Online Channel Allocation in FDMA Networks with Reuse Constraints

Tomás Feder* Sunil M. Shende†

Abstract

The topology of an FDMA network is modeled by a subdivision of a planar region into hexagonal cells; this defines a graph where every node has at most six neighbors. For channel allocation, every node has a weight that indicates the number of channels that must be assigned to it; the reuse distance defines the minimum distance between nodes that can use the same channel without radio interference. It is known that with reuse distance 2, the number of channels needed and their allocation can be approximated within a factor of 4/3 both offline and online. We describe an online algorithm for the case of reuse distance 3 that achieves an approximation factor of 7/3.

KEYWORDS: **Analysis of Algorithms, Combinatorial Problems, Design of Algorithms, Distributed Computing**

1 Introduction

Wireless networking based on Frequency Division Multiple Access (FDMA) technology is widely deployed in cellular telephony, networks for mobile computing and similar communications systems. A typical FDMA cellular network services a geographic area by dividing it into cellular regions or *cells*. Each cell contains a *base station* equipped with radio transceivers that allow the base station to maintain communication with active mobile clients (*e.g.* a cell-phone user) within the cell; the base station also communicates with other base stations in its geographic vicinity.

Any client that needs to make a *call* over the system is serviced upstream by its nearest base station; conversely, if the recipient of the call is also a mobile client, the base station closest to it manages the downstream wireless link. We will henceforth assume that a call is synonymous with a mobile client (whether upstream or downstream) that requires service. In both cases, the appropriate base station must assign to the call, a specific radio frequency from a communication channel (*i.e.* a frequency band) from a fixed and limited spectrum of frequencies; for instance, the Advanced Mobile Phone Service (AMPS) system for cellular telephony uses a total of 832 specified channels in the 800-900 MHz band. Naturally, when spatially proximate calls are assigned frequencies in spectrally promixate channels, this may cause *radio interference*: the objective, therefore, is to design a channel allocation protocol that tries to maximize the efficient utilization of the available spectrum while minimizing interference.

*Email: tomas@theory.stanford.edu

†Dept. of Computer Science, Rutgers University, Camden, NJ 08102; Email: `shende@crab.rutgers.edu`; FAX: (609) 225-6624. This work was completed during Feb.1998 while the author was visiting the Dept. of Computer Science at Stanford University.

Cellular networks are often modeled by dividing the planar calling area into mutually disjoint but identical *hexagonal* cells that cover the entire area [8]. Two cells are adjacent if they share a common boundary edge in this hexagonal tiling of the plane. In general, the distance between two cells is the smallest number of boundary edges crossed going from one cell to the other. Interference constraints are usually modeled in terms of a *reuse distance* $L \geq 2$; it is assumed that a channel can be *reused* without causing interference in two cells if and only if the distance between the cells is at least L.

Note that if prior assignments render it impossible to assign a channel to a new call, the call is *blocked* and may or may not be dropped depending on the specific admission control mechanism being employed. Our concern is not with the latter mechanism: we are only interested in analyzing protocols which attempt to allocate as much of the available frequency spectrum as possible before the need to block a call arises. Hence, we focus on a particular measure of performance, *viz.* the *guaranteed performance ratio* of a protocol. We assume that the spectrum is divided into some number of discrete channel. A protocol is defined to have a performance ratio of M or is said to be M-optimal if the following holds: whenever the proposed protocol blocks on a call, we can prove that at that instant, *every other protocol* (including an unknown but optimal one) must have used *at least* a fraction $1/M$ of the available spectrum[1].

There is a vast amount of literature that deals with the problem of efficient channel assignment in FDMA-based wireless networks. It is easy to construct protocols that rely on reservation-based strategies (*i.e.*, fixed allocation) to obtain performance ratios of 3 (for reuse distance 2) and 7 (for reuse distance 3). Many other proposed heuristics (see, *e.g.*, [1, 6, 7, 11, 12]) claim to achieve better performance in practice, but have no proven bounds on their ratio of optimality. Indeed, it was only recently shown [9] that even when the reuse distance is limited to 2 and the number of calls requiring service is known *a priori*, it is NP-hard to determine the optimal number of required channels. Thus, it is unlikely that any computationally feasible protocol, whether offline or online, can be shown to be perfectly optimal. For reuse distance 2, protocols that achieve guaranteed performance were first described by Janssen et.al [3, 4]. More recently, algorithms with a performance ratio of 4/3 have been described both in the offline [9, 10] as well as the online setting [5]; it is conjectured that this bound is optimal. In the present paper, we consider a more constrained model, *viz.* with reuse distance 3. For this model, we describe a simple algorithm that has a guaranteed performance ratio of 7/3. This improves significantly upon the currently best known ratio of 7 for reuse distance 3.

