Strong Edge Coloring for Channel Assignment in Wireless Radio Networks

Christopher L. Barrett† Gabriel Istrate* V. S. Anil Kumar†
Madhav V. Marathe† Shripad Thite‡ Sunil Thulasidasan*
*Basic and Applied Simulation Science (CCS-5), Los Alamos National Laboratory, USA; {istrate,sunil}@lanl.gov
†Virginia Bio-Informatics Institute and Dept. of Computer Science, Virginia Tech, USA; {cbarrett,akumar,marathe}@vt.edu
‡Department of Math & Computer Science, Technische Universiteit Eindhoven, The Netherlands; sthite@win.tue.nl

***Abstract*— We give efficient sequential and distributed approximation algorithms for strong edge coloring graphs modeling wireless networks. Strong edge coloring is equivalent to computing a conflict-free assignment of channels or frequencies to pairwise links between transceivers in the network.**

I. Introduction

A wireless radio network consists of a group of transceivers in space communicating with each other over a shared medium. Each transceiver has a range (a geographic region) within which it can communicate with other transceivers. Communication can take place between a pair of nodes that are both in range of each other via protocols that send data packets in one direction and acknowledgements in the other direction.

Assigning channels or frequencies to the links between transceivers to avoid *primary* and *secondary* interference corresponds to the *strong edge coloring* problem (also called *distance-2 edge coloring*) in the graph that models the radio network [1]–[3]. This undirected graph has one node for each transceiver; for every pair of transceivers such that each transceiver is in the range of the other, there is an edge between the corresponding nodes. In this graph, a valid strong edge coloring must assign distinct colors, corresponding to distinct channels, to any pair of edges between which there is a path of length at most two. Since each color corresponds to a channel, it is important to produce a strong edge coloring of the entire graph that uses a minimum number of colors or to maximize the number of edges colored in a *partial* coloring, i.e., a coloring of some subgraph. We consider the following two related problems.

Problem D2EC(G): Compute a strong edge coloring of a given graph $G = (V, E)$ with the fewest possible colors. Equivalently, compute an interference-free channel assignment with the fewest channels.

Problem D2EC(G,k): Maximize the number of edges colored in a partial strong edge coloring with at most k colors of a given graph $G = (V, E)$. Equivalently, compute a channel assignment with at most k channels that maximizes the pairs of transceivers that can communicate without interference.

Mahdian [4], [5] and Erickson *et al*. [6] proved that D2EC(G) is NP-complete. The hardness of D2EC(G) implies that problem D2EC(G,k) is also NP-complete. Hence, approximation algorithms are necessary.

The first unified coloring-based framework for resource allocation problems in channel assignment was described by Ramanathan [3], but most results [7], [8] consider node-based conflict models, i.e., vertex coloring problems. Our results extend those in [9], giving a better analysis, and solving the more general problem with multiple frequencies. The MAC layer, which is concerned with avoiding collisions due to signal interference, especially the 802.11 protocol, has been the topic of numerous papers, both theoretical [1] and experimental [10]–[12]. A good collision avoidance protocol in the MAC layer is necessary for efficient utilization of the bandwidth of the underlying medium [13]. Finding the exact number of colors required for a strong edge coloring of particular classes of graphs has been studied extensively [4], [5], [14]. We are concerned with efficient and provable algorithms to produce a strong edge coloring efficiently, even if it uses more than the optimum number of colors.

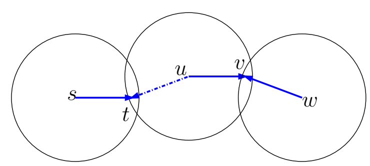

Fig. 1. Primary interference occurs at v because it receives signals from both u and w on the same channel. Secondary interference occurs at t because it receives signals from u intended for v on the same channel as from its own source s.

