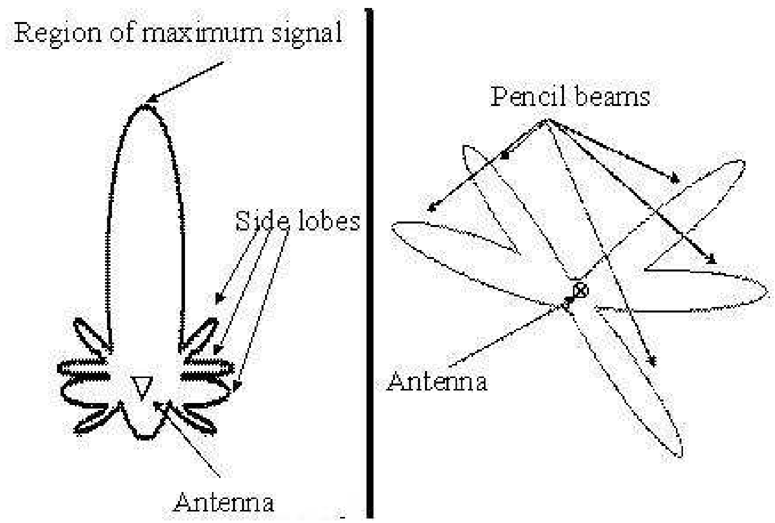

Figure 1: (A) A pencil beam. The contour bounds a region in which the reception level is above a certain level. Note that the main lobe points up. (B) The superposition of 5 pencil beams.

We consider a setting in which there are n clients in the vicinity of the antenna. The goal is to transmit message m_i to client i. We assume that all messages have the same length. The goal is to schedule the transmission of the messages to minimize the amount of time needed to transmit all messages. In each time slot, at most C messages can be sent simultaneously. This is achieved by pointing at most C beams towards the clients whose messages are transmitted in this time slot. Messages scheduled for transmission in the same time slot may not conflict. Namely, the angular difference between the corresponding clients must be at least Δ. We now formalize the problem of scheduling a smart antenna.

1.2 Formalizing the Model

The model in this paper is of a network with one antenna that serves a set of n clients. Both the antenna and the clients are modeled as points in the plane. The location of client i is denoted by its polar coordinate $(r(i), \phi(i))$, where the antenna is positioned in the origin. The antenna should transmit a unique message m_i for each client i. In the sequel, we do not distinguish between the client i and its message m_i, so we simply refer to $(r(i), \phi(i))$ as the location of the message m_i. The assumptions on the transmission model are the following:

- The transmission is done in time slices.
- During a time slice, the antenna transmits the messages of a subset of the clients.
- When transmitting to client i, the antenna aims a pencil beam in its direction $\phi(i)$.
- During a time slice, the antenna's beam is a superposition of multiple pencil beams. The antenna may support at most C pencil beams concurrently, provided that the angular difference between every two pencil beams is at least Δ.

  The angular difference between every two beams is at least Δ. Therefore, two clients i, j for which $|\phi(i) - \phi(j)| \leq \Delta$ are in conflict, and their messages cannot be scheduled concurrently.

Let M be the set of messages, i.e., $M = \{m_i | \; 1 \leq i \leq n\}$. Two messages i, j are *in conflict* if $|\phi(i) - \phi(j)| \leq \Delta$. A *schedule* for the antenna, is a partition of M such that: (i) $M = \bigcup_{i=1}^{k} C_i$, and (ii) every two messages in the same partition C_j are not in conflict. The *length* of the schedule, is simply the number of subsets k. The *Minimum Broadcast Length* problem (MIN-LEN), is the problem whose input is a set of messages with locations, and its output is a schedule of minimum length.

An extension to MIN-LEN problem is obtained by limiting the number of messages than can be transmitted simultaneously. This limit is caused by the specifications of the antenna's array that forms the beam. We refer to this bound as the *capacity.* The capacity C poses a constraint on a schedule, namely, $\forall j \; |C_j| \leq C$. The problem of minimizing the schedule length in the presence of a capacity constraint is called the *C-Minimum Broadcast Length* problem (C-MBL).