2 Preliminaries

Consider a tiling of the plane with identical hexagons, and its planar dual, the triangular grid. Let $G = (V, E)$ be a finite undirected subgraph of the triangular grid (induced by the subset of vertices V) and let $w(\cdot)$ be a *weight* function that assigns non-negative integer weights to the nodes of G. Let H denote the graph $G \cup G^2$, *i.e.* H has the node set $V(G)$ and edge set

$$E(G) \ \cup \ \{(u,v) : \ \exists x \notin \{u,v\} \text{ s.t. } (u,x),(x,v) \in E(G)\}.$$

We refer to the graphs G and H as the *triangle* and the *square* graph respectively. The channel allocation problem with reuse distance 3 is equivalent to the problem of multicoloring the square graph H so that (a) each node $u \in V(H)$ is assigned a set of $w(u)$ distinct colors, (b) any pair of adjacent nodes u, v in H are assigned *disjoint sets* of colors, and (c) the total number of colors used in the

[1]Note that our definition accounts for the worst scenario; it is entirely possible that when an M-optimal protocol blocks, a perfectly optimal protocol may have already used up considerably more than a fraction of $1/M$ of the spectrum.

graph H is minimized. To see the correspondence, note that the weight of a node is the number of calls requiring service at that node, the assigned colors are the allocated channels, and the same color (channel) may only be reused in nodes of G at distances greater than 2 or equivalently, at nodes of H at distances greater than 1.

Let $\chi(H)$ denote the *weighted chromatic number* of H, *i.e.* the minimum number of colors needed to color H so that conditions (a) and (b) above are satisfied. It is straightforward to extend the NP-hardness result for computing the chromatic number of triangle graphs [9] to square graphs as well. Hence, the best that one can do is to devise some M-optimal protocol for coloring square graphs, where M is a small constant. Let the weight of any clique in H be defined as the sum of the weights of the nodes forming that clique. The *weighted clique number* of H, denoted $W(H)$, is defined to be the maximum over the weights of all cliques in H. Clearly, $W(H)$ is a lower bound on $\chi(H)$. Since every allocation protocol (including an optimal one) will need at least $W(H)$ or more colors to properly color H, hence an algorithm that approximates the number of colors used to within a factor of M of the weighted clique number yields an M-optimal protocol.

3 Notation

Consider the triangle graph G (without node weights) and fix a particular node v in the graph. Refer to Figure 1(a) that depicts v as the central node in the figure. In the square graph H, the node v is adjacent to each of the eighteen other nodes shown in the figure. In particular, we call the six nodes in the hexagon surrounding v as the *inner* nodes with respect to v; the remaining twelve nodes are the *outer* nodes. The figure thus shows H_v, the subgraph of H induced by v and all its inner and outer neighbors.

For convenience, we adopt the following naming convention to identify these neighbors: the convention relies on the fact that any clique in H has at most 7 nodes and that the underlying unweighted graph corresponding to H (*i.e.* with all nodes having unit weight) can be colored with 7 colors. We fix a *base coloring* of H with 7 colors as follows. The node v has color 1 (as indicated by its subscript in the figure), and for every color $k \in \{2, 3, 4, 5, 6, 7\}$, v has an inner neighbor called x_k and two outer neighbors called y_k and z_k as indicated in the figure. Note that the base coloring of H is in fact, a proper 7-coloring of the unweighted graph underlying H, *i.e.* when all nodes of H have unit weights. This can be seen by observing that the pattern of colors in the hexagon consisting of the inner nodes can be replicated by translation: *e.g.* by translating the inner hexagon so that the marked edge (x_5, x_6) is aligned with the marked edge (z_5, y_6) etc. In general, every inner hexagon is surrounded by six such identically colored hexagons.