II. Approximation algorithms for D2EC(G)

A. For c-inductive graphs

A graph G is *c-inductive* if either G has at most c vertices or G has a vertex u of degree at most c such that $G - u$ is c-inductive. Equivalently, G is c-inductive if its vertices can be numbered so that at most c neighbors of any vertex v have higher numbers than v. Every graph G is c-inductive for some c in the range $\delta \leq c \leq \Delta$ where δ and Δ denote the minimum and maximum vertex degrees of G respectively. Trees are 1-inductive, outerplanar graphs are 2-inductive [15], and planar graphs are 5-inductive. By a simple application of Euler's formula [16], it can be seen that graphs of bounded genus are $O(1)$-inductive.

Greedy algorithm: Compute an inductive ordering of the vertices of G as follows. For each $i = 1, 2, \ldots$, let v_i be a vertex of degree at most c in $G - \{v_1, v_2, \ldots, v_{i-1}\}$. Such a vertex always exists because G is c-inductive. Consider the vertices in the reverse order, i.e., $v_n, v_{n-1}, \ldots, v_1$. At step i, color all previously uncolored edges incident on v_i (in arbitrary order) greedily: each uncolored edge is assigned the available color of smallest index.

Theorem 2.1: The greedy algorithm uses at most $(4c-3)\Delta - 2c + 3$ colors and runs in $O(n\Delta \log n)$ time.

Proof: Suppose the edge (v_i, v_j) where $j < i$ is being colored in step i. Recall that at the beginning of the ith stage only edges incident on vertices numbered greater than i have been colored. The edges within distance 2 of the edge (v_i, v_j) that have already been colored belong to two classes.

Edges incident on a neighbor of v_i. The vertex v_i has at most c neighbors among $\{v_{i+1}, v_{i+2}, \ldots, v_n\}$, and all the edges incident on these neighbors have been colored. This accounts for at most $c\Delta$ edges. In addition, v_i has at most $\Delta - 1$ neighbors other than v_j that are numbered smaller than i. Each of those neighbors has at most $c-1$ neighbors among $\{v_{i+1}, v_{i+2}, \ldots, v_n\}$ and these are the only edges already colored. This accounts for another $(\Delta-1)(c-1)$ edges at most. The total number of edges in this class is therefore at most $c\Delta + (c-1)(\Delta-1)$.

Edges incident on a neighbor of v_j. The vertex v_j has at most c neighbors among $\{v_{j+1}, v_{j+2}, \ldots, v_n\}$ and hence at most $c-1$ neighbors among $\{v_{i+1}, v_{i+2}, \ldots, v_n\}$, and all the edges incident on these neighbors have been colored. This accounts for at most $(c-1)\Delta$ edges. In addition, v_j has at most $\Delta - 1$ neighbors that are numbered smaller than j. Each of those neighbors has at most $c-1$ neighbors among $\{v_{i+1}, v_{i+2}, \ldots, v_n\}$ and these are the only edges already colored. This accounts for another $(\Delta-1)(c-1)$ edges at most. The total number of edges in this class is therefore at most $(c-1)\Delta + (c-1)(\Delta-1)$.

Thus, the total number of edges within distance 2 of the edge (v_i, v_j) that have been colored prior to the edge (v_i, v_j) itself is at most $4c\Delta - 2c - 3\Delta + 2$; our algorithm uses at most one color more. The inductive ordering can be computed in $O(n \log n)$ time along with an ordering of the edges using a heap. During the coloring, when edge (v_i, v_j) is being colored, one has to examine the colors of $O(\Delta)$ edges, assuming c is a constant. ■

Since every planar graph is 5-inductive, the algorithm yields a 17-approximation for planar graphs.

B. For graphs of bounded treewidth

A *tree decomposition* [7] of a graph $G(V, E)$ is a pair $(\{X_i | i \in I\}, T = (I, F))$ with $\{X_i | i \in I\}$ a family of subsets of V, one for each node of tree T such that (i) $\cup_i X_i = V$, (ii) $\forall (u, v) \in E$, there exists an $i \in I$ such that $u, v \in X_i$, and (ii) $\forall u \in V$, the set of all X_i containing u form a connected component in T. The *width* of such a decomposition is $\max_{i \in I} |X_i| - 1$ and the *treewidth* of a graph G is the minimum width over all possible tree decompositions of G.

