A unified framework and algorithm for channel assignment in wireless networks

S. Ramanathan

BBN Technologies Division, GTE Internetworking, Cambridge, MA 02138, USA

Channel assignment problems in the time, frequency and code domains have thus far been studied separately. Exploiting the similarity of *constraints* that characterize assignments within and across these domains, we introduce the first unified framework for the study of assignment problems. Our framework identifies eleven atomic constraints underlying most current and potential assignment problems, and characterizes a problem as a combination of these constraints. Based on this framework, we present a unified algorithm for efficient (T/F/C)DMA channel assignments to network nodes or to inter-nodal links in a (multihop) wireless network. The algorithm is parametrized to allow for tradeoff-selectable use as three different variants called RAND, MNF, and PMNF. We provide comprehensive theoretical analysis characterizing the worst-case performance of our algorithm for several classes of problems. In particular, we show that the assignments produced by the PMNF variant are proportional to the *thickness* of the network. For most typical multihop networks, the thickness can be bounded by a small constant, and hence this represents a significant theoretical result. We also experimentally study the relative performance of the variants for one node and one link assignment problem. We observe that the PMNF variant performs the best, and that a large percentage of unidirectional links is detrimental to the performance in general.

1. Introduction

While the rapidly growing area of wireless networking promises to bring ubiquitous mobile services into the everyday realm, it is also likely to place an extremely high premium on the communications spectrum. The scarcity of spectrum necessitates efficient channel assignment mechanisms. Whether the channel sharing is based upon Time Division Multiple Access (TDMA), Frequency Division Multiple access (FDMA), Code Division Multiple Access (CDMA), or a combination thereof, there exists a fundamental limit on the number of users sharing the same channel simultaneously. This has motivated the need for *spatial reuse* of the channel, that is, having users sufficiently far apart use the same frequency band, time slot, or code. Attesting to the importance of the issue, the literature is replete with considerations of several *specific* assignment problems [1–8,10,11,14,15,17,21–24,27–31,33].

In this paper, we postulate the *generalized* problem of channel assignment in the context of achieving efficient spatial reuse. The generalization stems primarily from two observations: that whatever the access technology, the solution depends largely on the nature of the *constraints*[1] to which an assignment must adhere; and that most (Time/Frequency/Code) Division Multiple Access assignments are characterized by a combination of a handful of underlying constraints. In this generalized formulation, the channel in question could be time, frequency, or code, and the assignment of channels may be to *nodes* or to *links* between nodes in the network. We develop a unified framework that identifies a set of eleven constraints as *atomic*, that is, as building blocks for the generation of *constraint sets* that characterize an assignment problem. Based on this framework, we present a unified algorithm and analysis for channel assignment that is applicable to 144 well-known and potential assignment problems.

While our results are applicable to *single-hop* (e.g., cell-based) as well as *multihop* wireless networks, our focus is on multihop networks, as it represents the most general and challenging manifestation of the problem. A multihop wireless network is one in which a packet may have to traverse multiple wireless links in order to reach its destination. The *Packet Radio Networks* (PRNs) as used in the Defense Advanced Research Projects Agency's PRNET [18,19], SURAN [20,32], and, more recently, the Global Mobility (GloMo) programs are multihop wireless networks. The *ad hoc networks*, a term adopted by the IEEE 802.1 subcommittee [9], that are appearing in the literature (e.g., [26]), are also multihop wireless networks. The terms packet radio networks, ad hoc networks, and multihop wireless networks are conceptually identical.[2]

1.1. Previous work

Much of the prior work in the spatial reuse of channel assignment in multihop wireless networks may be classified based on the technology – FDMA, TDMA or CDMA – that they cater to. Frequency assignment has been well studied, mostly in the context of cellular networks [4,15,22].

[1] An example of a constraint is: two nodes that are adjacent to a common node must not use the same code. We shall discuss this in detail in section 2.2.

[2] Unfortunately, neither "packet radio" nor "ad hoc" bring out the most important difference from conventional cell-based networks – namely, the multihop nature of the wireless communications. For this reason, and to emphasize the generality of our results (e.g., we are not restricted to *radio*-based networks), we have used the term "multihop wireless networks".

In [6,10,30], the TDMA scheduling of broadcasts is considered, while in [7,11,14], TDMA link scheduling is considered. An investigation into the complexity of the scheduling problem is given in [2,1]. Distributed scheduling algorithms are the subject of [8,6,21,30], and the work of [5,6,27] addresses re-scheduling when the network topology is dynamic. With regard to CDMA, [3] study the complexity of the problem, and [3,17,23] propose code assignment algorithms. Distributed algorithms are given in [3,17].

Many assignment problems in this area have been shown to be NP-complete, including TDMA broadcast scheduling [10,12,30], link scheduling [2,12] (even when the graphs are planar [28]), FDMA frequency assignment [15], and CDMA code assignment [3]. Indeed, for some of these problems, even constant times optimum polynomial algorithms appear highly unlikely (i.e., unless P = NP) [28]. However, these results are applicable to arbitrary networks, and thus the question arises whether multihop wireless networks might be modeled by *restricted* graphs. Among the restrictions proposed are trees [1], planar graphs [29], disc graphs [15], and planar point graphs [31]. However, trees are too restrictive, planar graphs are a reasonable model only when the transmission range is quite small, disc and planar point graphs are only valid when there are no obstacles in the signal path (e.g., a building) – an unreasonable assumption for all but a few real-life network environments. In this paper, we do not restrict the nature of the topology, but provide solutions that perform in proportion to deviation from planarity (details in section 1.2).

The work presented in this paper differs significantly from the previous works mentioned above. Unlike most previous work, we use a unified framework and a generalization based on the constraints, which in turn results in the first algorithmic and analytical unification within and across (T/F/C)DMA domains. In particular, our algorithm allows for the constraints as well as the topology to be part of the input parameters, and each of our performance analysis results are valid for not just one but a *class* of problems. Also, most of the prior work assumed links to be bidirectional, whereas we do not. The next section describes and motivates these and other contributions in greater detail.

1.2. Our contributions: motivation and overview

We use the traditional graph coloring problem as a convenient equivalent for channel assignment. However, our graphs are *directed*, and can thus model networks containing unidirectional links. Communicating equipment forming a wireless link may differ in terms of transmission power levels, multipath interference experienced, and noise or jamming experienced. Thus, the assumption that links are bidirectional, made in several previous works (e.g., [3,5–7,10,17,30]) is unrealistic. The generalization to directed graphs, however, is non-trivial as far as problem complexity is concerned, as will be evident from the analysis in section 4.

Using this graph-theoretic model, we develop in section 2 a unified framework for the postulation of assignment problems in time, frequency, and code based networks. Based on the element to be colored (vertex or edge), the forbidden element separation (e.g., two vertices cannot be adjacent, cannot be "distance-2", etc.) and the direction of the constraint (transmitter or receiver), we define 7 edge-based and 4 node-based constraints underlying most assignment problems (described in detail in section 2.2). An assignment problem, whether it be in the time, frequency, or code domain, is then defined in terms of the set of constraints characterizing it. A total of 128 possible link assignment and 16 possible node assignment problems are captured by this framework. From among these, we pick as examples eight problems that have a "real-life" counterpart in TDMA, FDMA and CDMA based networks, and map them into their corresponding constraint sets.

In the context of this unified framework, we present an algorithm for coloring a graph, which translates into a channel assignment for the corresponding network. The algorithm takes as input the topology graph and the set of constraints characterizing the problem. We study three greedy heuristics, each using a different ordering in the way the vertices are considered for coloring. Such an unification is not only of pedagogical interest, but has useful practical implications as well. For instance, it obviates the need to come up with a new algorithm for each new assignment problem that the ever-changing wireless hardware and environment might present. One simply needs to extract the constraints governing the problem within our framework (if subsumed), and use our algorithm with the constraint set as a parameter.

We present, in section 4, analysis characterizing the worst-case performance of our algorithm on a wide variety of assignment problems. We show that, for most problem classes, a novel "progressive minimum neighbors first" (PMNF) ordering on the vertices produces significantly better worst-case and in-practice performance than the traditional random (RAND) ordering. Specifically, the PMNF ordering results in colorings that are guaranteed to be within $O(\theta)$ of the optimum colorings for many problems, where θ is the *thickness* of the topology graph. Thickness is a measure of "nearness-to-planarity" and is the minimum number of planar graphs into which a given graph can be partitioned. Algorithms in most previous works were only able to guarantee performance bounds of $O(\rho)$ where ρ is the maximum graph degree (see, for example, [6,7,10,17,30,33]). Our study shows that in typical multihop wireless networks, the thickness is several orders of magnitude less than ρ. Figures 1(a) and (b) show the growth in thickness[3] compared to growth in maximum degree for increasing network size and transmission range, respectively. Clearly, solutions with guarantees proportional to network thickness are more scalable than those based

[3] Since determining the exact thickness of a graph is itself NP-complete [24], we have used an upper bound based on tree decompositions [25].