Observe that in the circular ordering of the outer nodes implicit in Figure 1(a), any group of three consecutive outer nodes along with v belong to a unique clique of H_v that also contains v. For example, the notation $\langle y_4, z_3, y_5\rangle$ describes the 6-clique $\{v_1, x_6, x_7, y_4, z_3, y_5\}$, whereas $\langle z_7, y_3, z_2\rangle$ describes the 7-clique $\{v_1, x_4, x_5, x_6, z_7, y_3, z_2\}$. In particular, Figure 1(b) illustrates how H_v can be covered with four such cliques obtained by partitioning the outer nodes into groups of three consecutive nodes, *viz.* $\langle y_4, z_3, y_5\rangle$, $\langle z_4, y_6, z_5\rangle$, $\langle y_7, z_6, y_2\rangle$, and $\langle z_7, y_3, z_2\rangle$. Clique covers such as these play a prominent role in the subsequent discussion.

For instance, if our only concern is to compute some *offline* coloring of H, then a conceptually easy idea yields a 2-optimal algorithm for the problem. Let W is the weighted clique number of H, and assume that we have $2W$ distinct *hues*[2] available for coloring. Consider the embedding of the (possibly

[2] We use the terminology to avoid confusion with base colors.

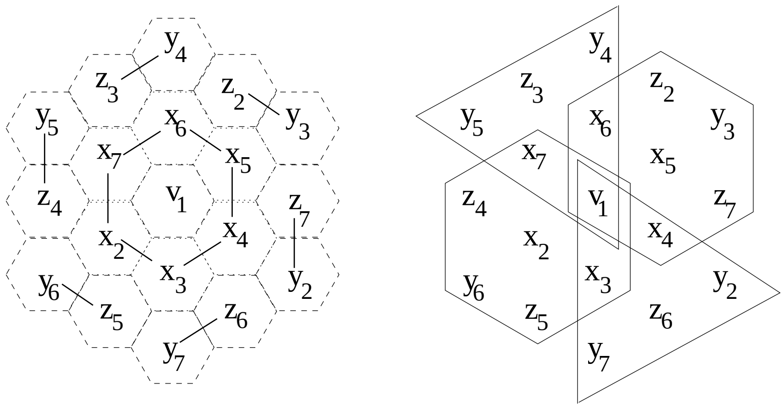

Figure 1: (a) The base coloring, (b) A clique cover of H_v

non-planar) square graph H on the planar triangular grid. Since H is a finite graph, we can always construct the convex hull of H. Suppose that v as shown in Figure 1(a) is a vertex of the convex hull. Then it is intuitively easy to visualize that the nodes with positive weight in H_v lie only on one *side* of v. In particular, it means that the nodes with positive weight in H_v can be covered with at most two cliques like those shown in Figure 1(b). Since the total weight in the two cliques is less than $2W$, it follows that if every node in H_v except v is already colored, then v has enough hues left over to color itself. Hence, the algorithm constructs a *total ordering* of the nodes by repeating the following steps until a single node remains. It computes the convex hull of the current graph, chooses a hull vertex, adds it to the ordering and deletes it from the graph. Now the nodes are simply assigned hues greedily in *reverse order* of the constructed ordering. From the discussion above, it becomes clear that a node will never run out of available hues.

However, the algorithm outlined above is not suitable for our application since we would like to compute the hues needed at a node without any offline computation. In fact, it is desirable to design a protocol in which nodes can be thought of as servers that *independently* color themselves based on their local neighborhood. In the main section of this paper that is presented next, we derive a simple greedy algorithm that can be implemented so that it runs *online* and in a *distributed* manner.

4 The Coloring Algorithm

Our algorithm for assigning colors (channels) to the nodes of a square graph is quite straightforward. For now, we assume that the square graph H is fixed and let W denote the weighted clique number of H. We associate with each base color a total of l distinct hues, where l is some fraction of W to be fixed later: thus, there are a total of $7l$ colors available in the spectrum. A node is called *heavy* if its weight exceeds the threshold l.

