A Message-Passing Algorithm for Wireless Network Scheduling *

Ioannis Ch. Paschalidis,† *Fellow, IEEE,* Fuzhuo Huang,‡ Wei Lai§

***Abstract*—We consider scheduling in wireless networks and formulate it as Maximum Weighted Independent Set (MWIS) problem on a "conflict" graph that captures interference among simultaneous transmissions. We propose a novel, low-complexity, and fully distributed algorithm that yields high-quality feasible solutions. Our proposed algorithm consists of two phases, each of which requires only local information and is based on message-passing. The first phase solves a relaxation of the MWIS problem using a gradient projection method. The relaxation we consider is tighter than the simple linear programming relaxation and incorporates constraints on all cliques in the graph. The second phase of the algorithm starts from the solution of the relaxation and constructs a feasible solution to the MWIS problem. We show that our algorithm always outputs an optimal solution to the MWIS problem for perfect graphs. Simulation results compare our policies against Carrier Sense Multiple Access (CSMA) and other alternatives and show excellent performance.**

***Index Terms*—Wireless networks, scheduling, maximum weighted independent set problem, message-passing, graph theory, distributed algorithms.**

I. Introduction

In a wireless network, the receiver of some data packet is able to successfully decode it only if the Signal to Noise and Interference Ratio (SNIR) during the transmission is above a certain level. Concurrent transmissions can cause interference to each other, thus, preventing packets from being successfully delivered to their respective receivers. A scheduling policy has as its main responsibility to eliminate such transmission "collisions." To that end, the seminal work in [1] proposes a scheduling policy that solves a *Maximum Weighted Independent Set (MWIS)* problem with appropriate weights to decide which links of the wireless network should be allowed to transmit at any point in time. This formulation is the starting point of our work in this paper.

* Research partially supported by the NSF under grants CNS-1239021 and IIS-1237022, by the ARO under grants W911NF-11-1-0227 and W911NF-12-1-0390, by the ONR under grant N00014-10-1-0952, and by the NIH/NIGMS under grant GM093147.

† Department of Electrical and Computer Engineering, and Division of Systems Engineering, Boston University, 8 St. Mary's St., Boston, MA 02215, yannisp@bu.edu, http://ionia.bu.edu/.

‡ Center for Information and Systems Engineering, Boston University, Boston, MA 02215, huangfz@bu.edu.

§ Center for Information and Systems Engineering, Boston University, Boston, MA 02215, wei@alum.bu.edu.

To make matters concrete, let us model the wireless network as a directed graph $\mathscr{W} = (\mathscr{D}, \mathscr{L})$, where $\mathscr{D}$ is the set of wireless devices, and $\mathscr{L}$ is the set of links in the network. For convenience we assume there are N links; let $\mathscr{L} = \{1, \ldots, N\}$. For each device $u \in \mathscr{D}$, if it can successfully transmit to another device $v \in \mathscr{D}$ in the absence of interference from other devices, we say that the ordered pair (u, v) is a link of the graph, i.e., $(u, v) \in \mathscr{L}$. For each link $i \in \mathscr{L}$ we denote by $\mathscr{N}_i$ the interference set of link i, namely, the set of all the links over any of which a simultaneous transmission will render the data delivery over link i unsuccessful. We assume the interference is symmetric, that is, for any links $i, j \in \mathscr{L}$, if $i \in \mathscr{N}_j$ then we also have $j \in \mathscr{N}_i$. Note that in our model two links do not have to share any nodes for them to be in each other's interference set, and this enables us to handle a variety of interference models.

Assume that time t is discrete. For each link $i \in \mathscr{L}$, $A_i^{(t)}$ denotes the number of packets arriving at link i during time slot t. Without loss of generality, we assume that all packets have equal lengths. We further assume that for any link $i \in \mathscr{L}$, $\{A_i^{(t)}\}$ is an independent and identically distributed process over time t, but it may be dependent on the arrival processes on other links. We also assume that the arrival processes across links have bounded covariances, that is, $\mathrm{Cov}(A_i^{(t)}, A_j^{(t)}) < \infty$, for any $i, j \in \mathscr{L}$. Let $\lambda_i < \infty$ be the expected number of packets arriving at link i during each time slot, namely, $\lambda_i = E[A_i^{(t)}]$, $\forall i \in \mathscr{L}$ and $\forall t$. For any link $i \in \mathscr{L}$, whenever the packets backlogged at link i are scheduled to be transmitted to the receiver, they are delivered with a constant transmission rate μ_i ($\mu_i > 0$) per time slot if link i is free of interference during the transmission. Let $X_i^{(t)} \in \{0, 1\}$ be the indicator of whether or not link $i \in \mathscr{L}$ is scheduled during any time slot t. Define the state of the system as the queue length $Q_i^{(t)}$ for link i at the end of time slot t; the system dynamics are written as

$$Q_i^{(t)} = \lceil Q_i^{(t-1)} + A_i^{(t)} - \mu_i X_i^{(t)} \rceil_+, \qquad (1)$$

for all $i \in \mathscr{L}$ and t, where $\lceil x \rceil_+ \stackrel{\triangle}{=} \max\{\lceil x \rceil, 0\}$, that is, the maximum of zero and the minimum integer that is no less than x. Let $\mathbf{Q}^{(t)} \triangleq (Q_1^{(t)}, \ldots, Q_N^{(t)})$. We note that for each i, $Q_i^{(t)}$, $A_i^{(t)}$, and $X_i^{(t)}$ take integer values only whereas μ_i can take real values.