Another type of capacity constraint arises in a weighted scenario. Assume that different messages require different transmission power. For example, the power required for broadcasting a message m_i may depend on the distance r_i^2, or on the reception quality of the client(i.e. the client's antenna). Another motivation for this problem is dealing with different noise levels. A natural way to deal with the noise level, is to use code words of different lengths to the clients. Namely, if a client has a low noise level, then the code words that are used for transmission to this client are short; whereas if a client has a high noise level, then the code words that are used for transmission are longer. Therefore we connect the bandwidth demands of a client with his noise level. Since the antenna has a limited bandwidth, the capacity constraints is a limit on the overall bandwidth for transmission in a time slice. However, this leads to a reduction between the problem of dealing with noise level and the problem of allocating transmission power. I.e. the antenna deals with higher noise level by assigning a wider bandwidth. A client with wider bandwidth is assigned a wider range of frequencies, and hence the antenna spends more power on its message. In order to capture a general scenario, we model this problem as follows. For each message m_i, we assign a positive number d_i, that is its demand. For the antenna we assign a number C (also referred to as its capacity), that stands for its maximum power per time slice. Hence we require: $\forall j \ \sum_{i \in C_j} d_i \leq C$. We refer to the problem of minimizing the broadcast length where demands exist as the *Non-Uniform C-Minimum Broadcast Length Problem* (NU-C-MBL).

1.3 Graph Theoretic Formulation

In this section we formulate the scheduling problems presented as graph theoretic problems. We define the *collisions graph* $G = (V, E)$ as follows. The vertex set $V(G)$ contains one vertex v_i for each message m_i, the edge set $E(G)$ contains the edge (v_i, v_j) if and only if the messages m_i, m_j interfere.

Due to the geometric origin of the constraints in our problem, the collisions graph is a unit circular-arc graph. A unit circular-arc graph, is a graph that has a realization as the collision graph of a set of unit length arcs. Namely, there exists a set R of unit length arcs on a circle, and a bijection $f : V \to R$ such that: $(v_i, v_j) \in E$ if and only if $f(v_i) \cap f(v_j) \neq \emptyset$ (the arcs are intersected).

In order to avoid confusion, we would like to emphasize that an arc which represents a message, corresponds to a vertex of the collision graph. The edges of the collision graph, that in some of the literature are referred to as arcs, corresponds to intersections between arcs. Nevertheless, we preferred this choice of terms since it follows the standard terms used for circular-arc graphs.

The arc realization of vertex v_i is simply the sector of size Δ centered at $\phi(i)$, i.e., the sector $[\phi(i) - \Delta/2, \phi(i) + \Delta/2]$. Since this is a projection of the clients locations on the unit circle, two arcs $f(v_i), f(v_j)$ intersect if and only if their corresponding messages m_i, m_j interfere.

The MIN-LEN problem, can be reduced to the problem of computing a minimum coloring the graph G. Given a minimum coloring, it assigns to each arc m_i a color from the set $\{1, \ldots, \chi^*\}$, where χ^* denotes the chromatic number of the graph. This coloring defines the schedule in which the clients scheduled to time slice t are all the clients whose arcs are colored t. Due to the requirement that the messages in each time slice are independent sets, the length of this schedule is minimum.

The C-MBL and NU-C-MBL are capacitated versions of the coloring problem. E.g., in C-MBL we need to partition the $V(G)$ into minimum number of independent sets $\{C_j\}$, such that the size $|C_j|$ of each set does not exceed the antenna's capacity.

1.4 Previous Results

Two problems related to C-MBL and NU-C-MBL are coloring of circular-arc graphs and bin-packing with conflicts. We briefly review previous works on these problems.

Previous results on coloring circular-arc graphs. Circular-arc graphs are a natural generalization of intervals graphs since the removal of any maximal clique from a circular-arc graph yields an intervals graph. (This is due to the fact that the set of arcs passing for a given point on the circle forms a subset of a maximal clique.) Since intervals graphs are perfect, finding their chromatic number χ^* is a well known polynomial problem (see for example [14]). However, despite the resemblance, finding the chromatic number of circular-arc graphs is NP-Hard ([2]). The coloring problem remains NP-Hard even for special families of circular-arc graphs ([3]). Based on this connection between intervals graphs and circular-arc graphs, the following simple algorithm is a 2-approximation for coloring circular-arc graphs: (i) Color a maximal clique, (ii) remove it from the graph so as to obtain an intervals graph, and (iii) color optimally the intervals graph.