In the beginning, each node v with weight $w(v)$ is assigned the first $\min\{w(v),\ l\}$ hues of the color of the node; hence, only the heavy nodes remain to be completely colored. For the base colors $1, 2, \ldots, 7$ in order, we now examine, one by one, all the heavy nodes of the considered color. Each such node v is completely colored in a *greedy* manner by assigning any currently unused hues within the subgraph H_v. For this strategy to work correctly, we must be able to guarantee that at most $7l$ hues are ever used among v and its neighbors in H_v; otherwise, v could potentially run out of available hues. We first show that this holds when $l = 2W/5$, thus implying a guaranteed performance ratio of $14/5 = 2.8$ for the algorithm. The argument is fairly simple. We then give a more refined argument which shows that in fact, $l = W/3$ suffices; this establishes a stronger guarantee of $7/3 \approx 2.333$ on the performance ratio of the greedy algorithm.

Theorem 1 *If $l = 2W/5$, then for any heavy node v, the total weight over all nodes in the subgraph H_v does not exceed $7l$.*

Proof. Without loss of generality, let v be a node with base color 1 and weight $w(v) > l$. As shown in Figure 1(b), the subgraph H_v can be covered with four cliques each with total weight at most W. The total weight of H_v is at most $w(v) + 4(W - w(v)) = 4W - 3w(v) < 4W - 3l = 7l$ as required. $\square$

To obtain a tighter bound on the number of hues actually required by the greedy algorithm, we note the following invariant. If two nodes in v's neighborhood have the same base color k, then they will both *reuse the same initial segment* of the l hues of color k. For example, suppose that $w(x_k) \geq w(y_k) \geq w(z_k)$ for the three base color k neighbors of v. If $w(z_k) \leq l$, then the initial segment of $w(z_k)$ hues of color k is also used at the nodes y_k and x_k. Hence, we may as well assume that z_k's color demand is charged to x_k. Likewise, if $w(y_k) \leq l$, then the initial segment of $w(y_k)$ hues of color k is also used at the node x_k: thus, y_k's color demand may be charged to x_k. Finally, if either of y_k or z_k have weights exceeding l, then all the l hues of base color k are used at the nodes x_k, y_k and z_k; thus, l units of demand may be charged to x_k. This observation prompts the following definition.

Definition 1 *Consider a fixed heavy node v surrounded by its inner and outer neighbors (see Figure 1(a)). For a given base color k, let $U_k \in \{x_k, y_k, z_k\}$ be the neighbor of v with maximum weight among all three color k neighbors (assume that ties are broken arbitrarily). The* demand, *$d_v(\cdot)$, at a node $u \in \{x_k, y_k, z_k\}$ with respect to v is defined as follows:*

$$d_v(u) \quad = \quad \begin{cases} w(u) & \text{if } u \equiv U_k \\ w(u) - l & \text{if } u \not\equiv U_k, \text{ and } u \text{ is heavy} \\ 0 & \text{otherwise.} \end{cases}$$

Finally, $d_v(v)$, the demand at v with respect to itself, is defined to be $w(v)$.

It is easy to see that the total demand with respect to v over all nodes in the subgraph H_v is an upper bound on the number of hues that would be actually used in H_v by the greedy algorithm. Hence, to ensure that v does not run out of available hues, it suffices to establish the following result:

Theorem 2 *If $l = W/3$, then for any heavy node v, the total demand with respect to v over all nodes in H_v does not exceed $7l$. Hence, the greedy coloring algorithm is a $7/3$-optimal algorithm for the reuse distance 3 allocation problem.*

Proof. As before, we assume that v is a heavy node with base color 1 and focus our attention on the subgraph H_v. Since all demands considered are with respect to v, we may drop the subscript on the demand function without any ambiguity. Clearly, v has positive demand; let $S \subseteq (H_v - \{v\})$ denote the collection of all other nodes in H_v with positive demand. We first make some simple observations about the members of S. All the heavy nodes of base color $k \neq 1$ are members of S. Alternatively, if all three nodes of some color $k \neq 1$ are non-heavy, then S contains at most one node of color k.

Further, since $l = W/3$, the subset of heavy nodes in S must form an independent set of vertices in the graph H_v; otherwise, contrary to assumption, a pair of such adjacent nodes along with v would form a 3-clique in H_v with weight exceeding W. From the geometry of H_v, it follows that S contains at most 4 heavy nodes if all the heavy nodes are from among v's outer neighbors, and at most 3 heavy nodes otherwise.