Salavatipour [17] gives a polynomial-time algorithm to determine for a given s whether there exists a strong edge coloring with at most s colors of a graph with treewidth k; the algorithm can be modified to find a coloring with s colors if it exists. We give a faster, simpler approximation algorithm.

If G is a graph with treewidth W and maximum degree Δ, then it is easy to see that its *line graph* $L(G)$ has treewidth at most $W\Delta$. Hence, the treewidth of $L(G)^2$ is at most $(W+1)\Delta^2$. Therefore, if W, Δ are constants, then $L(G)^2$ also has constant treewidth. A strong edge coloring of G is equivalent to a proper vertex coloring of $L(G)^2$. Using results on coloring treewidth bounded graph [18], we have the following theorem.

Theorem 2.2: There exists a polynomial-time algorithm for D2EC(G) if G has bounded treewidth and bounded degree.

C. For disk graphs

We are given n transceivers located in the plane. Each transceiver u has a communication range which is a disk D_u. Two transceivers u and v can communicate if and only if $v \in D_u$ and $u \in D_v$. In other words, u and v can communicate if and only if each can receive signals (data and acknowledgment packets) from the other.

A *disk graph* $G = (V, E)$ is a directed graph with n vertices corresponding to the n transceivers and a directed edge $u \rightarrow v$ if and only if the disk D_u contains v. We want to color only bidirected edges (since communication can happen only on those). The other (unidirectional) edges contribute to the interference: so bidirected edges (u, v) and (u', v') can get the same color provided none of the edges $(u, u'), (u, v'), (v, u'), (v, v')$ or their reversals are present.

Let $d(u, v)$ denote the length of the shortest path between any two nodes u and v of G. For edge $e = (u, v)$, define $r(e) = r(u) + r(v)$. Let $N_{\geq} = \{e' \in N_2(e) : r(e') \geq r(e)\}$. For node v, let $N(v) = \{w \in V | w \in D(v)\}$. Equivalently $N(v) = \{w | d(v, w) \leq 1\}$. Let $N_2(v) = \{w | d(v, w) \leq 2\}$. For edge e, let $N_2(e) = \{e' | d(e, e') \leq 1\}$ and for $E' \subset E$, let $N_2(E') = \cup_{e \in E'} N_2(e)$. Let OPT denote the number of colors in an optimum strong edge coloring of G.

Greedy algorithm: First, order the edges in E' as $e_1, \ldots, e_m$, such that $r(e_1) \leq r(e_2) \leq \ldots \leq r(e_m)$. Then, color the edges in E' in the reverse order—for each edge e_j, color it with the smallest numbered unused color.

As a result of packing constraints, for any edge e, the number of edges in $N_{\geq}(e)$ that can be given the same color in an optimum solution is $O(1)$. We thus have the following theorem.

Theorem 2.3: The number of colors used by the above algorithm is $O(\text{OPT})$.

A disk graph is a *unit disk graph* if each disk has unit radius. A unit disk graph is undirected since $u \in D_v$ if and only if $v \in D_u$. Order the disks in increasing lexicographic order of their (y, x) coordinates. So, $u \prec v$ if and only if either (i) u is below v, or (ii) u and v are on the same horizontal line, and u is to the left of v. The ordering of disks induces a corresponding ordering of the edges. For each disk u, the edges (u, v) incident on u are considered in the same order as the order of the neighbors of u.

Algorithm UnitDisk: Greedily color every edge (u, v) of G with $v \prec u$ taken in order with the color of smallest index not yet used on any edge within distance 2 of the edge (u, v).

Theorem 2.4: Algorithm UnitDisk uses at most $8\,\textsc{Opt} + 1$ colors where $\textsc{Opt}$ is the number of colors in an optimum strong edge coloring of G.