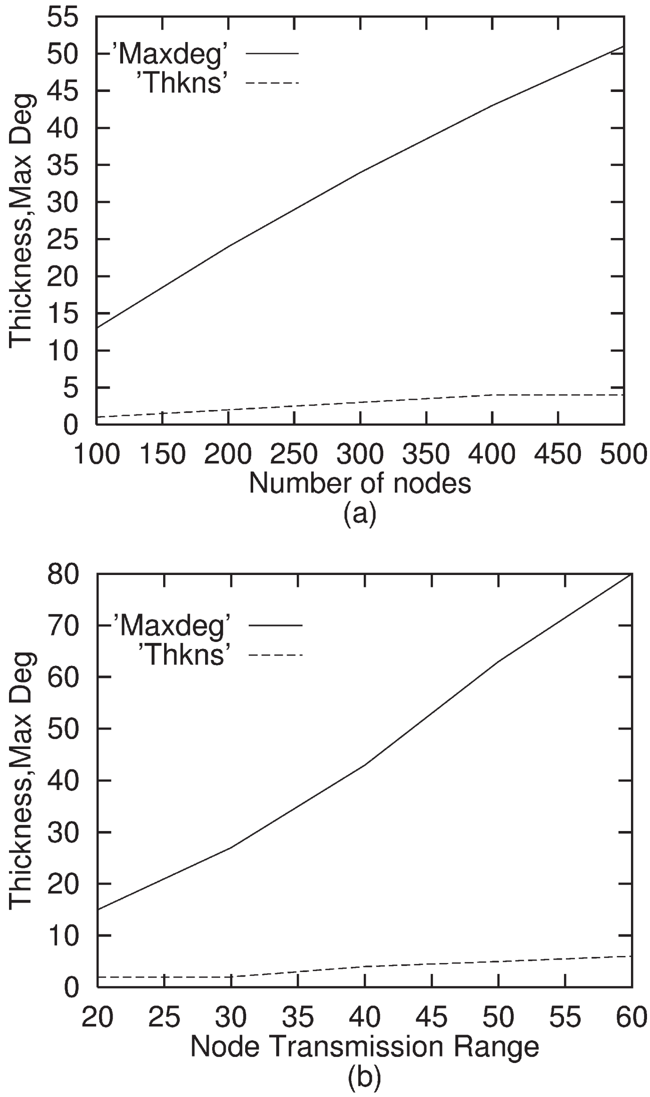

Figure 1. Max. degree and thickness versus (a) number of nodes, with each node having a range of 40 units, and (b) range of a node, with 400 nodes. Experiments conducted by placing nodes randomly on a 400×400 unit grid (details in section 5).

on maximum degree. Our algorithm does not require the calculation of thickness itself in its operation and hence is of polynomial time complexity.

Note that we do *not* consider a restricted class of graphs as a model, and thus, unlike [15,31], are not confined to obstacle-free networks. Nonetheless, our $O(\theta)$ solution implies an O(1) (constant-times-optimum performance guarantee) solution for most real-life networks because θ can usually be bounded by a small number, as indicated by figures 1(a) and (b).

In section 5, we present experimental results characterizing the in-practice performance of our algorithm on a large number of randomly generated multihop network topologies. The performance (number of slots/frequencies/codes used) is studied as a function of the number of nodes, the range of each node, and the fraction of unidirectional links (introduced by having unequal transmission ranges, see section 5 for details). We observe that, for the problems studied, PMNF improves upon RAND by about 8–13%, and provides an in-practice guarantee of 1.1 times optimum.

2. A unified framework for assignment problems

In this section, we describe our network model, terminology, and concepts, and develop a unified framework within which our channel assignment algorithm can be conveniently described and analyzed.

2.1. Graphical representation

Our presentation is based on a standard representation of a (multihop) wireless network as a directed graph $G = (V, A)$. Here, V is a set of *vertices* denoting the nodes comprising the wireless network and A is a set of *directed edges* between vertices representing inter-node wireless links. For any two distinct vertices $u, v \in$ V, the edge from u to v, denoted by $(u \to v)$, is in A if and only if v can receive u's transmission. Note that we do not assume that the edges of the graph are bidirectional. That is, $(u \to v) \in A$ does not necessarily imply that $(v \to u) \in A$.

Assignment of channels to nodes corresponds in a natural fashion to *coloring* the corresponding graph. Depending on the nature of the assignment problem, it is either the vertices or the edges of the graph that receive colors. The mapping is one-to-one, that is, two nodes/links are assigned different channels if and only if the corresponding vertices/edges have different colors.

Formal definitions of terms we use throughout this paper are as follows. Two vertices are *adjacent* if there is an edge from one to the other, and two edges are adjacent if they have a common vertex. If $(v \to w)$ is an edge, then v is an *in-neighbor* of w and w is an *out-neighbor* of v. A *neighbor* is a vertex that is an in-neighbor or an out-neighbor.

The *in-degree* of a vertex v is the number of in- neighbors of v and the *out-degree* of v is the number of out-neighbors of v. The *total degree* (or simply *degree*) of a vertex is the sum of its in- and out-degrees. The *maximum in/out/total-degree* of a graph is the maximum of the (in/out/total) degrees taken over all vertices of the graph and will be denoted by $\rho_{\text{in}}/\rho_{\text{out}}/\rho$. The *maximum degree ratio* is a measure of the asymmetry of the graph and is defined as $\delta_{\text{r}} = \max(\rho_{\text{out}}/\rho_{\text{in}}, \rho_{\text{in}}/\rho_{\text{out}})$.

Planarity of graphs and related notions are of particular interest in this paper. A *planar* graph is one that can be embedded in the plane such that no two edges intersect. The *thickness* of a graph G is the minimum number of planar subgraphs of G whose union is G [16], and will be denoted by θ.

With respect to algorithmic concepts, we use terminology from [13]. In particular, if D_Π is a set of instances of a minimization problem Π, $S_\Pi(I)$ denotes the set of candidate solutions for a specific instance $I \in D_\Pi$, then an *approximation algorithm* for Π is one which, given any instance $I \in D_\Pi$, finds a candidate solution $s \in S_\Pi(I)$. The

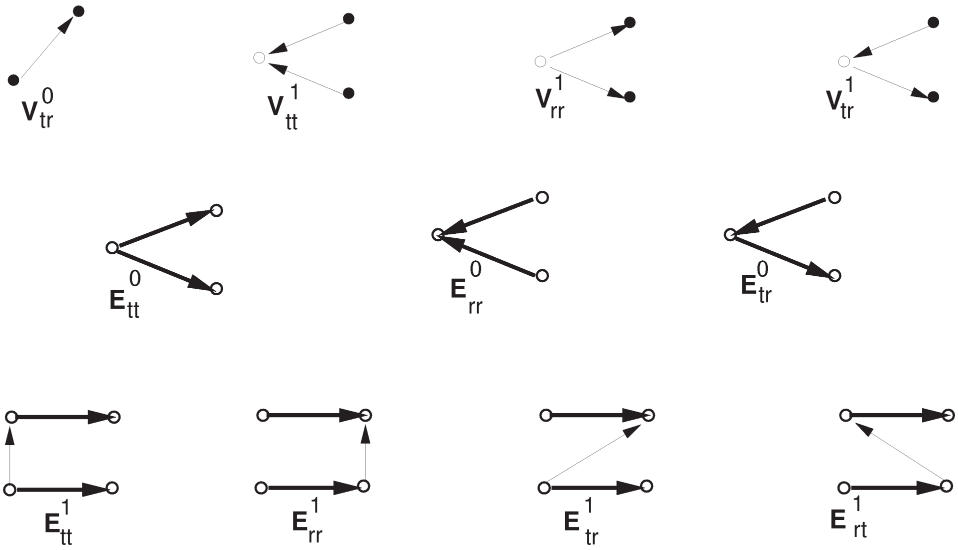

Figure 2. The eleven atomic constraints constituting the unified framework. The first row are vertex constraints, and the next two rows are edge constraints. The darkened vertices/edges are the ones that are mutually constrained, with the other vertices/edges causing the constraint.

performance guarantee R_A of an approximation algorithm A is the maximum, taken over all instances $I \in D_\Pi$, of $A(I)/\mathrm{OPT}(I)$, where $A(I)$ and $\mathrm{OPT}(I)$ are, respectively, the "sizes" of the solution given by an algorithm A and an optimum algorithm. In the context of graph coloring, the size of a solution is the number of colors used by the solution. Motivated by scalability considerations, we are mostly worried about the order of magnitude of the performance guarantee, and use the well-known "big-oh" notation for this. In conformance with existing notions, an approximation algorithm is *good* if its performance guarantee is O(1) (constant times optimum).

2.2. Constraints

We define a *constraint* as a symmetric relation between two vertices or two edges in a graph. If two vertices/edges are mutually related by some constraint c, we say that the vertices/edges are *constrained*. A constraint imposes a restriction on coloring: two vertices/edges that are mutually constrained must receive different colors for an assignment to be legal. Different constraints and combinations thereof give rise to different coloring problems, as discussed below.

We classify constraints according to whether they are between vertices or edges, the "separation" between them, and whether it is a transmitter and/or a receiver based constraint. Specifically, a constraint c is denoted using the syntax $c = \langle\varepsilon\rangle_{\langle d\rangle}^{\langle s\rangle}$, where $\varepsilon \in \{\mathrm{N}, \mathrm{E}\}$, $s \in \{0, 1\}$, $d \in \{\mathrm{tr}, \mathrm{tt}, \mathrm{rr}, \mathrm{rt}\}$. Here, ε is the entity (Node or Edge) being constrained, s is the forbidden separation between two vertices or edges, and d qualifies the separation by specifying its direction with respect to the transmitter and receiver. A separation of 0 between two vertices (edges) means that the vertices (edges) are adjacent, and a separation of 1 between two vertices (edges) means that there is one vertex (edge) between them. For example, if $c = V_{\mathrm{tt}}^1$, then two vertices u and v that are separated by another vertex w (i.e., $s = 1$) with an edge from the transmitters (i.e., $d =$ tt) of u, v to w are constrained. Figure 2 illustrates the various atomic constraints in our framework. Two constraints subsumed by this framework, namely V_{tt}^0 and V_{rr}^0 are invalid, since if two vertices are adjacent, then one of them must be the transmitter and the other the receiver. Note that V_{tr}^0 and V_{rt}^0 are equivalent constraints, as are E_{tr}^0 and E_{rt}^0. Henceforth, we shall treat V_{tt}^0 and V_{rr}^0 as "null" constraints and treat V_{tr}^0 (E_{tr}^0) as a "null" constraint when V_{rt}^0 (E_{rt}^0) is already present, and vice-versa.