Therefore, S conforms to one of four possible cases: it contains (1) at most one node of each distinct color, except for two pairs of heavy nodes corresponding to some base colors $1 \neq k_1 \neq k_2 \neq 1$, (2) at most one node of each distinct color, except for a triple of heavy nodes corresponding to all nodes of some base color $k \neq 1$, (3) at most one node of each distinct color, or (4) at most one node of each distinct color, except for a pair of heavy nodes of some base color $k \neq 1$.

Cases (1) and (2): Consider the four cliques that cover H_v in Figure 1(b). We know that in both cases under consideration, S contains exactly two heavy nodes whose demands are l less than their respective weights. Hence the total demand in the subgraph H_v is given by:

$$\begin{aligned} d(H_v) &\leq w(H_v) - 2l \\ &\leq d(v) + 4(W - d(v)) - 2l \\ &< 4W - 3l - 2l \\ &= 7l. \end{aligned}$$

We next prove cases (3) and (4). We make frequent references to the configuration of nodes in H_v shown in Figure 1 so the reader is urged to refer to the figure as necessary in the subsequent discussion. To simplify the analysis, we restrict ourselves to choosing only certain kinds of cliques in the remainder of the proof: in particular, we only consider cliques formed from three consecutive outer nodes like those ones shown in Figure 1(b). Note that every inner node is adjacent to one such group of three consecutive outer nodes. As a purely technical device, we can imagine that the entire positive demand on any inner node has been *moved* to a suitably chosen adjacent outer node. Any one of the three outer neighbors can be chosen to be the recipient of such a move. Let S' denote the subset of outer nodes with positive demands after all demands from inner nodes are moved out appropriately. Clearly, S' contains at most as many nodes as does S. In any event, a node has non-zero demand before (resp. after) demands are moved from the inner nodes to the outer ones if and only if it belongs to S (resp. S').

Case (3): Recall that S contains at most one node of each distinct color (other than the color 1 of v). If three consecutive outer nodes are not in S', then the remaining nine outer nodes can be partitioned into three groups of three consecutive outer nodes. Each group determines a clique and together, these three cliques suffice to cover H_v; the total demand is then at most $d(v) + 3(W - d(v)) \leq 3W - 2l = 7l$.

Otherwise, consider the case when at least one among any two consecutive outer nodes is in S'. It follows that exactly six alternating outer nodes have positive demand, *i.e.* either all (and only) the nodes y_k or all (and only) the nodes z_k, $2 \leq k \leq 7$ belong to S'. In the former situation, three cliques *viz.* $\langle y_2,\ z_7,\ y_3 \rangle$, $\langle y_4,\ z_3,\ y_5 \rangle$, and $\langle y_6,\ z_5,\ y_7 \rangle$ account for all the demand in H_v, and the net demand is at most $7l$ as argued in the preceding paragraph. The latter situation can be disposed of via a similar argument.

This leaves the only remaining possibility where there is a pair of two consecutive outer nodes not in S', but at least one among every three consecutive outer nodes does belong to S'. The two consecutive outer nodes absent in S' could either belong to the same outer hexagon, *e.g.* the nodes y_3, z_2 or to two different outer hexagons, *e.g.* the nodes z_7, y_3.

Consider first the case when $z_2, y_3 \notin S'$. Clearly, y_4 is in S' because no three consecutive outer nodes have zero demand after the move. Further, we can assume that y_4 is also in S, because any demand on x_6 could have been moved to z_3 instead of to y_4. This implies that $z_4 \notin S$ because S contains at most one node of any color. Moreover, we can assume that $z_4 \notin S'$, since demands, if any, in x_2 and x_7 can as well be moved to nodes other than z_4. Hence, z_2, y_3 and z_4 are absent from S' and hence, the cliques $\langle y_4, z_3, y_5 \rangle$, $\langle y_6, z_5, y_7 \rangle$ and $\langle z_6, y_2, z_7 \rangle$ account for the demand in H_v. As before, a bound of $7l$ on total demand follows.