A different 2-approximation algorithm for coloring circular-arc graph was given by Tucker ([11]). His algorithm is a greedy, first-fit algorithm that has two stages: (i) first generate an order on the arcs, and (ii) assign consecutive blocks of arcs according to the same color class, according to the order generated. Tucker also conjectured that $\frac{3\omega}{2}$ colors are always sufficient to color a circular-arc graph, where ω is the size of the maximum clique. This conjecture was proved by Karapetian ([6]). Shih and Hsu gave a $\frac{5}{3}$ approximation algorithm ([10]). Their algorithm is based on extending Tucker's ideas to cases where no three arcs cover the circle. Recently Valencia-Pabon showed that if the minimum size of a dominating set is ℓ, then the approximation ratio of Tucker's algorithm is $\frac{(\ell-1)}{(\ell-2)}$ ([12]).

An approximation algorithm based on linear programming for coloring circular-arc graphs was given by Kumar ([7]). His algorithm achieves an approximation ratio of $(1 + 1/e + o(1))$ for graphs whose coloring number χ^* is $\Omega(\log n)$. His algorithm first reduces the coloring problem into a multi-commodity flow problem, then it solves the corresponding linear program, and finally it rounds up the solution.

An important family of circular-arc graphs is proper circular-arc graphs. A proper circular-arc graph is a circular-arc graph for which there exists a realization R such that no arc is fully contained within another. An even more restricted family of graphs are unit circular-arc graphs, for which there exists a realization such that all arcs lengths are the same, trivially such a realization is also proper. Orlin, Bonuccelli and Bovet ([9]) showed that coloring proper circular-arc graphs is polynomial. The running time of their algorithm is $O(n^2 \log n)$, and it is based on a procedure that checks whether a graph is k-colorable, for a given value of k. We refer to this algorithm as COL-PROPER(G).

Corollary 1 *By the reduction given in the previous section, we obtain that* COL-PROPER(G) *optimally solves the* MIN-LEN *problem.*

An important balancing property of COL-PROPER(G), is that the size of all the color classes it generates is either $\left\lceil \frac{n}{\chi^*} \right\rceil$ or $\left\lfloor \frac{n}{\chi^*} \right\rfloor$, thus the color classes have the same cardinality up to rounding.

Previous results on bin-packing with conflicts. The input to a bin-packing problem with conflicts is a set V of elements with weights between zero and one. In addition, the input contains a conflict graph $G = (V, E)$ whose vertex set is the set of elements. An edge $(u, v) \in E$ indicates that u and v cannot be assigned to the same bin. The goal is to find a packing of V into as few bins as possible, where the size of each bin is one. The NU-C-MBL problem is a bin-packing problem with conflicts where the conflict graph is a unit circular-arc graph.

Jansen [5] considered bin-packing with conflicts where an optimal coloring of the conflict graph is given (e.g., coloring is polynomially solvable for a class of graphs that contains the conflict graph). For such cases, an r-approximation is presented, where $2.69 \le r \le 2.7$. In addition, a 2.5-approximation algorithm for graphs that can be precolored is presented[1]. The approximation algorithms are based on an analysis of the first-fit-decreasing algorithm for bin packing after small elements are reweighted and matched with big elements.

An asymptotic approximation scheme for bin-packing with conflicts was presented by Jansen [4] for d-inductive graphs[2].

Epstein & Levin [1] present improvements for bin-packing with conflicts. The improvements in [1] over [5] are due to a better choice of weights assigned to small elements and a tighter analysis. The contributions in [1] are as follows: (i) A 5/2-approximation algorithm for conflict graphs that can be optimally colored in polynomial time. This implies a 5/2-approximation algorithm for NU-C-MBL. (ii) A 7/3-approximation algorithm for conflict graphs that can be optimally precolored. (iii) A 7/4-approximation algorithm for bipartite conflict graphs. (iv) An online algorithm with a 4.7-competitive ratio for conflict graphs that are interval graphs.

1.5 Our Results

In this paper we present the following results:

- An optimal algorithm for C-MBL, whose running time is $O(n^2 \log n)$.
- An $O(n \log n)$ heuristic for C-MBL based on Tucker's coloring algorithm. We bound the approximation ratio of this heuristic by 3. This heuristic is optimal under a condition that seems to hold in practice.
- NP-Hardness of NU-C-MBL.
- A very simple 3-approximation algorithm for NU-C-MBL(see [1] for better approximation).