Assignment problems are characterized by one or a combination of constraints, that is, a *constraint set*. For instance, $C = \{V_{\mathrm{tr}}^0, V_{\mathrm{tt}}^1, V_{\mathrm{rr}}^1\}$ is a constraint set. A problem characterized by a constraint set requires an assignment that satisfies *each* of the constraints in the constraint set. For instance, in an assignment problem characterized by $C = \{V_{\mathrm{tr}}^0, V_{\mathrm{tt}}^1, V_{\mathrm{rr}}^1\}$ two vertices cannot receive the same color if they are either adjacent (the V_{tr}^0 constraint), or have a common out-neighbor (the V_{tt}^1 constraint), or have a common in-neighbor (the V_{rr}^1 constraint). Formally,

Definition 2.1. An assignment characterized by a constraint set C, or a *C-assignment* of a graph G is a coloring of G such that for any two vertices/edges, if these vertices/edges are constrained by $c \in C$, then they are colored different.

Although our framework allows for a constraint set to contain a mixture of vertex and edge constraints, we are not aware of any practical application that demands this flexibility, and therefore, restrict our attention in this paper to constraint sets in which all constraints are either vertex based, or edge based.

The objective of this paper is to study the *optimization*

Table 1
Mapping real-life assignment problems into constraint sets.

Problem id	Constraint set	Real-life problem
1	V^0_{tr}	Cellular network freq. assignment
2	V^1_{tt}	TOCA/ROCA CDMA code assignment
3	V^0_{tr}, V^1_{tt}	(T/F)DMA broadcast schedule/assignment
4	E^0_{rr}, E^0_{tr}, E^0_{tt}	POCA CDMA code assignment
5	E^0_{rr}, E^0_{tt}, E^0_{tr}, E^1_{tr}	(T/F)DMA link schedule/assignment
6	E^0_{rr}, E^0_{tt}, E^1_{tr}	Full Duplex (T/F)DMA link schedule/assignment
7	E^0_{rr}, E^0_{tr}	(T/F)DMA schedule/assignment with directed antennas
8	E^0_{rr}, E^0_{tt}, E^0_{tr}, E^1_{tt}, E^1_{tr}	RTS-CTS protocols

version of the C-assignment problem for various C, that is, to minimize the number of colors used in the assignment.

Given an assignment problem P, we shall denote the set of constraints characterizing it by $\Gamma(P)$. We shall describe a constraint set by enumerating its contents explicitly, or express it using standard set theoretic notation, using the variables s, x and y. For instance, the following are equivalent:

$$C = \{V^s_{\mathrm{t}y},\ s \in \{0,1\},\ y \in \{\mathrm{t},\mathrm{r}\}\},$$
$$C = \{V^0_{\mathrm{tt}}, V^0_{\mathrm{tr}}, V^1_{\mathrm{tt}}, V^1_{\mathrm{tr}}\}$$

and so are the following:

$$C = \{E^1_{xx},\ x \in \{\mathrm{t},\mathrm{r}\}\},$$
$$C = \{E^1_{\mathrm{tt}},\ E^1_{\mathrm{rr}}\}.$$

Using the 4 vertex constraints illustrated in figure 2, 16 different constraint sets are possible (including the empty set indicating no constraints), and correspondingly 16 assignment problems. Similarly, using the 7 edge constraints illustrated in figure 2, 128 different constraint sets are possible, and correspondingly 128 assignment problems. As a study of each problem individually is clearly impractical, we use the notion of a *class of problems*, as defined below.

Definition 2.2. A class of assignment problems *covered* by a constraint set C, or a *C-covered class* is the set P of problems such that, for every $C_i \subseteq C$, the C_i-constrained assignment problem is an element of P.

For instance, the class of problems covered by $\{E^0_{\mathrm{tt}}, E^1_{\mathrm{tr}}\}$ is the set $\{P_1, P_2, P_3, P_4\}$ of problems where P_1 is characterized by $\{E^0_{\mathrm{tt}}\}$, P_2 is characterized by $\{E^1_{\mathrm{tr}}\}$, P_3 is characterized by $\{E^0_{\mathrm{tt}}, E^1_{\mathrm{tr}}\}$, and P_4 is characterized by $\{\ \}$ (no constraints on coloring). Such groupings are very useful since performance bounds, a subject of section 4, can often be established for a class as a whole.

2.3. Constraints for common assignment problems

Several of the assignment problems arising in practice can be mapped into corresponding constraint sets. We give in table 1 eight such real-life problems, and explain the scenarios in detail below. While the list is by no means exhaustive, the development of this paper will be guided by problems in this set, as they seem to have a relatively important relevance for today's applications. To facilitate reference in subsequent sections, each problem is assigned a "problem identifier" (first column) in table 1.

We explain the problems briefly, summarizing relevant previous work, if any. Unless otherwise specified, half duplex communications is assumed.

1. *Cellular network frequency assignment.* In a cellular network, adjacent cells need to be assigned different frequencies [22]. Consider a graph model of the network where vertices represent cells and edges represent cell-adjacency. Then, two adjacent vertices must receive different colors (frequencies), and thus are V^0_{tr} constrained. In graph-theoretic terms, this is the well-known vertex coloring problem. This problem has been well studied graph-theoretically [16] and in the context of cellular network design [4,15,22].
2. *TOCA/ROCA CDMA code assignment.* This problem arises in a spread-spectrum based access scheme in which orthogonal CDMA codes are to be assigned to nodes. In Transmitter Oriented Code Assignment (TOCA), a node transmits on the code and is receiving code agile, while in Receiver Oriented Code Assignment (ROCA), a node receives on this code and is transmission code agile. In both cases, for collision-free transmissions, it is required that if there is a node that can hear two nodes A, B (the V^1_{tt} constraint), then A, B use different codes. This problem has been considered in [3,17,23].
3. *(T/F)DMA Broadcast Schedule/Assignment.* This problem arises in multihop networks when a node's transmission is intended for all of its neighbors (e.g., when the packet has to be broadcast). Thus, if a node X is assigned a time-slot/frequency q, then every out-neighbor Y of X should not be assigned q since it will have to tune to receive on q (V^0_{tr} constraint), and every in-neighbor Z of Y should also not be assigned q since otherwise there will be collision at Y (V^1_{tt} constraint). This problem has been considered in [6,12,29,30].
4. *POCA CDMA code assignment.* This is the "edge version" of problem number 2. CDMA codes are to be assigned not to nodes, but to links between nodes such that no two links sharing the same node are assigned

the same code (E^0_{tt},E^0_{rr},E^0_{tr}). This problem is mentioned in [17]. In graph-theoretic terms, this is the well-known edge coloring problem [16].

5. *(T/F)DMA link schedule/assignment.* Here, links between nodes are to be assigned time slots/frequency bands such that interference is completely avoided. This requires that if two links are assigned the same slot/frequency, then there be no link from the transmitter of one link to the receiver of the other link (E^1_{tr}). Moreover, half duplex considerations imply that no two adjacent links are assigned the same slot/frequency (E^0_{tt},E^0_{rr},E^0_{tr}). This problem has been considered in [7,12,29,33].

6. *Full Duplex (T/F)DMA link schedule/assignment.* This is the version of problem 5 when the nodes are capable of full duplex communications, that is, can transmit and receive simultaneously. The E^0_{tr} constraint can then be removed.

7. *(T/F)DMA link schedule/assignment with multiple directional antennas.* In systems with multiple directional antennas, multiple transmissions on the same time/frequency channel are possible without interference. Further, the transmission being directional, it does not cause interference with transmissions from other nodes. Thus, the E^0_{tt} and E^1_{tr} constraints can be removed from the standard link schedule/assignment problem (number 5) resulting in the indicated constraint set.

8. *RTS-CTS protocols.* In wireless LAN protocols such as IEEE 802.11, a control handshake (request-to-send (RTS), clear-to-send (CTS)) is required before data packets can be exchanged. Suppose a node A wishes to transmit a data packet to a node B. Then, no node C that A (or B) can hear is allowed to transmit, since the transmission will collide with the CTS from B to A (or RTS from A to B). These manifest themselves as E^1_{tr} and E^1_{tt} constraints, along with the usual E^0_{xy} constraints.

3. A unified algorithm

We now describe a unified algorithm for (T/F/C) DMA channel assignment (henceforth referred to as algorithm UxDMA). Given a graphical representation of a (multihop) wireless network, algorithm UxDMA produces an assignment of positive integers ("colors") to vertices or edges of G, subject to a set of constraints on the assignment.

The algorithm consists of two phases – a *labelling* phase and a *coloring* phase. In the labelling phase, each vertex in the graph is assigned a *unique* label between 1 and n, where n is the total number of vertices. The coloring phase follows the labelling phase. Here, vertices are considered in decreasing order of labels, and if it is edge coloring, then the edges incident to the considered vertex are colored, or if it is vertex coloring, the considered vertex itself is colored. The color is chosen in a *greedy* fashion. That is, the least color (integer) that can be assigned without violating any of the constraints is chosen. After the vertex labelled 1 one is colored, the algorithm terminates.

The crux of the algorithm lies in *how the labelling is done*. Several *ordering* heuristics are possible, but in this paper we have studied only three – random, minimum neighbors first, and progressive minimum neighbors first.

- *Random* (RAND). This is the simplest ordering, and as the name indicates, the vertices are labelled in random order.
- *Minimum Neighbors First* (MNF). Here, vertices that have a smaller number of neighbors are assigned a smaller label. Consequently, the coloring is done by first picking high neighborhood vertices first.
- *Progressive Minimum Neighbors First* (PMNF). This is similar to MNF with a subtle but crucial difference - after labelling a vertex, this vertex and the edges incident on this vertex are ignored while processing the rest of the vertices (one could think of it as "deleting" the vertex and the edges or just the edges). The result is that unlike MNF, the neighborhood of a vertex keeps on changing as other vertices are processed. The ordering could be significantly different from MNF, as could the performance.