We are interested in distributed and efficient scheduling policies so that $\mathbf{Q}^{(t)}$ is stable, namely, the long-term average backlog in the network is finite:

$$\lim\sup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N} E[Q_i^{(t)}] < \infty. \qquad (2)$$

Further, we are also interested in stable policies that minimize the above quantity.

To formulate the scheduling problem, we construct an undirected graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ at the beginning of each time slot t as follows. Let the set of nodes $\mathscr{V}$ be $\mathscr{L}$, namely, the set of links in the network. For any two nodes i and $j \in \mathscr{V}$, the unordered pair (i, j) is in $\mathscr{E}$ if and only if $i \in \mathscr{N}_j$, or equivalently, $j \in \mathscr{N}_i$. Each node $i \in \mathscr{V}$ is assigned a weight $w_i^{(t)} \triangleq Q_i^{(t-1)}/\mu_i$. We will call graph $\mathscr{G}$ the *conflict graph* corresponding to network $\mathscr{W}$ since each node in $\mathscr{V}$ represents a potential transmission and the edges in $\mathscr{E}$ indicate the conflicts between interfering transmissions. Selecting an interference-free schedule now amounts to finding an independent set in $\mathscr{G}$, that is, select a set of nodes so that no two selected nodes are connected by an edge (i.e., have a conflict). Because we are interested in ensuring stability and minimizing average delay, we seek to solve the MWIS problem on graph $\mathscr{G}$, that is, select the heaviest independent set of nodes (with the weights specified above).

The MWIS on general graphs, however, is NP-hard. Our objective, therefore, is to find a low-complexity algorithm that yields "good-quality" feasible solutions (i.e., interference-free). Our main contribution in this paper is the development of a novel algorithm. The algorithm consists of two phases. In the first phase we solve a relaxation of the MWIS using a gradient projection method. We show that the algorithm converges and does so in pseudo-polynomial time in the size of the input. In the second phase we leverage the relaxed solution and construct a feasible solution to the MWIS problem, using either a greedy algorithm or a more elegant estimation based on graph coloring. As we will see both phases can be implemented in a distributed fashion and require only local information exchanges between the wireless nodes. We show that the algorithm finds an optimal MWIS solution if the graph $\mathscr{G}$ is *perfect* [21]. [1] Our result generalizes earlier results concerning bipartite graphs ([2]) to a much larger class of graphs. This is useful since we show that there exist typical wireless networks with a perfect conflict graph, e.g., tree networks with a node-exclusive interference model. Further, we also show that the proposed algorithm guarantees the stability of the underlying wireless network.

We devise a practical way of using our algorithm to schedule wireless networks and use simulation to compare its performance with Carrier Sense Multiple Access (CSMA) and a greedy alternative. The numerical results indicate that our algorithm achieves significantly smaller aggregate long-run average queue length than CSMA and outperforms the greedy policy (substantially in some instances).

The rest of this paper is organized as follows. Sec. II surveys the related literature. Background material on notions from graph theory and the two relaxations of MWIS are presented in Sec. III. The gradient projection method for solving the MWIS relaxation is in Sec. IV. The second (estimation) phase is in Sec. V. A practical implementation is presented in Sec. VI where we also compare against two CSMA variants and a greedy policy. We offer concluding remarks in Sec. VII and all the proofs are in Sec. VIII.

Notational Conventions: Throughout the paper all vectors are assumed to be column vectors. We use lower case boldface letters to denote vectors and for economy of space we write $\mathbf{x} = (x_1, \dots, x_R)$ for the column vector $\mathbf{x}$. $\mathbf{x}'$ denotes the transpose of $\mathbf{x}$ and $\mathbf{0}$ the vector of all zeroes. By $||\mathbf{x}||_1$ and $||\mathbf{x}||_2$ we denote the ℓ_1 and ℓ_2 norms of $\mathbf{x}$, respectively. We use upper case boldface letters to denote matrices. We use script letters to define sets, and denote by $|\mathscr{A}|$ the cardinality of set $\mathscr{A}$.

II. Related Work

There are two schools of thought in allocating the wireless channel to interfering links that compete for transmission: scheduling and random multiple access. As mentioned earlier, applying [1] to the wireless network model we described yields an MWIS-based scheduling policy where the link weights are equal to the corresponding expected backlog. [1] shows that such a policy is throughput optimal, that is, it stabilizes any arrival vector $\boldsymbol{\lambda} = (\lambda_i;\ \forall i \in \mathscr{L})$ in the maximum throughput region – the set of $\boldsymbol{\lambda}$'s for which the network is stabilized under some policy. Subsequent work (see, e.g., [3–7]) has devised a host of policies that differ in their complexity and the fraction of the maximum throughput region they can achieve.