Proof: Consider the edge $e = (u, v)$ incident on u currently being assigned a color by algorithm UnitDisk; we have $v \prec u$. Any edge e' within distance 2 of (u, v) must have at least one endpoint w in $D_u \cup D_v$ and both endpoints must precede u in the lexicographic ordering. The region $D_u \cup D_v$ can be partitioned into at most 10 sectors of unit diameter (Figure 2). In fact, if $e' = (w, z)$ has been colored before the edge $e = (u, v)$ then neither w nor z can belong to the subset of $D_u \cup D_v$ above u. Therefore, neither w nor z can belong to any one of the 2 sectors of $D_u \setminus D_v$ above the horizontal line through u. (Since $v \prec u$, node v must lie on or below the horizontal line through u.) For any two nodes a and b in one of the remaining 8 sectors, it is the case that both $b \in D_a$ and $a \in D_b$, and so all nodes in any one sector form a clique in G.

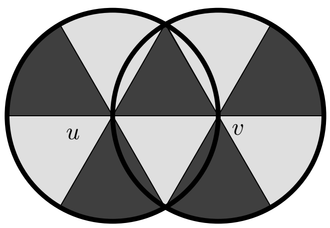

Fig. 2. The union $D_u \cup D_v$ of two disks of unit radius can be partitioned into 10 subsets of unit diameter

If algorithm UnitDisk chooses color $k + 1$ for the edge e it must be the case that k distinct colors already appear on the edges within distance 2 of e. Since at least $k/8$ colors are necessary to colors these edges, the algorithm uses at most $8\,\textsc{Opt} + 1$ colors. ■

III. Distributed algorithms for D2EC(G,k)

We give distributed algorithms in a synchronized radio broadcast model of computation, running in a logarithmic number of rounds and producing an $O(1)$-approximate solution to the D2EC(G,k) problem for disk graphs. We show that the distributed algorithm of Balakrishnan *et al.* [9] runs in $O(\rho \log n)$ rounds. Here, ρ denotes the time it takes to compute the *active degree* (defined later) of each node v in a radio broadcast model. Using this result we devise an $O(1)$-approximate solution in $O(\log n \log R)$ rounds for disk graphs, where R is the ratio of largest to smallest radii. Using the above algorithms as subroutines, we design distributed algorithms with $O(1)$ performance for the D2EC(G,k) problem with additional multiplicative factor of k in the number of rounds for both unit disk and general disk graphs. The techniques developed in order to obtain the desired bounds exploit the geometric nature of wireless networks; these techniques might be of independent interest. The algorithms can be extended to some of the variants considered by Ramanathan [3].

In this model, transmission by a node is always a broadcast: all nodes within the transmission range will (simultaneously) receive the signal. In each round a node can either transmit, i.e., be *active*, or keep silent. A node v receives a message from a node w in a given round if and only if v keeps silent and w is the only neighbor to transmit. If more than one neighbor of v transmits in a given round, there is a collision, and the receiving node v hears only noise. We will assume that nodes can determine whether there is a collision or not. For a node v, let $\deg(v)$ denote its *active degree*, i.e., the number of neighbors of v that are active in the current round. Most of our algorithms assume synchronous computing: this will require periodic transmission of synchronization messages. For one of our algorithms, we will not assume any synchronization.

Each node has a unique ID and is aware of the size of its neighborhood, but not necessarily the IDs of neighboring nodes. For general disk graphs, we also need to know the minimum and maximum broadcast ranges. All our algorithms are randomized, and the guarantees on the running time, i.e., number of rounds, hold with high probability (arbitrarily close to 1).

We count separately the time that is required to compute the active degree of each node in the radio broadcast model. We assume that, at the beginning of each round of our algorithm, every vertex computes its active degree in a distributed fashion in ρ rounds.

The edges assigned any one color form a *strong matching*. Let D2EMIS(G) denote the problem of finding a strong matching of maximum cardinality in the unit disk graph G. We will repeatedly use a distributed algorithm for the D2EMIS problem as a subroutine for computing a strong edge coloring of G with k colors. Algorithm D2M-Dist-UnitDisk of Figure 3 is executed by each node that wants to reserve the channel; this is the same algorithm as in [9].

Theorem 3.1: Algorithm D2M-Dist-UnitDisk terminates in expected $O(\rho \log n)$ rounds and produces an $O(1)$-approximate solution to D2EMIS(G) for unit disk graph G.