For the RAND and MNF based heuristics, it is not necessary to first label the vertices. We have chosen to describe it thus for the sake of uniformity in presentation and ease of unification.

Many of the heuristics mentioned in packet-radio channel assignment literature (e.g., [6,7,10,17,30,33]) are equivalent to our algorithm with a RAND ordering. Its appeal lies in its simplicity and relative ease with which a distributed version can be implemented. However, as we mentioned in section 1, and as we shall see in sections 4, the RAND ordering is significantly inferior to PMNF in terms of worst-case guarantees and somewhat inferior to both MNF and PMNF in terms of expected performance.

A notable feature of our algorithm is that a single implementation can be made to function, by a runtime choice of parameters, either as a simple RAND assigner, or a sophisticated PMNF/MNF assigner with improved performance guarantees. This should be clear from the specification below.

3.1. Formal specification

The main algorithm is given first, followed by the procedures it uses. Indentation is used to delineate block structure.

Algorithm UxDMA

Input: (1) Directed graph $G = (V, A)$, (2) element type $\Psi \in \{$V(ertex), E(dge)$\}$ to be colored (assigned), (3) constraint set C which is either $C \subseteq \{V^0_{\text{tr}}, V^1_{\text{tt}}, V^1_{\text{tr}}, V^1_{\text{rr}}\}$, or

$C \subseteq \{E^0_{tt}, E^0_{tr}, E^0_{rr}, E^1_{tt}, E^1_{tr}, E^1_{rt}, E^1_{rr}\}$, (4) ordering $\omega \in$ {RAND, MNF, PMNF}

Output: A C-constrained assignment of q colors 1 through q to vertices (if $\Psi = \text{V}$) or edges (if $\Psi = \text{E}$) of G

begin
1. **for** all elements e of type Ψ **do**
2. color(e) $\leftarrow$0
3. Assign-Label(G, ω)
4. **for** j from largest down to smallest label **do**
5. let u $\leftarrow$vertex with label j
6. **for** each element e in Surround(G, Ψ, u) **do**
7. color(e) $\leftarrow$First-Available-Color(G, e, C)

end

Procedure Assign-Label($G = (V, A)$, ω)
Label vertices in G according to specified ordering $\omega \in$ {RAND, MNF, PMNF}
1. **for** labels l from 1 through $|V|$ **do**
2. **if** ω is RAND **then**
3. pick unlabelled vertex $u \in V$ at random
4. label(u) $\leftarrow l$
5. **if** ω is MNF **then**
6. pick unlabelled vertex $u \in V$ of minimum neighbors in G
7. label(u) $\leftarrow l$
8. **if** ω is PMNF **then**
9. pick unlabelled vertex $u \in V$ of minimum neighbors in G
10. label(u) $\leftarrow l$
11. delete all edges incident on u from G

Procedure Surround(G, Ψ, u)
1. **if** Ψ is VERTEX **then**
2. $S \leftarrow u$
3. **else**
4. $S \leftarrow$set of incoming and outgoing edges incident on u in G
5. **return**(uncolored subset of S)

Procedure First-Available-Color(G, e, C)
Return the least color that e can be assigned
1. Taken $\leftarrow$\{ \}
2. **for** each constraint $c \in C$ **do**
3. **for** each colored element f of type same as e such that e, f, are related by constraint c **do**
4. Taken $\leftarrow$Taken $\cup$ color(f)
5. **return**(smallest color $\notin$ Taken)

4. Theoretical analysis

The correctness of algorithm UxDMA follows from the fact that the color chosen for an element (line 7 of UxDMA) is never in the set of constraining elements' colors (lines 3, 4 of First-Avail-Color). In this section, we analyze the worst-case performance and running time of UxDMA.

We present results on the worst-case performance for algorithm UxDMA as a function of the input parameters, in particular the constraint set and the ordering. The results are presented in the order of decreasing cardinality of the constraint set covering the problem class. That is, we first describe loose upper bounds on the performance for a large class of problems, then describe tighter bounds for a subclass, still tighter bounds for a sub-sub-class and so on. A similar attitude is taken toward the ordering – first results that are applicable to all three orderings are presented, and then results applicable only to PMNF.

Note that each of the theorems below is applicable to not just *one* assignment problem, but a *class* of assignment problems, namely those covered by indicated constraint set.

We begin with a straightforward "universal" theorem that is applicable to all of the problems and all of the orderings. Recall that ρ is used to denote the maximum total degree of the graph.

Theorem 4.1. For the class of assignment problems covered by $C = \{\Psi^s_{xy}, \Psi \in \{V, E\}, x, y \in \{\text{t}, \text{r}\}, s \in \{0, 1\}\}$, algorithm UxDMA with any ordering $\in$ {RAND,MNF, PMNF} has a performance guarantee of O(ρ^2).

Proof. This follows easily from the fact that any combination of constraints is such that only entities within 0 or 1 separation from a given entity can restrict the color of that entity. For both vertices and edges, a given vertex (or edge) has at most ρ vertices (or 2($\rho - 1$) edges) at a separation of 0, and at most $\rho(\rho - 1)$ vertices (or $2(\rho - 1)^2$ edges) at a separation of 1. While coloring this vertex (or edge), we use the least available color (line 7 of algorithm UxDMA). Even if all of the vertices and edges at separation 0 and 1 are different and restrict the color of the given vertex (or edge), the vertex can be assigned color $1 + \rho + \rho(\rho - 1) = \rho^2 + 1$, and the edge can be assigned color $1 + 2(\rho - 1) + 2(\rho - 1)^2 = 2\rho^2 - 2\rho + 1$. Thus, any input graph can be colored with O(ρ^2) colors. Since the optimum is at least 1, the performance guarantee of UxDMA is O(ρ^2). □

Our next result shows that we can obtain an improvement by a factor of ρ for vertex coloring when only tt and rr separation-1 constraints are considered.

Theorem 4.2. For the class of assignment problems covered by $C = \{V^0_{xy}, V^1_{xx}, x, y \in (\text{t}, \text{r})\}$, algorithm UxDMA with any ordering $\in$ {RAND,MNF,PMNF} has a performance guarantee of O(ρ).

Proof. Let P be an assignment problem. We consider two cases (recall that $\Gamma(P)$ denotes the constraint set characterizing P):

(1) $V^1_{xx} \notin \Gamma(P)$. That is, the problem is characterized by zero-separation constraints only. Consider a vertex u picked in line 5 of algorithm UxDMA. Since only V^0_{xy} constraints are present, exactly the vertices adjacent to u are picked

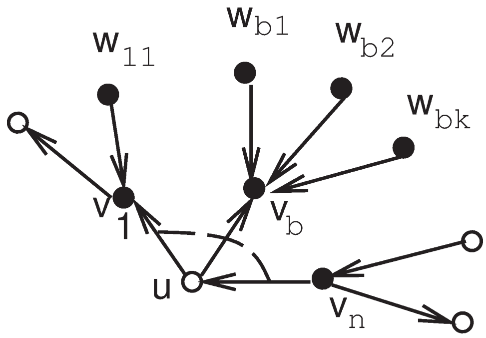

Figure 3. Example showing color restricting vertices (darkened) for vertex u, when the constraint set is $\{V^0_{\mathrm{tr}}, V^1_{\mathrm{tt}}\}$.

for f in line 3 of procedure First-Available-Color. Since there are at most ρ such adjacent vertices, no more than ρ different colors can be used up for them, leaving the $(\rho+1)$th color for u. Therefore, the graph can be colored using $\rho+1$ colors, and since the optimum is trivially at least 1, the performance guarantee is O(ρ).

(2) $V^1_{xx} \in \Gamma(P)$. That is, both zero and one separation constraints are present. In this case, in line 3 of First-Available-Color, vertices adjacent to the vertex u picked in line 5 of UxDMA as well as vertices adjacent to these vertices are candidates for f (refer figure 3). Suppose that vertices v_i, $1 \leqslant i \leqslant n$, adjacent to u and vertices w_{ij}, $1 \leqslant j \leqslant m_i$, adjacent to v_i are picked for f. Clearly, u can be colored with a color at most

$$\text{color}(u) = 1 + n + \sum_{i=1}^{n} m_i. \tag{1}$$

Consider a vertex v_b adjacent to u and the vertices w_{bj} adjacent to v_b that were picked for f in line 3 of procedure First-Available-Color. Suppose that a vertex w_{bq} constrains u by $c \in C$. Then, by definition, $c \in \{V^1_{\mathrm{tt}}, V^1_{\mathrm{rr}}\}$. However, for either of these two constraints, every vertex w_{bl}, $l \neq q$, is related by the same constraint c to w_{bq}. In other words, the w_{ij} vertices, $1 \leqslant j \leqslant m_i$, constrain each other. Figure 3 illustrates a scenario when the separation-1 constraint is V^1_{tt} (the V^1_{rr} case is similar). Notice that the w_{bi} vertices are mutually V^1_{tt} constrained. Therefore, these vertices must receive different colors in any coloring, including the optimum coloring. Clearly then, if OPT denotes optimum number of colors,

$$\text{OPT} \geqslant \text{MAX}(m_1, m_2, \ldots, m_n). \tag{2}$$

From (1),

$$\begin{aligned}\text{color}(u) &\leqslant 1 + n + n\text{MAX}(m_1, m_2, \ldots, m_n)\\ &\leqslant 1 + n\text{OPT} + n \quad \text{(from 2)}\\ &\leqslant 1 + \rho(\text{OPT} + 1) \quad (\text{since } n \leqslant \rho).\end{aligned}$$

Thus, at most 1 + ρ(OPT+1) colors are used by UxDMA to color the input graph. It follows that the performance guarantee is O(ρ). □

If there are no unidirectional edges, then V^1_{tt} would imply V^1_{rr}, and hence the above theorem would be valid for $C = \{V^s_{xy},\ s \in \{0,1\},\ x,y \in \{\mathrm{t},\mathrm{r}\}\}$.