2 The Capacitated Minimum Broadcast Length Problem

In this section we present two results. The first result is an optimal algorithm for C-MBL, to which we refer by OPT-CMBL. The second result is a heuristic, based on Tucker's algorithm, which is optimal under some assumptions that may hold in practical cases. We think this heuristic is interesting since it is simple, and its running time is $O(n \log n)$, whereas the running time of OPT-CMBL is $O(n^2 \log n)$.

2.1 An optimal algorithm for c-mbl

Let C denote the antenna's capacity and n the number of messages. Let G denote the collision graph (G is a unit circular-arc graph). Let χ^* be the chromatic number of G. We denote by ℓ^* the length of an optimal schedule. In this section we prove the following theorem; the proof is separated into a few claims.

Theorem 2 $\ell^* = \max\{\chi^*, \lceil \frac{n}{C} \rceil\}$. *Moreover, an optimal cover can be computed in polynomial time.*

[1] The precoloring extension problem is defined as follows: Given a graph $G = (V, E)$ and a subset of vertices $V' = v_1, ..., v_k$. Find a minimum coloring of of G satisfying $f(v_i) = i$ for all $i = 1, .., k$.

[2] A graph is d-inductive, if there exists an ordering of the vertices such that the "left" degree of every vertex is at most d.

We begin by proving the lower bound.

Claim 3 $\ell^* \geq \max\{\chi^*, \lceil \frac{n}{C} \rceil\}$.

Proof: The first bound follows since the messages transmitted concurrently in every time slice are independent. The second bound follows simply from the fact that the number of messages transmitted in any time slice is bounded by C. ■

The following corollary shows that if the color classes are small, then the optimal coloring induces an optimal schedule.

Corollary 4 *If* $\left\lceil \frac{n}{\chi^*} \right\rceil \leq C$ *then* COL-PROPER(G) *induces an optimal schedule.*

Proof: The balancing property of COL-PROPER(G) implies that the largest color class in the coloring computed by COL-PROPER(G) contains $\left\lceil \frac{n}{\chi^*} \right\rceil$ clients. Since $\left\lceil \frac{n}{\chi^*} \right\rceil \leq C$ the coloring induces a valid schedule. Since $\ell^* \geq \chi^*$, optimality follows. ■

We now deal with the case of large color classes.

Claim 5 *If* $\left\lceil \frac{n}{\chi^*} \right\rceil \geq C$ *then* $\ell^* = \lceil \frac{n}{C} \rceil$.

Proof: We first reduce the graph G so that the sizes of the color classes computed by COL-PROPER(G) are not divisible by C. The reduction proceeds as follows: (i) Partition the arcs into color classes using COL-PROPER(G). (ii) If the size of a color class C_i is divisible by C, schedule the messages in C_i in groups of size C. Denote the number of arcs assigned in this step by n'', and let $n' = n - n''$. If $n' = 0$, then we are done, and the theorem holds. Let G' denote the collision graph induced by the n' remaining nodes. Obviously, $\ell^*(G) \leq \ell^*(G') + \frac{n''}{C} = \lceil \frac{n}{C} \rceil$. It suffices to prove that $\ell^*(G') = \left\lceil \frac{n'}{C} \right\rceil$. We may therefore assume that the size of every color class computed by COL-PROPER(G) is not divisible by C. Namely, neither $\left\lceil \frac{n'}{\chi} \right\rceil$ or $\left\lfloor \frac{n'}{\chi} \right\rfloor$ is divisible by C. As a result of this reduction, we assume that the sizes of the color classes in the coloring computed by COL-PROPER(G) are not divisible by C and that $C < \left\lfloor \frac{n}{\chi^*} \right\rfloor$ (i.e., every color class contains more than C arcs).

We use the following terminology. If S is an independent set and $|S| = k \cdot C + r$. Then S can be partitioned to C-*groups* and a *leftover-group*. The C-groups are k disjoint subsets of C_i (each of size C), and the leftover group is a subset of C_i of size r.