Due to lack of space, we only give proof sketches. For a node v, $\deg(v)$ is the number of neighbors of v that are still active because their $\hat{b}()$ value is still set to -1. Given a square region A of size $\ell \times \ell$, we will also denote the set of points in it by A. Let $S(A) = \{v \in A : \deg(v) \leq |A|\}$. A region A that satisfies $|S(A)| \geq \gamma |A|$ for some constant γ is said to be *good*. In a unit disk graph, for any edge e, there can be at most $O(1)$ edges in $N_2(e)$ in any D2-matching [7], [9]. This fact implies that only a constant number of edges can be chosen in the neighborhood of an edge in any strong matching as a result of packing constraints. Let $D_2(v)$ denote the disk of radius 2 centered at v.

Lemma 3.2: Let $\mathcal{E}(v)$ denote the event that $b(v) = 1$ and $\forall w \in D_2(v), w \neq v, b(w) = 0$. Then, $\Pr[\mathcal{E}(v)] \geq \epsilon/(\deg(v)+1)$, for a constant $\epsilon > 0$.

Proof: Because of the independent trials, $\Pr[\mathcal{E}(v)] = \frac{1}{(\deg(v)+1)}\Pi_{w\in D_2(v),w\neq v}(1-\frac{1}{\deg(w)+1})$. Partition $D_2(v)$ into a constant number of regions $R_1,\ldots,R_s$, such that the diameter of each R_i is at most $\frac{1}{2}$; this can be done in any arbitrary manner, as long as s is a constant. Then, for each R_i and $\forall w \in R_i$, $\deg(w) \geq |R_i|$. Hence,

$$\begin{aligned}
\Pr[\mathcal{E}(v)] &\geq \frac{1}{\deg(v)+1}\prod_{i=1}^{s}\prod_{w\in R_i}\left(1-\frac{1}{\deg(w)+1}\right)\\
&\geq \frac{1}{\deg(v)+1}\prod_{i=1}^{s}\prod_{w\in R_i}\left(1-\frac{1}{|R_i|}\right)\\
&= \frac{1}{\deg(v)+1}\prod_{i=1}^{s}\left(1-\frac{1}{|R_i|}\right)^{|R_i|}\\
&\geq \frac{1}{\deg(v)+1}(\epsilon' e)^s \quad = \frac{\epsilon}{(\deg(v)+1)}
\end{aligned}$$

■

Lemma 3.3: Let A be a 1×1 square region. Consider some phase i of the algorithm in which points in A are still participating. Let $\mathcal{E}(v)$ denote the event that node v sets $\hat{b}(v)=1$. Let $\mathcal{E}(A)$ denote the event $\cup_{v\in A}\mathcal{E}(v)$. Then, $Pr[\mathcal{E}(A)] \geq \epsilon$, where ϵ is a constant if A is good.

Proof: The events $\mathcal{E}(v),\ v\in A$ are mutually disjoint, because $\hat{b}(v)=1$ implies all nodes $w\in D_2(v)$ set $\hat{b}(w)=0$ and $A\subset D_2(v)$. Therefore, $\Pr[\mathcal{E}(A)] \geq \sum_{v\in A}\Pr[\mathcal{E}(v)] \geq \sum_{v\in A}\frac{\epsilon}{\deg(v)+1} \geq \sum_{w\in S(A)}\frac{\epsilon}{|A|+1} = \frac{\epsilon'|S(A)|}{|A|+1} \geq \epsilon$. The second inequality follows from Lemma 3.2, and the last inequality follows since ϵ',γ are constants and A is good. ■

Any good grid cell A in $\mathcal{T}_1^{(0,0)}$ will lose some points with constant probability. Our algorithm would be making good progress in each round, if a large fraction of the cells are good. It need not be true that a constant fractions of the cells in $\mathcal{T}_1^{(0,0)}$ will be good; but, if we perturb the grid $\mathcal{T}_1^{(0,0)}$ along either axis by $\frac{1}{2}$, and consider all the cells in these grids together, a constant fraction of these cells will be good.