A similar improvement can be obtained for edge coloring, for the problem class such that each problem de facto has all of the 0-separation constraints in the characterizing constraint set.

Theorem 4.3. For the class of assignment problems covered by $C = \{E^s_{xy},\ s \in \{0,1\},\ x,y \in \{\mathrm{t},\mathrm{r}\}\}$ such that for each problem P in the class, $\{E^0_{xy},\ x,y \in \{\mathrm{t},\mathrm{r}\}\} \subseteq \Gamma(P)$, algorithm UxDMA with any ordering has performance guarantee O(ρ).

Proof. The class of problems here is subsumed by the class of problems addressed in theorem 4.1. From the derivation there, at most O(ρ^2) colors are needed by UxDMA to color the input graph. However, here we have $\{E^0_{xy},\ x,y \in \{\mathrm{t},\mathrm{r}\}\} \subseteq \Gamma(P)$, implying that the optimum is at least ρ. It follows then that the performance guarantee is O(ρ). □

If separation-1 constraints are ignored, then we can get a constant times optimum algorithm.

Theorem 4.4. For the class of assignment problems covered by $C = \{E^0_{xy},\ x,y \in \{\mathrm{t},\mathrm{r}\}\}$ algorithm UxDMA with any ordering has performance guarantee O(1).

Proof. Since only separation-0 constraints are present, the number of colors restricting the color of an edge $(u \to v)$ is at most $2(\rho-1)$ ($\rho-1$ incident on each of u and v). Thus, UxDMA uses no more than 2ρ-1 colors. The optimum is clearly at least ρ, and thus the performance guarantee is O(1). □

The results derived thus far were applicable no matter what ordering was used. We now show that improved performance guarantees can be obtained if the PMNF ordering is used.

Crucial to the performance improvement obtained by a PMNF ordering is the following fact, which is basically a generalization of a well-known result on planar graphs [16, 25].

Lemma 4.1. Every undirected graph of thickness θ contains at least one vertex with $6\theta - 1$ neighbors or less.

Proof. Since G is of thickness θ, G can be partitioned (by definition of thickness) into mutually disjoint graphs $G_1, G_2, \ldots, G_\theta$, such that for all $1 \leqslant i \leqslant \theta$, G_i is a planar subgraph of G. Let $m(G_i)$ and $n(G_i)$ denote the number of edges and vertices of G_i, respectively. Let m and n denote the same for the entire graph G. By a well-known basic property of planar undirected graphs [16,25], we have, for all i,

$$m(G_i) \leqslant 3n(G_i) - 6. \tag{3}$$

Summing this over all partitions maintains the inequality and so we have

$$\sum_{i=1}^{\theta} m(G_i) \leqslant \sum_{i=1}^{\theta} 3\big(n(G_i) - 6\big)$$
$$\Rightarrow m \leqslant \sum_{i=1}^{\theta} (3n-6)$$
$$\Rightarrow m \leqslant (3n-6)\theta. \quad (4)$$

Now, to prove the lemma, assume to the contrary that every vertex in G is of degree 6θ or more. Clearly, then $2m \geqslant 6\theta n$, that is, $m \geqslant 3n\theta$, which contradicts equation (4). □

As a consequence of this fact, there is a limit on the number of larger-labelled neighbors for a given vertex, as proven below.

Lemma 4.2. In the labelling (by procedure Assign-Label) of an input directed graph $G = (V, A)$ of thickness θ using ordering PMNF, every vertex has at most $6\theta - 1$ neighbors labelled larger than itself.

Proof. Consider the undirected equivalent G' of G, obtained using the same set of vertices and an undirected edge between a pair of vertices u, v if and only if $(u \rightarrow v)$ or $(v \rightarrow u)$ is an edge in G. We shall prove the result for G'. Since it is the number of *neighbors* that are of interest, it is easily seen that the result extends to G.

Consider a vertex u picked for labelling in the jth iteration in line 9 of Assign-Label using PMNF ordering. Let G'_j denote the subgraph containing u, during the jth iteration, formed by successive deletions of edges at iterations 1 through $j-1$, from G'. Since G' is of thickness θ, G'_j (a subgraph) is also of thickness θ. Thus, by lemma 4.1, G'_j contains at least one vertex with at most $6\theta - 1$ neighbors. Since u is chosen to be the vertex with the minimum degree, it follows that it has at most $6\theta - 1$ neighbors in G'_j. Further, these neighbors are exactly those that have not yet been labelled because by virtue of line 11 of Assign-Label, a labelled vertex is no longer adjacent to any vertex. Therefore, these neighbors will receive a label greater than u's (in a later iteration). Consequently, when the loop in line 1 terminates, every vertex has at most $6\theta - 1$ neighbors labelled larger than itself. □

We are now ready to prove the UxDMA performance guarantee results with respect to the PMNF ordering. The first result depends on the maximum degree ratio $\delta_{\mathrm{r}} = \max(\rho_{\mathrm{out}}/\rho_{\mathrm{in}}, \rho_{\mathrm{in}}/\rho_{\mathrm{out}})$.

Theorem 4.5. For the class of assignment problems covered by $C = \{V_{xy}^s,\ x \in \{\mathrm{t},\mathrm{r}\},\ s \in \{0,1\}\}$, and ordering PMNF (Progressive Minimum Neighbors First), algorithm UxDMA has a performance guarantee of $\mathrm{O}(\delta_{\mathrm{r}}\theta)$.

Proof. We consider two cases:

(1) $V_{xy}^1 \notin C$. Consider a vertex u picked in line 5 of UxDMA. Since only zero-separation constraints are present, the color for u is precluded only by vertices adjacent to u. By lemma 4.2, PMNF labelling is such that at most $(6\theta - 1)$ neighbors of u are labelled larger than u. Since the coloring is done in decreasing order of vertex labels, at most $(6\theta - 1)$ neighbors are already colored. Thus, the graph can be colored with at most 6θ colors. Since the optimum is trivially at least 1, the performance guarantee is $\mathrm{O}(\theta)$, and also $\mathrm{O}(\delta_{\mathrm{r}}\theta)$ since δ_{r} is at least 1 by definition.

(2) $V_{xy}^1 \in C$. Consider a vertex u picked in line 5 of UxDMA. We separate the neighbors of u into two disjoint sets of vertices: L-vertices, that are labelled larger than u, and S-vertices that are labelled smaller than u. Suppose that there are n_{l} L-vertices vertices $p_1, p_2, \ldots, p_{n_{\mathrm{l}}}$, and n_{s} S-vertices $q_1, q_2, \ldots, q_{n_{\mathrm{s}}}$. Note that because of the sequence in which coloring is done (largest to smallest), only the L-vertices can be already colored when u is picked for coloring. Now consider neighbors of L-vertices and S-vertices. Let $A(p_i)$ and $A(q_i)$ denote the set of vertices adjacent to p_i and q_i, respectively, and not adjacent to u that are picked as constraining u (i.e., as f in line 3 of First-Available-Color). Thus, in the worst-case, the first available color for u

$$\mathrm{color}(u) \leqslant 1 + n_{\mathrm{l}} + \sum_{i=1}^{n_{\mathrm{l}}} \big|A(p_i)\big| + \sum_{i=1}^{n_{\mathrm{s}}} \big|A(q_i)\big|$$
$$\leqslant 1 + n_{\mathrm{l}} + M_p \cdot n_{\mathrm{l}} + M_q \cdot n_{\mathrm{s}}, \quad (5)$$

where $M_p = \max_i(|A(p_i)|)$, $1 \leqslant i \leqslant n_{\mathrm{l}}$, and $M_q = \max_i(|A(q_i)|)$, $1 \leqslant i \leqslant n_{\mathrm{s}}$.

The variables making up equation (5) are subject to the following:

$$n_{\mathrm{l}} \leqslant 6\theta - 1,$$
$$M_p \leqslant \rho - 1,$$
$$n_{\mathrm{s}} + n_{\mathrm{l}} \leqslant \rho,$$
$$M_q \leqslant 6\theta - 2.$$

The first of the above equations is due to lemma 4.2, the second and third are due to the fact that the number of neighbors of a vertex is no more than ρ. To see why the last equation is true, consider a S-vertex q_j, and note that $\mathrm{label}(q_j) < \mathrm{label}(u)$. Now consider a vertex y that is a neighbor of q_j but not of u, and suppose it is included as f in line 3 of First-Available-Color. For this to happen, y must be colored, and hence $\mathrm{label}(y)$ must exceed $\mathrm{label}(u)$. But we just noted that $\mathrm{label}(q_j) < \mathrm{label}(u)$, and hence it must be that $\mathrm{label}(y) > \mathrm{label}(q_j)$. By lemma 4.2, however, there can be at most $6\theta - 1$ neighbors labelled larger than q_j, and u is by definition one of them. Thus, there can be at most $6\theta - 2$ vertices adjacent to any q_j restricting the color of u.

Thus, using these constraints in (5), and ignoring '−' terms as inequality preserving terms,

$$\begin{aligned}\text{color}(u) &\leqslant 1 + 6\theta - 1 + (6\theta - 1)(\rho - 1) \\ &\quad + (\rho - 6\theta + 1)(6\theta - 2) \\ &\leqslant 6\theta + 6\theta\rho + (\rho + 1)(6\theta). \end{aligned} \tag{6}$$

Since the constraints (see theorem statement) are one or both of V^1_{tt} or V^1_{rr}, the optimum must be at least $\min(\rho_{\text{out}}, \rho_{\text{in}})$. This is because the neighbors of the vertex with ρ_{in} neighbors are mutually V^1_{tt} constrained, and the neighbors of the vertex with ρ_{out} neighbors are mutually V^1_{rr} constrained.