A listing of Algorithm SCHEDULE appears as Algorithm 1. The algorithm computes an optimal balanced coloring of the collision graph G using COL-PROPER(G). The idea is to schedule a C-group from each color class. The issue of the leftover group is solved by finding a C-group in the union of the current color class and the next color class (see the Replacement Claim below) to schedule C additional arcs. Since all slots, but perhaps the last slot, contain C arcs, Algorithm SCHEDULE computes a schedule whose length is $\lceil \frac{n}{C} \rceil$. Hence, we only need to prove the Replacement Claim. ■

Claim 6 (Replacement Claim) *In Line 8 of Algorithm* SCHEDULE *there always exists a set* $S \subset C_i \cup C_{i+1}$ *such that: (i)* $|S| = C$*, and (ii) both* S *and* $(C_i \cup C_{i+1}) \setminus S$ *are independent sets.*

Algorithm 1 Algorithm SCHEDULE - scheduling when $\left\lfloor \frac{n}{\chi^*} \right\rfloor > C$.

1: Run COL-PROPER(G).
2: **for** $i = 1$ to $\chi^* - 1$ **do**
3:   **while** $|C_i| \geq C$ **do**
4:     Schedule a C-group $T \subseteq C_i$.
5:     $C_i = C_i \setminus T$.
6:   **end while**
7:   **if** $C_i \neq \emptyset$ **then**
8:     Find an independent set $S \subset (C_i \cup C_{i+1})$ of size C such that $(C_i \cup C_{i+1}) \setminus S$ is also independent. {*Existence guaranteed by the Replacement Claim*}
9:     Schedule S
10:     $C_{i+1} = (C_{i+1} \cup C_i) \setminus S$
11:   **end if**
12: **end for**
13: Schedule the C-groups and leftover group contained in C_{χ^*}.

Proof: Consider the sets C_i and C_{i+1} in Line 8. Note that the set C_i is not the original color class C_i that was generated by COL-PROPER(G), since its size is less than C. We have $0 < |C_i| < C < |C_{i+1}|$. The graph induced by C_i and C_{i+1} is 2-colorable unit circular-arc graph. Hence, the union $C_i \cup C_{i+1}$ is a cycle or a union of disjoint simple paths in G. If $C_i \cup C_{i+1}$ is a cycle, it should contain the same number of edges from C_i and from C_{i+1}, contradicting the fact that $|C_i| < C < |C_{i+1}|$. Hence, $C_i \cup C_{i+1}$ is a union of disjoint simple paths in G. In each path the difference between the number of vertices from each color classes is zero or ± 1.

We apply the following iterative "balancing" procedure to obtain S. The procedure partitions $C_i \cup C_{i+1}$ into S_1 and S_2. Initially, $S_1 = C_i$ and $S_2 = C_{i+1}$, so $|S_1| < C < |S_2|$. In each iteration, arcs are moved between S_1 and S_2 so that $|S_1|$ increases by one and $|S_2|$ decreases by one. The invariant that S_1 and S_2 are a partition of $C_i \cup C_{i+1}$ into independent sets is always kept. The balancing procedure stops as soon as the size of one of the sets S_1 or S_2 is C.

Consider an iteration of the procedure in which $|S_1| < |S_2|$. The moving of arcs proceeds as follows. Pick an odd path P in which $|S_2 \cap P| > |S_1 \cap P|$. We update the sets S_1, S_2 as follows: $S_1 = (S_1 \setminus P) \cup (P \cap S_2)$, $S_2 = (S_2 \setminus P) \cup (P \cap S_1)$. ■

We remark that the proof of Claim 6 implies an efficient algorithm to find the required set S, hence Algorithm SCHEDULE is polynomial.

In the appendix we present an efficient scheduling algorithm for the case that $\left\lfloor \frac{n}{\chi^*} \right\rfloor > 2C$. This algorithm captures the intuition that if the color classes are large, then a leftover can be augmented by arcs from the next color class to fill a time slot.

2.2 A Practical Heuristic: Spiral Scheduling

The most time consuming part of the algorithm for C-MBL is the coloring of the graph by procedure COL-PROPER(G). In this section we present an algorithm called SPIRAL that avoids the coloring step. SPIRAL is based on Tucker's algorithm for coloring circular-arc graphs. Thanks to the unit length of the arcs, the implementation of SPIRAL is easier than the implementation of Tucker's algorithm.