Consider next the case when $z_7, y_3 \notin S'$. The nodes y_2 and z_2 must both belong to S' to avoid three consecutive outer nodes being absent from S'. Further, we can assume that $y_2 \in S$, because any demand on x_4 can be moved to z_6 instead of y_2. Therefore, $z_2 \notin S$ because S contains at most one node of any color. Hence, the positive demand on z_2 after the move, must have originated at one or both of the inner nodes x_5 or x_6. In fact, we can assume that the demand on x_5 is positive before the move (*i.e.* $x_5 \in S$) and is moved to z_2 because while any demand on x_6 can be moved to outer nodes other than z_2, the demand on x_5 can only have been moved to z_2 (since its other two outer neighbors, *viz.* z_7 and y_3, are both absent from S'. So, since $x_5 \in S$, it must hold that both $y_5, z_5 \notin S$; in particular, we can again assume that $y_5 \notin S'$ because the demand, if any, on x_7 can be moved to outer nodes other than y_5. Hence, the nodes $z_7, y_3, y_5 \notin S'$, and the cliques $\langle z_2, y_4, z_3 \rangle$, $\langle z_4, y_6, z_5 \rangle$ and $\langle y_7, z_6, y_2 \rangle$ can be used to cover the demand in H_v. A bound of $7l$ on total demand again follows.

Case (4): Recall that S contains at most one node of each distinct color, except for a pair of heavy nodes of some base color $k \neq 1$. Among these two monochromatic nodes, exactly one has a demand that is l less than its weight, while the demands of the other nodes in S equal their respective weights. Hence, showing that the total demand in H_v is at most $7l$ is equivalent to showing that the *total weight* of all the nodes in S is at most $8l$. Consider the six marked pairs of outer nodes that belong to the six hexagons, *e.g.* the pair y_3, z_2. We claim that among these pairs there is at least one, such that (a) both nodes in the pair are absent from S, or that (b) exactly one of the two nodes is in S, and it is a non-heavy node. Suppose instead that neither condition holds for each of the six pairs. Since S contains at most seven nodes (counting the two nodes of base color k), there must be at least five pairs, each containing exactly one heavy node in S. But this is impossible, since H_v contains at most four heavy nodes other than v.

So, without loss of generality, let z_2, y_3 be a pair that satisfies condition (a) or (b) above. We can further assume that neither of these nodes gets any additional demand after moving demands, if any, from the inner nodes (x_5's demand can be moved to z_7 while x_6's demand can be moved to either y_4 or z_3); if only one among z_2 and y_3 is a member of S and has weight (demand) at most l, then the same holds in S' as well. Hence, our goal is to show that ignoring the nodes z_2 and y_3, the remaining nodes that belong to S' only contribute a total *weight* of at most $7l$.

Consider the situation now after inner weights have been moved out subject to the assumption stated in the previous paragraph. If $y_4 \notin S$, then we may assume that y_4 is not in S' as well because x_6's demand, if any, can be moved to z_3. The three cliques, *viz.* $\langle z_3, y_5, z_4 \rangle$, $\langle y_6, z_5, y_7 \rangle$, and $\langle z_6, y_2, z_7 \rangle$ now cover the remainder of S'; the bound of $7l$ on total weight follows. A similar argument works if $z_4 \notin S'$.

Otherwise, suppose that both y_4 and z_4 belong to S, *i.e.* for every other base color $1 \neq k \neq 4$, there is at most one node of color k in S. Furthermore, if $y_7 \notin S$, then we can assume that y_7 is not in S' as well since any demand on x_3 can be moved to z_5 and z_6 instead. Now the three cliques, $\langle y_5, z_3, y_4 \rangle$, $\langle z_4, y_6, z_5 \rangle$, and $\langle z_6, y_2, z_7 \rangle$, cover the remaining weight which can be at most $7l$. On the other hand, if $y_7 \in S$, then z_7 cannot belong to S holds because there can be at most one node of color 7 in S. If z_7 has zero demand after any moves, *i.e.* $z_7 \notin S'$ then the remaining weight in H_v can be covered with the three cliques, $\langle y_5, z_3, y_4\rangle$, $\langle z_4, y_6, z_5\rangle$, and $\langle y_7, z_6, y_2\rangle$. Otherwise, if $z_7 \in S'$ but is not in S, then it must be that some demand on x_5 has been moved to z_7. It follows that x_5 is in S and therefore, that y_5 and z_5 are absent from S. The remaining weight in H_v can now be covered with three cliques, $\langle z_3, y_4, z_2\rangle$, $\langle y_2, z_6, y_7\rangle$, and $\langle z_4, y_6, z_5\rangle$. This completes the proof.