Since vertex' total degree is equal to the sum of its incoming and outgoing degrees, which are in turn at most ρ_{in} and ρ_{out}, respectively, we have $\rho \leqslant \rho_{\text{in}} + \rho_{\text{out}}$. We substitute for ρ in (6), and express the performance guarantee in terms of the degree ratio δ_{r}.

$$\begin{aligned} R_{\text{UxDMA}} &\leqslant \frac{6\theta}{\min(\rho_{\text{in}}, \rho_{\text{out}})} + \frac{6\theta(\rho_{\text{out}} + \rho_{\text{in}})}{\min(\rho_{\text{in}}, \rho_{\text{out}})} \\ &\quad + \frac{6\theta(\rho_{\text{out}} + \rho_{\text{in}} + 1)}{\min(\rho_{\text{in}}, \rho_{\text{out}})} \\ &\leqslant \frac{6\theta}{\min(\rho_{\text{in}}, \rho_{\text{out}})} + 6\theta\left(\max\left(1 + \frac{\rho_{\text{out}}}{\rho_{\text{in}}}, 1 + \frac{\rho_{\text{in}}}{\rho_{\text{out}}}\right)\right) \\ &\quad + 6\theta\left(1 + \max\left(1 + \frac{\rho_{\text{out}}}{\rho_{\text{in}}}, 1 + \frac{\rho_{\text{in}}}{\rho_{\text{out}}}\right)\right) \\ &\leqslant \frac{6\theta}{\min(\rho_{\text{in}}, \rho_{\text{out}})} + 6\theta(1 + \delta_{\text{r}}) + 6\theta(2 + \delta_{\text{r}}). \end{aligned}$$

The first term is $O(\theta)$ and the second and third terms are $O(\theta\delta_{\text{r}})$. Thus, the performance guarantee of UxDMA is $O(\theta\delta_{\text{r}})$. □

For edge assignment too, PMNF can guarantee improved bounds compared to RAND and MNF. We have two results, both based on the following lemma.

Lemma 4.3. For the class of assignment problems covered by $C = \{E^s_{xy},\ s \in \{0,1\},\ x, y \in \{\text{t}, \text{r}\}\}$ algorithm UxDMA with PMNF ordering uses no more than $O(\rho\theta)$ colors.

Proof. Consider the coloring of an edge $(u \to v)$ in line 7 of UxDMA. Without loss of generality, assume that label(v) $>$ label(u). We divide colored edges constraining the color of $(u \to v)$ into three disjoint sets – U-edges, incident on u alone, V-edges, incident on v alone, and the edge $(v \to u)$ (see figure 4). To prove the result, we prove bounds on the number of U- and V-edges. We first count the V-edges and then the U-edges below.

V-edges. We further divide vertices adjacent to v into n_{l} vertices (say $p_1, p_2, \ldots, p_{n_{\text{l}}}$) such that label($p_i$) $>$ label(v), and n_{s} vertices (say $q_1, q_2, \ldots, q_{n_{\text{s}}}$) such that label($q_i$) $<$ label(v). Since we color (see line 4 of UxDMA) edges incident on larger labelled vertices before smaller labelled ones, all edges incident on p_i are already colored. Some of the edges incident on each q_i may also be colored (refer figure 4). Letting $d(p_i)$ denote the degree of p_i and $I(q_i)$

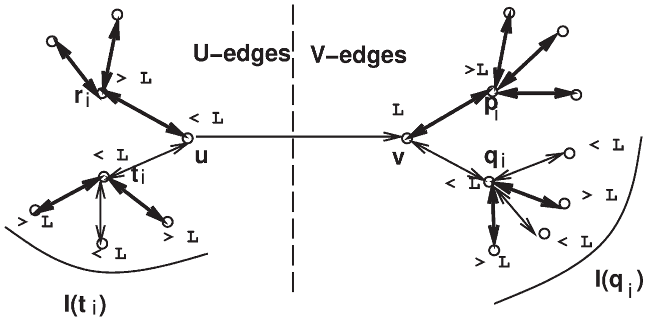

Figure 4. Example showing color restricting edges (darkened) for edge $(u \to v)$, the partition into U- and V-edges, and the relative labelling on the relevant vertices.

denote the set of edges already colored and incident on q_i, we have

$$n(V\text{-edges}) = \sum_{i=1}^{n_{\text{l}}} d(p_i) + \sum_{i=1}^{n_{\text{s}}} \left|I(q_i)\right|. \tag{7}$$

These variables are bounded by

$$\begin{aligned} n_{\text{l}} &\leqslant 6\theta - 1, \\ d(p_i) &\leqslant \rho, \\ n_{\text{s}} &\leqslant \rho, \\ \left|I(q_i)\right| &\leqslant 2(6\theta - 1). \end{aligned}$$

The first equation above is due to lemma 4.2 (note that n_{l} is the number of vertices labelled larger than v), the second and third are obvious since ρ is the maximum degree. To see why the last equation is true, consider a particular vertex q_j, and consider an edge $x \leftrightarrow y$ (recall that this means an edge $(q_j \to y)$ or $(y \to q_j)$, and observe that it is represented by a double-arrowed edge in figure 4) that is colored. Since $q_j \leftrightarrow y$ is colored, either label(q_j) $>$ label(v) or label(y) $>$ label(v). But by premise, label(q_j) $<$ label(v), and hence label(y) must be greater than label(v) (refer figure 4), and consequently greater than label(q_j). By lemma 4.2, however, there can be at most $6\theta - 1$ such neighbors y, and hence $2(6\theta - 1)$ such edges. Since such edges are precisely those that constitute I(q_i), the last equation above is true.

Substituting the bounds on variables into equation (7), we have

$$\begin{aligned} n(V\text{-edges}) &\leqslant \rho(6\theta - 1) + 2\rho(6\theta - 1) \\ &\leqslant 3\rho(6\theta - 1). \end{aligned} \tag{8}$$

U-edges. This parallels the case for V-edges, but has subtle differences that justify the elaboration. Similar to that case, we divide vertices adjacent to u into m_{l} vertices $r_1, r_2, \ldots, r_{m_{\text{l}}}$ such that label(r_i) $>$ label(v), and m_{s} vertices $t_1, t_2, \ldots, t_{m_{\text{s}}}$ such that label(t_i) $<$ label(v). Letting $d(r_i)$ denote the degree of r_i and $I(t_i)$ the set of colored edges incident on t_i,

$$n(U\text{-edges}) = \sum_{i=1}^{m_{\text{l}}} d(r_i) + \sum_{i=1}^{m_{\text{s}}} \left|I(t_i)\right|. \tag{9}$$

Table 2
Best performance guarantees for eight real-life problems identified in section 2.3, for RAND, MNF and PMNF orderings.

Problem id	Constraint set	PG for RAND, MNF (theorem)	PG for PMNF (theorem)
1	V^0_{tr}	O(ρ) (4.2)	O($\theta\delta_r$) (4.5)
2	V^1_{tt}	O(ρ) (4.2)	O($\theta\delta_r$) (4.5)
3	V^0_{tr}, V^1_{tt}	O(ρ) (4.2)	O($\theta\delta_r$) (4.5)
4	E^0_{rr}, E^0_{tr}, E^0_{tt}	O(1) (4.4)	O(1) (4.4)
5	E^0_{rr}, E^0_{tt}, E^0_{tr}, E^1_{tr}	O(ρ) (4.3)	O(θ) (4.7)
6	E^0_{rr}, E^0_{tt}, E^1_{tr}	O(ρ^2) (4.1)	O($\rho\theta$) (4.6)
7	E^0_{rr}, E^0_{tr}	O(1) (4.4)	O(1) (4.4)
8	E^0_{rr}, E^0_{tt}, E^0_{tr}, E^1_{tt}, E^1_{tr}	O(ρ) (4.3)	O(θ) (4.7)

From the premise (see beginning of proof), label(u) $<$ label(v), and hence label(r_i) $>$ label(u). Thus, by lemma 4.2, $m_{\mathrm{l}} \leqslant 6\theta - 1$. Thus the bounds are

$$m_{\mathrm{l}} \leqslant 6\theta - 1,$$
$$d(r_i) \leqslant \rho,$$
$$m_{\mathrm{s}} \leqslant \rho,$$
$$\left|I(t_i)\right| \leqslant 2(6\theta - 1).$$

To see why the last equation is true, consider a particular t_j and notice that for an edge $t_j \leftrightarrow y$ to be colored, label(y) must be greater than label(v) (since label(t_j) $<$ label(v) by premise). But since label(v) $>$ label(u) by premise, we have label(y) $>$ label(u), and by lemma 4.2, there can be at most $6\theta - 1$ such vertices, and, consequently, at most twice this number of edges.

Thus, equation (9) can be bounded as

$$n(U\text{-edges}) \leqslant 3\rho(6\theta - 1). \qquad (10)$$

From equations (8) and (10), it follows that the entire graph can be colored with at most O($\rho\theta$) colors. □

Theorem 4.6. For the class of assignment problems covered by $C = \{E^s_{xy},\ s \in \{0,1\},\ x,y \in \{\mathrm{t},\mathrm{r}\}\}$ algorithm UxDMA with PMNF ordering has a performance guarantee of O($\rho\theta$).

Proof. Follows from lemma 4.3 and the fact that the optimum is at least 1. □

Theorem 4.7. For the class of assignment problems covered by $C = \{E^s_{xy},\ s \in \{0,1\},\ x,y \in \{\mathrm{t},\mathrm{r}\}\}$ such that each problem P in the class, $\{E^0_{xy},\ x,y \in \{\mathrm{t},\mathrm{r}\}\} \subseteq \Gamma(P)$, algorithm UxDMA with PMNF ordering has a performance guarantee of O(θ).