SPIRAL has three major steps: ordering the arcs, coloring, and scheduling. We begin by describing the ordering step. Consider a representation of a circular-arc graph using a circle and arcs of the circle. We define the *load* of a point P on the circle to be the number of arcs that contain P. We order the arcs in spiral ordering as follows: (i) Set A_1 to be an arc whose starting point has the biggest load. (ii) Initially, all arcs but A_1 are unmarked. For every i between 1 and $n-1$, set A_{i+1} to be the unmarked arc whose starting point is the closest to the end of A_i. We refer to an order generated by the sorting as a spiral ordering.

We now describe the coloring step. Given a spiral ordering $A_1, \ldots, A_n$ of the arcs, the arcs are colored by iteratively "peeling" a maximal prefix of arcs that is an independent set. Each such maximal prefix is assigned a new color. In other words, initially $\mathcal{A}$ is the set of all arcs. In iteration i, pick a maximal prefix P_i of $\mathcal{A}$ that is independent. Color P_i with the color i, and update $\mathcal{A} \leftarrow \mathcal{A} \setminus P_i$. This procedure is repeated until $\mathcal{A}$ is empty.

In the scheduling step each color class is simply partitioned into C-groups and perhaps a leftover group.

In the case of unbounded capacity, this algorithm simply reduces to Tucker's algorithm. Therefore, even for the unbounded capacity case the algorithm is not optimal. We prove that SPIRAL is a constant ratio approximation algorithm.

Claim 7 *The approximation ratio of* SPIRAL *is no bigger than* 3.

Proof: Let $\chi_T(G)$ denote the number of colors used in the coloring step. Let $\ell_s(G)$ denote the length of the schedule computed SPIRAL. We claim that

$$\ell_s(G) \leq \frac{n}{C} + \chi_T(G).$$

Indeed, there are at most $\frac{n}{C}$ slots with C arcs and there is at most one leftover group from each color class.

The claim now follows from Theorem 2 and from the fact that $\chi_T(G) \leq 2 \cdot \chi^*$ (this was proved by Tucker [11]). ■

Suppose that subsets of C consecutive arcs in the spiral ordering are independent. In this case, after spiral ordering we could peel off a prefix of C arcs in each step to obtain an optimal schedule. Hence, a variation of SPIRAL leads to an optimal schedule if this condition holds.

Consider the following two conditions: (i) $C \cdot \Delta < \pi/2$ and (ii) The number of clients is proportional to the angle of a section, provided that the angle is large (say, $\pi/4$). Under such conditions most C consecutive arcs are independent, and the actual approximation ratio of the algorithm is much better than the theoretical bound. We believe that these two conditions hold in practice.

The running time of SPIRAL is dominated by the time to compute the spiral ordering which is $\theta(n \log n)$. This is more efficient than the optimal algorithm whose running time is $\theta(n^2 \log n)$.

3 The Non-Uniform Capacitated Minimum Broadcast Length Problem

In this section we deal with non-uniform demands, i.e., the NU-C-MBL problem. We prove that NU-C-MBL is NP-Hard, and present a simple 3-approximation algorithm for it.

Claim 8 *The* NU-C-MBL *problem is NP-Hard.*

Proof: We reduce bin-packing to NU-C-MBL. Consider a set $\{1, \ldots, n\}$ of elements, where the size of element i is $s_i \in [0, 1]$. Reduce this bin-packing instance to an NU-C-MBL instance by constructing a set of n pairwise disjoint arcs, where the demand d_i of the ith arc is s_i. Finally, set the antenna's capacity to be $C = 1$. Since there are no coloring constraints on this graph, every packing induces a valid schedule, and vice-versa. ■

We present a 3-approximation algorithm for NU-C-MBL called NCMBL-APP. The algorithm NCMBL-APP starts with an optimal coloring calculated by COL-PROPER(G). Next the algorithm schedules the messages in each color class C_i independently as follows: (i) Sort the messages in C_i according to their demands in non-increasing order. (ii) Partition the C_i into time slots by "peeling" off maximal prefixes with demand at most C. The algorithm could try to add messages to fill gaps in previous time slices, but the analysis does not take this into account.

The correctness of NCMBL-APP follows from the fact that each color class is an independent set. The following claim bounds the approximation ratio of the NCMBL-APP.

Claim 9 *The approximation ratio of* NCMBL-APP *is at most* 3.