Lemma 3.4: In any phase i, let $\mathcal{S}$ be the set of all grid cells in $\mathcal{T}_1^{(0,0)}$, $\mathcal{T}_1^{(0,1/2)}$, $\mathcal{T}_1^{(1/2,0)}$ and $\mathcal{T}_1^{(1/2,1/2)}$ that contain at least one point. Then a constant fraction of the cells in $\mathcal{S}$ are good.

Proof: Consider any point v that still does not have $\hat{b}(v) = -1$ in the current phase. There is a unique cell containing this point in any grid $\mathcal{T}_1^{(p,q)}$; so there are four cells, say A_1, A_2, A_3, A_4, in $\mathcal{S}$ that contain v. $D(v)$ is completely contained in the union of these four cells, and therefore for at least one of these cells A_i, it is true that $|N(v)\cap A_i| \geq \deg(v)/4$, and so $\deg(v)\leq 4|A_i|$.

Partition the bounding box of the points into $c\times c$ sized squares (for a constant c) and consider one such square B. There are a constant number of cells in $\mathcal{S}$ that overlap with B; let $\mathcal{S}(B)$ denote the set of such cells. Let $V(B)$ denote the set of points in B. Now, consider a bipartite graph $H(V(B),\mathcal{S}(B),E')$: one set of vertices is $V(B)$ and the other is the set $\mathcal{S}(B)$; there is an edge (v,A) between $v\in V(B)$ and $A\in\mathcal{S}(B)$ if and only if $v\in A$. Clearly, $deg_H(v)=4$ for each $v\in V(B)$. By the argument in the previous paragraph, for each v there is an A such that $(v,A)\in E'$ and $\deg(v)\leq 4|A|$; color such an edge (v,A) red. Then, at least a quarter of all the edges in E' are red. Hence, at least one cell A in $\mathcal{S}(B)$ must have the property that a quarter of the edges incident on A are colored red; such a cell A satisfies $|S(A)|\geq |A|/4$. Each such constant-sized part B has at least one good cell A. Since there are only a constant number of cells in $\mathcal{S}(B)$, a constant fraction of the cells in $\mathcal{S}$ are good. ■

Proof: [Theorem 3.1] The size of the D2EMIS found by algorithm D2M-DIST-UNITDISK is within a constant factor of the optimal, since only a constant number of edges can be chosen in the neighborhood of an edge in any strong matching [9]. For Phase 1, we will show that the expected number of rounds is $O(\rho\log n)$, in contrast with the analysis in [9]. For Phase 2, we still use the analysis from [9], which proves $O(1)$ rounds.

Let $\mathcal{S}$ be as defined in Lemma 3.4: the union of all the grid cells in $\mathcal{T}_1^{(p,q)}, (p,q) = (0,0),(0,\frac{1}{2}),(\frac{1}{2},0),(\frac{1}{2},\frac{1}{2})$ that contain at least one point. The Lemma implies that at least a constant fraction of the (non-empty) cells in $\mathcal{S}$ are good; let α be this constant fraction. By Lemma 3.3, for each good cell, event $\mathcal{E}(A)$, which denotes the event that some point from A is picked in phase 1, holds with probability at least ϵ. If event $\mathcal{E}(A)$ holds, all the nodes in A will stop participating in subsequent rounds, and so A will be removed from $\mathcal{S}$.

To compute wake-up probability in each round we require that each node compute its degree. By assumption this task takes ρ inner rounds. Let $\mathcal{S}_i$ denote the subset of cells that still contain (active) points after step i and let X_i denote the number of points that become inactive in step i, i.e. their $\hat{b}()$ value is set to 0 or 1. By the argument in the previous paragraph, $E[X_i]\geq \epsilon\alpha|\mathcal{S}|$, and $\mathcal{S}_{i+1} = \mathcal{S}_i \setminus X_i$. It follows that $E[|\mathcal{S}_{i+1}|] \leq (1-\epsilon\alpha)E[|\mathcal{S}_i|]$, and by standard arguments (for instance, as in [19]), it follows that the expected number of rounds is $O(\rho\log n)$. ■

For general disk graphs, we again use the algorithm from [9]. Assume without loss of generality that the minimum radius is 1 and the maximum radius is R. The algorithm in [9] partitions the edges into $O(\log R)$ subsets $E_i = \{e : r(e)\in[2^{i-1}, 2^i - 1]\}$, and considers each subset E_i in a separate round. See Figure 4. Our improved analysis gives us the following result.