Proof. Since $\{E^0_{xy},\ x,y \in \{\mathrm{t},\mathrm{r}\}\}$ constraints are present in each problem, all adjacent edges have to receive different colors in any assignment. Thus, the optimum is at least ρ. From lemma 4.3, the assignment problem class can be colored with at most O($\rho\theta$) colors. It follows that the performance guarantee is O(θ). □

The foregoing theorems are summarized in table 2. For each problem, we note the best provable performance guarantee for the RAND and MNF orderings (column 3) and for the PMNF ordering (column 4), along with the theorem that proves it. The reader is referred to table 1 for a description of each problem.

Running time. Let v and e denote the number of vertices and edges in the graph. Then, lines 1, 2 of UxDMA takes O(e). Line 3 (Assign-Label) takes O(v) for RAND, O(v^2) for MNF and O($v(v+\rho)$) for PMNF using simple (linear search) structures.[4] Lines 6, 7 take O(ρ^2), and hence lines 4–7 take O($e\rho^2$). Thus, the worst-case running time is O($v(v+\rho)$ + $e\rho^2$) for PMNF, MNF (simple structures) and is O($e\rho^2$) for RAND.

5. Experimental results

For our experimental study, we have taken as representatives one node assignment problem, namely problem number 3 (T/F DMA broadcast schedule/assignment), and one link assignment problem, namely problem number 5 (T/F DMA link schedule/assignment).

Our experiments have been conducted under the assumption of a noiseless, immobile radio network, where the nodes are distributed in a given area and may each have a different transmission range. In this context, the network may be represented by the 3-tuple (N, R_i, P), where N is the number of nodes, R_i is the transmission range of node i and $P = \{(x_i, y_i),\ 1 \leqslant i \leqslant N\}$ is the set of locations for each of the nodes. The location of a node is generated randomly, using a uniform distribution for its X and Y coordinates, in a given area. We convert this network into a graph $G = (V, A)$, so that $|V| = N$, and $(u,v) \in A$ if and only if the Euclidean distance between (x_u, y_u) and (x_v, y_v) is less than or equal to R_u. Note that this might result an in unidirectional edge, but only if $R_u \neq R_v$.

We have studied the experimental performance of our algorithms by generating a large number of random graphs in an area of 400×400 units for values for N = 100–500 nodes and R_i = 20–60 units. For each of the broadcast and link assignments, we study the performance (number of slots (colors) used) by algorithm UxDMA for each ordering (RAND, MNF, PMNF), along with a lower bound for the

[4] The use of Fibonacci heaps will reduce the time for PMNF to O($e + v\log(v)$).

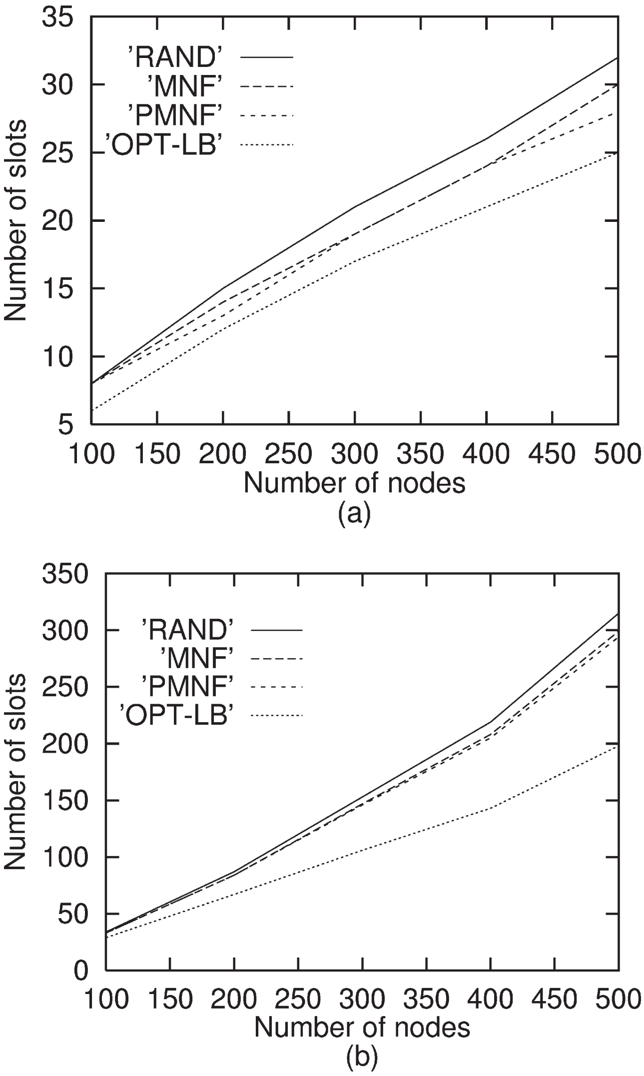

Figure 5. Performance of heuristics and the lower bound on optimum, versus N, with $R = 40$ (same for every node), for broadcast assignment (a) and link assignment (b).

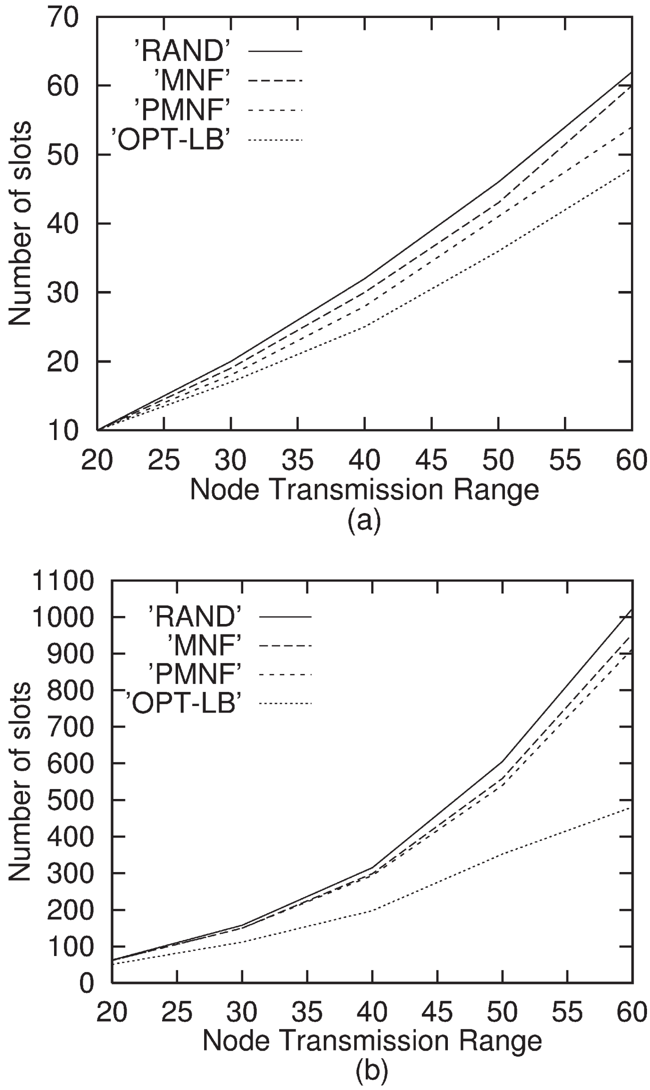

Figure 6. Performance of heuristics and the lower bound on optimum, versus R (same for every node), with $N = 500$, for broadcast assignment (a) and link assignment (b).

optimum.[5] Figures 5(a) and (b) show, respectively, the number of slots for broadcast scheduling (number 3) and link scheduling (number 5) as a function of N, with $R_i = R = 40$ units. Figures 6(a) and 6(b) show, respectively, the number of slots for broadcast and link scheduling as a function of $R_i = R$, with $N = 500$ nodes. To study the effect of unidirectional links, we experimented with unequal transmission ranges, by giving each node a random range uniformly distributed between $R+r$ and $R-r$. Figures 7(a) and (b) show the number of slots as a function of $(r/R)100$ (percent variability) for 2 values of R, for broadcast and link scheduling, respectively.

As expected, the number of slots used by each of RAND, MNF and PMNF, as well as the slots needed (i.e., the optimum) increases with both network size and node range. This is true for both broadcast and link scheduling. The increase appears to be more rapid with increasing range, especially for link scheduling. This is probably because adding more nodes results in a linear increase in a neighborhood (nodes per unit area) of a given node, whereas increasing the range results in a quadratic increase (πR^2 nodes per unit area).

PMNF performs best in practice, followed by MNF, and then RAND. On average, PMNF uses 9.6% fewer slots than RAND for broadcast schedules, and 8.2% fewer slots for link schedules. The difference tends to be larger for higher sizes and ranges. For $N = 500$, and $R = 60$, the performance improvement is as much as 12.9% and 10.7% for broadcast and link scheduling, respectively. MNF is somewhere in the middle for broadcast schedules, but performs nearly as well as PMNF for link schedules.

In terms of performance relative to optimum, broadcast scheduling using PMNF is within 1.113 of the optimum

[5] The problem being NP-complete, it is prohibitively time-consuming to compute the optimum number of slots. The lower bound for broadcast scheduling is ρ_{in} and is fairly tight, and for link scheduling is the maximal number of mutually constrained edges in the vicinity of the maximum degree vertex and is quite loose.