Proof: Consider the schedule computed by NCMBL-APP. Note that every time slot, except perhaps for the last time slot in each color class, is at least half full. Indeed, let j be a time slice that is not the last time slice of the color class C_i. Let D_j be the sum of demands of all the messages that was scheduled to time slice j. Let d' be the demand of the first message that was scheduled to time slice $j+1$. Note that there is a message in time slice j whose demand is at least d', since the messages are sorted in non-increasing demand order. Therefore if $d' \geq C/2$, then $D_j \geq C/2$. If $d' < C/2$, then because this message was not assigned to time slice j we get that: $D_j > C - d' > C - C/2 = C/2$.

It follows that there is at most one time slot per color class that is not half full. Let ℓ denote the length of the schedule computed by NCMBL-APP. We obtain $\ell \leq \left\lceil \frac{n}{C/2} \right\rceil + \chi^* \leq 3 \cdot \ell^*$, and the claim follows. ■

4 Discussion and open problems

There are many open problems regarding smart antennas, and we outline here just a few. First, the assumption of unit arc-length may not hold for more complicated cases. Such cases may occur either due to the antenna's abilities or due to environmental conditions. Therefore, both capacitated models should be generalized to arbitrary circular-arc graphs.

We note that an approximation algorithm for capacitated coloring follows from an approximate coloring algorithm. Let ℓ^* denote the length of an optimal schedule and χ^* the chromatic number of the collision graph. By partitioning each color class into slots of size C and a possible leftover, it follows that $\ell^* \leq \lceil \frac{n}{C} \rceil + \chi^*$ (see Claim 9). This implies also that a ρ-approximation for the chromatic number, leads to $(1 + \rho)$-approximation for capacitated coloring (with unit demands). In the case of non-unit demands, a ρ-approximation algorithm for coloring, leads to a $(2 + \rho)$-approximation algorithm.

The transmission model used in this paper is simple since we ignored physical effects such as reflections and side lobes. In order to take into account such effects, the model should be extended as follows. (i) Each message m_i defines a set of disjoint arcs. (ii) Two messages m_i, m_j can be transmitted concurrently only if every pair of arcs $a_i \in m_i$, $a_j \in m_j$ do not intersect (or if the amount of overlapping is limited). The collision graph that corresponds to this problem is more complicated, and finding an optimal schedule is an interesting open question even for the case where the number of arcs per message is constant.

Finally, improving the approximation factor for NU-C-MBL is interesting, since such an algorithm may be useful for more complicated models.

Acknowledgments

We thank Tony Weiss for interesting conversations about smart antennas. We thank the anonymous reviewers for the helpful remarks. Guy Even was partly funded by REMON, Israel 4G mobile consortium.

References

[1] L. Epstein and A. Levin. On bin packing with conflicts. *Manuscript*, 2006.

[2] M. R. Garey, D. S. Johnson, G. L. Miller, and C. H. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM Journal on Algebraic and Discrete Methods*, 1(2):216–227, June 1980.

[3] F. Gavril. Intersection graphs of helly families of subtrees. *Discrete Appl. Math.*, 66(1):45–56, 1996.

[4] K. Jansen. An approximation scheme for bin packing with conflicts. *J. Comb. Optim*, 3(4):363–377, 1999.

[5] K. Jansen and S. Öhring. Approximation algorithms for time constrained scheduling. *Information and Computation*, 132(2):85–108, 1 Feb. 1997.

[6] I. Karapetian. On coloring arc graphs. *Dokladi (Reports) of the Academy of Science of the Armenian Soviet Socialist Republic*, 70:306–311, 1980.

[7] V. Kumar. An approximation algorithm for circular arc colouring. *ALGRTHMICA: Algorithmica*, 30, 2001.

[8] M. Mouhamadou, P. Armand, P. Vaudon, and M. Rammal. Interference supression of the linear antenna arrays controlled by phase with use of sqp algorithm. *PIER: Progress In Electromagnetics Research*, 59:251–265, 2006.

[9] J. B. Orlin, M. A. Bonuccelli, and D. P. Bovet. An $O(n^2)$ algorithm for coloring proper circular arc graphs. *SIAM Journal on Algebraic and Discrete Methods*, 2(2):88–93, June 1981.