Theorem 3.5: Algorithm D2M-DIST-DISK runs in $O(\rho\log n\log R)$ time and gives an $O(1)$-approximate solution to D2EMIS(G) for general disk graph G, where ρ denotes the time to compute the active degree of each node in the radio broadcast model.

IV. CONCLUSION

We gave efficient sequential and distributed approximation algorithms for strong edge coloring of graphs modeling wireless radio networks. Our algorithms can be used to compute an interference-free channel assignment for pairwise links in the network either by a global omniscient entity or by a local distributed computation at each node.

Phase 1: This phase consists of the following three steps, constituting one round. These steps are repeated until $\hat{b}(v) \in \{0,1\}$, for each node v. Only nodes v with $\hat{b}(v) = -1$ participate in these steps. Initially, each node v has $\hat{b}(v) = -1$. Every node v will run the following protocol if none of its neighbors is currently transmitting.

1) Initially $\forall v,\ b(v) = 0$. Each node v sets $b(v) = 1$ with probability $1/(\deg(v)+1)$ (wake up probability). If $b(v) = 1$ then v sends an RTS.
2) If any node w hears a collision, it sends a COLLISION signal.
3) If $b(v) = 1$ and v hears no COLLISION or RTS signal(s) from any other nodes in $D(v)$, it sends an RTS-SUCCESSFUL signal and sets $\hat{b}(v) = 1$.
4) If v hears an RTS-SUCCESSFUL signal from some node in $D(v)$, it sets $\hat{b}(v) = 0$ and retransmits an RTS-SUCCESSFUL signal.
5) If any node w hears an RTS-SUCCESSFUL signal or a collision due to multiple such signals, it sets $\hat{b}(w) = 0$.

Phase 2: Let $S = \{v|\hat{b}(v) = 1\}$. Note that S forms a distance-2 independent set. In this phase, for each node $v \in S' \subseteq S$, we choose a node $m(v) \in D(v)$ such that $(v, m(v))$ form a D2-matching.
Node v maintains a variable $\hat{c}(v)$, initially $\forall v,\ \hat{c}(v) = -1$. For each $v \in S$, the pair v and $m(v)$ work together in the following steps constituting one round. The steps are repeated until $\forall v \in S,\ \hat{c}(v) \in \{0,1\}$.

1) Node v keeps a random variable $c(v)$, initially $c(v) = 0$. Each v sets $c(v) = 1$ with constant probability α. If $c(v) = 1$, then v and broadcasts an RTS1 signal.
2) For each v, if node $m(v)$ hears an RTS1 signal it rebroadcasts it and sets $c(m(v)) = 1$.
3) If node v or $m(v)$ sees a collision, due to multiple transmissions, it sends a COLLISION signal.
4) If $c(v) = c(m(v)) = 1$ and v and $m(v)$ do not hear COLLISION signal, they both send a RTS1-SUCCESSFUL signal, and set $\hat{c}(v) = 1$.
5) Any node v that hears a RTS1-SUCCESSFUL signal sets $\hat{c}(v) = 0$.

Fig. 3. Distributed algorithm D2M-DIST-UNITDISK for D2EMIS(G) for a unit disk graph G

1) If $\forall w \in N(v), v \notin N(w)$ holds for node v, then it does not participate in the following steps.
2) Round i comprises of the following steps. Perform these rounds for each $i = 0, \ldots, \log R + 1$.
   a) All nodes v with $2^{i-1} < r(v) \leq 2^i$ that have $w \in N(v)$ with $v \in N(w), r(w) \leq 2^i$ participate. Such nodes run Step 2 of algorithm D2M-DIST-UNITDISK and choose a distance-2 independent set S.
   b) For each node $v \in S$, v chooses node $m(v) \in N(v)$ such that (i) $v \in N(m(v))$, and (ii) $r(v) + r(m(v)) \leq 2^{i+1}$.
   c) The nodes $v, m(v), \forall v \in S$ run Step 4 of algorithm D2M-DIST-UNITDISK and choose an set of edges to add to existing D2EMIS. All nodes within distance 1 of the chosen edges do not participate any further.