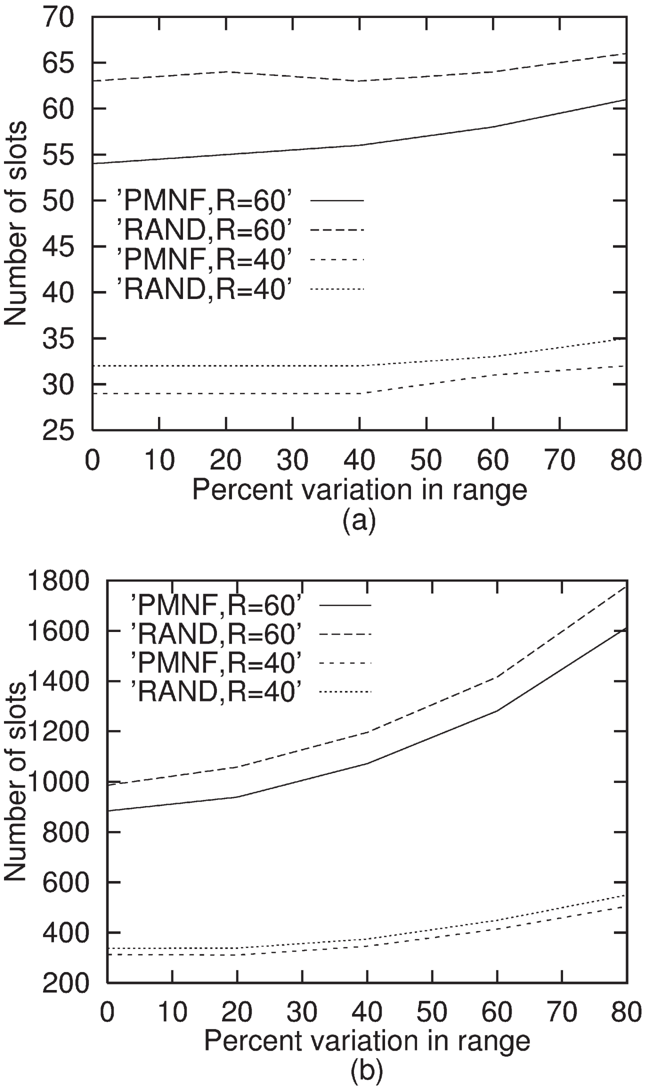

Figure 7. Performance of PMNF and RAND, versus range variability, with $N = 500$, for broadcast assignment (a) and link assignment (b). A higher range variability implies a higher fraction of unidirectional links.

even for the extreme case $N = 500$, $R = 60$, and in most typical cases, much less. The corresponding figure for link scheduling is 1.898 using the extremely loose lower bound on optimum. We expect the in-practice performance to be much better than this.

For both broadcast and link scheduling, the number of slots used by UxDMA for PMNF as well as RAND increases gradually with increasing variability of transmission range (figure 7). Note that increasing the variability in transmission range typically increases the fraction of unidirectional links. The increase in number of slots used seems to be more pronounced at higher ranges, especially for link scheduling. Although not shown in order to avoid cluttering the plot, the MNF and the optimum lower bounds show similar behavior. Thus, everything else being same, unidirectional links seem to consume more spectrum compared to bidirectional links.

6. Concluding remarks

We have developed a unified framework which captures a number of current and potential channel assignment problems. Solution to an assignment problem can be obtained simply by identifying the set of atomic constraints characterizing the problem, and using algorithm UxDMA. The user has a runtime choice, within the same algorithm/implementation, of three variants RAND, MNF and PMNF, each offering a different tradeoff between simplicity (running time), and performance. Our theoretical and experimental analyses are useful in predicting how the algorithm will perform for each problem. Our analysis shows that PMNF, although a little more complicated, and slightly more expensive in terms of running time, not only has a significantly improved worst-case bound in comparison to RAND, but also does 8–12% better in-practice. Since assignments are often made once and used repeatedly, the investment appears worthwhile, especially under scarce-bandwidth conditions.

Due to space constraints, we have not been able to describe distributed algorithms, which facilitate node mobility. We note, however, that the unification itself is completely orthogonal to obtaining a distributed version of UxDMA, which can be done by generalizing the specific-problem distributed algorithms of [21,28] for PMNF, and the schemes of [10,7,30] for RAND and [3] for MNF.

Acknowledgements

We are grateful to Sudha Kapali, Northeastern University, and Martha Steenstrup, BBN Technologies, for their valuable comments and suggestions on improving this paper. We also acknowledge the contribution of Prof. Errol Lloyd, University of Delaware, who supervised the dissertation upon which some of the results are based.

References

[1] N. Alon, A. Bar-Noy, N. Linial and D. Peleg, On the complexity of radio communication, in: *Proceedings of 21st Annual ACM Symposium on Theory of Computing* (1989).

[2] E. Arikan, Some complexity results about packet radio networks, IEEE Transactions on Information Theory 30 (July 1984) 910–198.

[3] A.A. Bertossi and M.A. Bonuccelli, Code assignment for hidden terminal interference avoidance in multihop packet radio networks, in: *Proceedings of INFOCOM* (1992).

[4] F. Box, A heuristic technique for assigning frequencies to mobile radio nets, IEEE Transactions on Vehicular Technology 27 (May 1978) 57–74.

[5] I. Chlamtac and A. Farago, Making transmission schedules immune to topology changes in multi-hop packet radio networks, IEEE/ACM Transactions on Networking 2(1) (February 1994) 23–29.

[6] I. Chlamtac and S. Kutten, A spatial reuse TDMA/FDMA for mobile multi-hop radio networks, in: *Proceedings of INFOCOM* (1985).

[7] I. Chlamtac and A. Lerner, Fair algorithms for maximal link activation in multihop radio networks, IEEE Transactions on Communications 35 (July 1987) 739–746.

[8] I. Cidon and M. Sidi, Distributed assignment algorithms for multihop radio networks, IEEE Transactions on Computers 38(10) (October 1989) 1353–1361.
[9] W. Diepstraten, G. Ennis and P. Berlanger, DFWMAC: Distributed Foundation Wireless Medium Access Control, IEEE Document P802.11-93/190 (November 1993).
[10] A. Ephremedis and T. Truong, Scheduling broadcasts in multihop radio networks, IEEE Transactions on Communications 38 (April 1990) 456–460.
[11] A. Ephremedis, J.E. Wieselthier and D.J. Baker, A design concept for reliable mobile radio networks with frequency hopping signalling, Proceedings of the IEEE 75(1) (January 1987) 56–73.
[12] S. Even, O. Goldreich, S. Moran and P. Tong, On the NP-completeness of certain network testing problems, Networks 14 (1984).
[13] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, New York, 1979).
[14] B. Hajek and G. Sasaki, Link scheduling in polynomial time, IEEE Transactions on Information Theory 34(5) (September 1988).
[15] W.K. Hale, Frequency assignment: Theory and applications, Proceedings of the IEEE 68(12) (December 1980).
[16] F. Harary, *Graph Theory* (Addison-Wesley, Reading, MA, 1972).
[17] L. Hu, Distributed code assignments for CDMA packet radio networks, in: *Proceedings of INFOCOM* (1991).
[18] J. Jubin and J.D. Tornow, The DARPA packet radio network protocols, Proceedings of the IEEE 75 (1987).
[19] R.S. Kahn, J. Gronemeyer, J. Burchfiel and R. Kunzelman, Advances in packet radio technology, Proceedings of the IEEE 66(11) (November 1978).
[20] G. Lauer, Packet-radio routing, in: *Routing in Communication Networks*, ed. M. Steenstrup (Prentice-Hall, Englewood Cliffs, NJ, 1995).
[21] E.L. Lloyd and S. Ramanathan, Efficient distributed algorithms for channel assignment in multihop radio networks, Journal of High Speed Networks 2(4) (1993).
[22] V.H. MacDonald, The cellular concept, The Bell Systems Technical Journal 44 (1965) 547–588.
[23] T. Makansi, Transmitter oriented code assignment for multihop packet radio, IEEE Transactions on Information Theory 34(5) (September 1988).
[24] A. Mansfield, Determining the thickness of graphs is NP-hard, Mathematical Proceedings of the Cambridge Philosophical Society 93 (1983) 9–23.
[25] T. Nishizeki and N. Chiba, *Planar Graphs: Theory and Algorithms*, Annals of Discrete Mathematics, Vol. 32 (North-Holland, Amsterdam, 1988).
[26] C.E. Perkins and P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: *Proceedings of ACM SIGCOMM*, 1994.
[27] M. Post, P. Sarachik and A. Kershenbaum, A biased greedy algorithm for scheduling multi-hop radio networks, in: *Proceedings of the Conference on Information Science Systems* (1985).
[28] S. Ramanathan, Scheduling algorithms for multihop radio networks, Ph.D. thesis, University of Delaware (1992).
[29] S. Ramanathan and E.L. Lloyd, Scheduling algorithms for multihop radio networks, IEEE/ACM Transactions on Networking 1(2) (April 1993).
[30] R. Ramaswami and K.K. Parhi, Distributed Scheduling of Broadcasts in a radio network, in: *Proceedings of IEEE INFOCOM* (1989).
[31] A. Sen and M.L. Huson, A new model for scheduling packet radio networks, in: *Proceedings of INFOCOM* (1996) pp. 1116–1124.
[32] N. Shacham and J. Westcott, Future directions in packet radio architectures and protocols, Proceedings of the IEEE 75(1) (January 1987) 83–99.
[33] J.A. Silvester, Perfect scheduling in multi-hop broadcast networks, in: *Proceedings of ICCC* (1982).

S. Ramanathan received his B.Tech degree from the Indian Institute of Technology, Madras, and his Ph.D. degree from the University of Delaware. He is a research scientist in the Department of Internetwork Research at BBN Technologies (a division of GTE Internetworking), where he is working on projects on self-organization and routing, both for wired and wireless ad hoc networks. His research interests are in the design and analysis of distributed algorithms, channel access, routing, personal communications networks, and graph theory.
E-mail: ramanath@bbn.com