[10] W. K. Shih and W. L. Hsu. An approximation algorithm for coloring circular-arc graphs. In *SIAM Conference on Discrete Mathematics*, 1990.

[11] A. Tucker. Coloring a family of circular arcs. *SIAM Journal on Applied Mathematics*, 29(3):493–502, Nov. 1975.

[12] Valencia-Pabon. Revisiting tucker's algorithm to color circular arc graphs. *SICOMP: SIAM Journal on Computing*, 32, 2003.

[13] P. K. Varlamos and C. N. Capsalis. Electronic beam steering using switched parasitic smart antenna arrays. *PIER: Progress In Electromagnetics Research*, 36:101–119, 2002.

[14] D. B. West. *Introduction to Graph Theory (2nd Edition).* (Prenctice Hall, Upper Saddle River), 2001.

[15] Z. Zhang, M. Iskander, Z. Yun, and A. Host-Madsen. Hybrid smart antenna system using directional elements - performance analysis in flat rayleigh fading. *IEEE Transactions on Antennas and Propagation*, 51(10):2926–2935, 2003.

A A scheduling algorithm for big color classes

In this section we present an optimal scheduling algorithm for the case that $\left\lfloor \frac{n}{\chi^*} \right\rfloor > 2C$.

Definition 10 *Let C_i, C_j be two disjoint independent sets in a unit circular-arc graph. The sets $S_i \subseteq C_i$ and $S_j \subseteq C_j$ C-*agree *if: (i) $S_i \cup S_j$ is an independent set, and (ii) $|S_i \cup S_j| = C$.*

We now present an algorithm called BIG to handle the "big classes" case. We first compute an optimal balanced coloring by calling COL-PROPER(G). The algorithm BIG proceeds with $\chi^* - 1$ iterations. In the first iteration it does the following. (i) Partition the color class C_1 into C-groups, and a leftover group S_1. (ii) Schedule the C-groups. (iii) Find an independent set $S_2 \subseteq C_2$ that C-agrees with S_1. The existence of S_2 is proved in Claim 11. Moreover, finding S_2 is easy: simply take a a subset of S_2 that are not neighbors of S_1. A listing of BIG appears as Algorithm 2. Since all slots, but perhaps the last slot, contain C arcs, Algorithm BIG computes a schedule whose length is $\lceil \frac{n}{C} \rceil$. Hence, we only need to prove Claim 11.

Algorithm 2 Algorithm BIG - handles the "big classes" case

Run COL-PROPER(G).
for $i = 1$ to $\chi^* - 1$ **do**
  Schedule the C-groups in what remains of C_i {*Scheduled messages are removed from C_i*}
  if $C_i \neq \emptyset$ **then**
    Find a set $S_{i+1} \subset C_{i+1}$ that C-agrees with C_i {*see Claim 11*}
    Schedule $S_{i+1} \cup C_i$, and remove S_{i+1} from C_{i+1}
  end if
end for
Schedule the C-groups and the leftover group in what remains of C_{χ^*}.

Claim 11 *There is always a set $S_{i+1} \subset C_{i+1}$ that C-agrees with the leftover subset of C_i.*

Proof: Let S' and S'' denote two disjoint independent sets of arcs in a unit circular-arc graph G. We first show that if S'' is contained in the set $\Gamma(S')$ of neighbors of S', then $|S''| \leq 2 \cdot |S'|$. Assume otherwise, then there exists an arc $u \in S'$ that intersects at least 3 arcs $v_1, v_2, v_3 \in S''$. The arcs

v_1, v_2, v_3 are independent. Therefore u must strictly contain one of these arcs (i.e., the middle arc). However, this contradicts the fact that all arcs are of unit length.

Consider the subset that remains of the color class C_i in Line 4. Let r denote the size of this leftover. By definition, $r < C$. By the previous paragraph, it follows that the number of arcs in C_{i+1} that intersect arcs in S_i is at most $2r$. Since $|C_{i+1}| - 2r > 2C - 2r > C - r$, we conclude that $|C_{i+1} \setminus \Gamma(S_i)| > C - r$. Hence, there are enough arcs in $C_{i+1} \setminus \Gamma(S_i)$ to be added to S_{i+1}. ■

We remark that the proof of the claim induces a linear time algorithm to find the required set S_{i+1}; simply add the first $C - t$ arcs that do not intersect S_i.