□

4.1 Online Distributed Implementation

It is easy to see that our algorithm can be easily converted to a distributed, online one when N, the total number of available colors in the spectrum is known. Thus, the algorithm will block if and only if the total weight of any clique in the network exceeds $3N/7$. We briefly sketch the implementation below.

Each node has knowledge of the current weights of its neighbors at distance two or less in the grid. Further, every node has a prescribed base color k, $1 \leq k \leq 7$ in some fixed 7-coloring of the grid, and each of the seven base colors consist of $N/7$ hues. During successive time instants, the weights on nodes may change and can be serviced over two stages as follows. In the first stage, each node will try to use as many hues of its base color. The next stage proceeds over seven phases: during phase k, every node of base color k simultaneously emulates the greedy borrowing strategy in its neighborhood.

A small detail is worthy of mention. In the context of channel allocation, each call has a specific identity that gets blurred when the call is treated just as a single unit of weight in the coloring formulation of the problem. What happens between steps when the weight at a node goes up or down? The new coloring recomputed by the algorithm would correspond to channels being allocated anew to calls, even those that may have survived from the previous step. In practice however, it is not difficult to see that the recoloring can be implemented so that only the hues borrowed by a node from other base color hues could possibly lead to any changes in allocated channels.

5 Conclusions

We have shown that a simple greedy algorithm with performance ratio 7/3 can assign channels (colors) in any FDMA-based network without interference, subject to the constraint that channels can be reused at distances greater than or equal to 3 in the triangular grid. The algorithm can be implemented to run online in a distributed manner at each node. Reusing channels beyond a given distance is an idealization of interference constraints. A more refined model [2] prescribes a minimum *separation* between channels assigned to two cells, where the separation is a non-increasing function of the distance between the cells. An interesting open question is to design protocols that provide guaranteed performance ratios in the context of such a separation constraint model.

Acknowledgments

The second author would like to thank Rajeev Motwani for graciously hosting his visit to the Computer Science department at Stanford University during February 1998.

References

[1] D. Dimitrijević and J. Vučetić. Design and performance analysis of algorithms for channel allocation in cellular networks. *IEEE Transactions on Vehicular Technology*, 42(4):526–534, 1993.

[2] W. K. Hale. Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 68(12):1497–1514, 1980.

[3] J. Janssen and K. Kilakos. Optimal multicolouring algorithms with limited recolouring. Submitted for publication, April 1995.

[4] J. Janssen, K. Kilakos, and O. Marcotte. Fixed preference frequency allocation for cellular telephone systems. Unpublished manuscript, April 1995.

[5] J. Janssen, D. Krizanc, L. Narayanan, and S. M. Shende. Distributed on-line frequency assignment in cellular networks. In *15th Symposium on Theoretical Aspects of Computer Science*, Feb 1998. to appear in a forthcoming LNCS (Springer-Verlag) volume.

[6] T. Kahwa and N. Georganas. A hybrid channel assignment scheme in large-scale cellular-structured mobile communication systems. *IEEE Transactions on Communications*, 4:432–438, 1978.

[7] S. Kim and S. L. Kim. A two-phase algorithm for frequency assignment in cellular mobile systems. *IEEE Transactions on Vehicular Technology*, 1994.

[8] V. H. MacDonald. The cellular concept. *The Bell System Technical Journal*, 58:15–41, 1979.

[9] C. McDiarmid and B. Reed. Channel assignment and weighted colouring. Submitted for publication, 1997.

[10] L. Narayanan and S. M. Shende. Static frequency assignment in cellular networks. In *Proceedings of the 4th International Colloquium on Structure of Information and Communication Complexity*, 1997. in press.

[11] P. Raymond. Performance analysis of cellular networks. *IEEE Transactions on Communications*, 39(12):1787–1793, 1991.

[12] W. Wang and C. Rushforth. An adaptive local-search algorithm for the channel-assignment problem. Technical Report, August 1995.