Fig. 4. Distributed algorithm D2M-DIST-DISK for D2EMIS(G) for a general disk graph G

ACKNOWLEDGMENTS

Part of the research by Christopher L. Barrett, V.S. Anil Kumar, Madhav V. Marathe was done while at the Los Alamos National Laboratory. Research by Shripad Thite was done while visiting the Los Alamos National Laboratory and at the University of Illinois at Urbana-Champaign. We thank the anonymous referees for their suggestions for improvement.

REFERENCES

[1] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC layer fairness in wireless packet networks," in *Proc. 6th ACM Conf. on Mobile Computing and Networking*, August 2000, pp. 87–98.
[2] S. Ramanathan and E. L. Lloyd, "Scheduling algorithms for multi-hop radio networks," in *IEEE/ACM Trans. Networking*, vol. 2, 1993, pp. 166–177.
[3] S. Ramanathan, "A unified framework and algorithm for (T/F/C) DMA channel assignment in wireless networks," in *Proc. IEEE INFOCOM'97*, 1997, pp. 900–907.
[4] M. Mahdian, "The strong chromatic index of graphs," Master's thesis, Department of Computer Science, University of Toronto, 2000.
[5] ——, "On the computational complexity of strong edge coloring," *Discrete Applied Mathematics*, vol. 118, pp. 239–248, 2002.
[6] J. Erickson, S. Thite, and D. Bunde, "Distance-2 edge coloring is NP-complete," 2002, arXiv:cs.DM/0509100. http://arxiv.org/abs/cs/0509100/
[7] S. O. Krumke, M. V. Marathe, and S. S. Ravi, "Models and approximation algorithms for channel assignment in radio networks," *Wireless Networks*, vol. 7, pp. 575–584, 2001.
[8] A. Gräf, M. Stumpf, and G. Weißenfels, "On coloring unit disk graphs," *Algorithmica*, vol. 20, pp. 277–293, 1998.
[9] H. Balakrishnan, C. L. Barrett, V. S. A. Kumar, M. V. Marathe, and S. Thite, "The distance-2 matching problem and its relationship to the MAC-layer capacity of ad hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1069–1079, 2004.
[10] T. Ozugur, M. Naghshineh, P. Kermani, C. Olsen, B. Rezvani, and J. A. Copeland, "Balanced media access methods for wireless networks," in *Proc. ACM MOBICOM*, 1998.
[11] V. Bhargavan, S. Demers, S. Shenker, and L. Zhang, "Macaw: A media access protocol for wireless LANs," in *Proc. ACM SIGCOMM*, 1994.
[12] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Proc. ACM MOBICOMM*, 2001.
[13] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.
[14] R. Faudree, A. Gyarfas, R. Shelp, and Z. Tuza, "The strong chromatic index of graphs," *Ars Combinatoria*, vol. 29B, pp. 205–211, 1990.
[15] A. Proskurowski and M. M. Syslo, "Efficient vertex and edge coloring of outerplanar graphs," *SIAM J. Algebraic and Discrete Methods*, vol. 7, pp. 131–136, 1986.
[16] R. J. Wilson, *Introduction to Graph Theory*, 2nd ed. Longman Publishers, UK, 1979.
[17] M. R. Salavatipour, "A polynomial time algorithm for strong edge coloring of partial k-trees," *Discrete Applied Math*, vol. 143, pp. 285–291, 2004.
[18] H. L. Bodlaender, "A tourist guide through treewidth," Utrecht University, Tech. Rep. RUU-CS-92-12, 1992.
[19] M. Luby, "A simple parallel algorithm for the maximal independent set problem," *SIAM J. Computing*, vol. 15, no. 4, pp. 254–278, 1986.