Simpler policies have also been considered in the literature. Empirical evidence has shown (and this is confirmed in our results) that a greedy policy can be effective. [8] shows that if the network satisfies some local-pooling property, then the greedy policy is indeed efficient.

In the random multiple access camp, several CSMA-based algorithms have been proposed [9–11]. These algorithms are distributed, low-complexity, and easy to

[1] In perfect graphs the chromatic number is equal to the size of the maximum clique and this remains true for all induced subgraphs as well.

implement. They use a random access scheme according to which a link seizes the channel with a probability that increases exponentially with a certain link-dependent weight (e.g., the expected backlog at the link). As explained in [12], these schemes sample over independent sets of $\mathscr{G}$ and converge to the optimal MWIS in graphs with large weights. This has led to showing that, like the original MWIS policy of [1], they are throughput optimal. In fact, the link weights do not have to be equal to the expected backlog; more general functions can lead to the same throughput optimality [13, 14]. Throughput optimality, however, does not imply small average backlog. The CSMA-type policies converge to the optimal MWIS when the backlog becomes large, and even then convergence is slow (the mixing time of the "sampling" Markov chain can be exponential in the size of the graph).

In this work we attempt to "solve" the MWIS problem using a different strategy: solve a relaxation and then construct a feasible solution. A relaxation strategy was also used in [15] but for a different throughput maximization problem in wireless sensor networks. [15], which followed earlier work in [16], developed a maximum weighted matching relaxation which is looser than the one we develop in this paper. The difficulty of MWIS has motivated substantial work on approximations, heuristics, and special cases [17, 18]. A polylogarithmic time algorithm is proposed in [19] that gives an almost exact approximation for planar graphs, where the idea is to decompose the entire graph into clusters. In [20], a randomized algorithm is developed that produces the maximal independent set with high probability for growth-bounded graphs including unit disk graphs.

A message passing algorithm for producing a feasible solution to the MWIS was proposed in [2]. As in our work, [2] solves a relaxation of the MWIS but one that is looser than ours. It shows that the solution obtained is optimal for bipartite graphs on which the MWIS has a unique solution. Our work generalizes those results in two ways: (a) by solving a tighter relaxation we can show optimality for perfect graphs which subsume bipartite graphs, and (b) for bipartite graphs we can relax the assumption on the uniqueness of the solution imposed in [2].

III. Background Material and MWIS relaxations

In this section we review some relevant background material from graph theory and provide two relaxations of the MWIS.

Let us consider an undirected graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ where $\mathscr{V} = \{1, \dots, N\}$ is the set of all the nodes and $\mathscr{E}$ is the edge set.

A *clique* is a set of vertices such that any two vertices in the set are adjacent. The maximum size of a clique in $\mathscr{G}$ is called the *clique number* of $\mathscr{G}$. A *maximal clique* $\mathscr{C}$ of graph $\mathscr{G}$ is a clique of $\mathscr{G}$ which can not be expanded by adding a vertex. Since maximal cliques contain all the cliques, we only consider maximal cliques in the rest of paper. We let $\mathscr{S} = \{\mathscr{C}_1, \mathscr{C}_2, \dots\}$ denote the set of all maximal cliques of $\mathscr{G}$.

An *independent (or stable) set* $\mathscr{I}$ is a set of vertices any two of which are nonadjacent. The *complement graph* $\bar{\mathscr{G}}$ of $\mathscr{G}$ is a graph with the same node set $\mathscr{V}$ and edge set $\bar{\mathscr{E}}$ which contains all edges not in $\mathscr{E}$. It follows that a clique in $\mathscr{G}$ is an independent set in $\bar{\mathscr{G}}$. Thus, every term defined for cliques in $\mathscr{G}$ has a corresponding term for independent sets in $\bar{\mathscr{G}}$. In particular, a *maximal independent set* is a set of vertices that can not be expanded without violating the independence property.

A *coloring* of $\mathscr{G}$ is a partition of $\mathscr{V}$ into independent sets $\mathscr{I}_1, \dots, \mathscr{I}_k$. The independent sets $\mathscr{I}_1, \dots, \mathscr{I}_k$ are called the *colors* of the coloring. The minimum number of colors in a coloring of $\mathscr{G}$ is called the *coloring (or chromatic) number* of $\mathscr{G}$. Determining the clique number and the coloring number are NP-complete problems [21].

The incidence vector $\mathbf{x}_{\mathscr{T}}$ of a vertex set $\mathscr{T} \subseteq \mathscr{V}$ is defined as $\mathbf{x}_{\mathscr{T}} = (x_1, \dots, x_N)$, where $x_i = 1$ for all $i \in \mathscr{T}$ and $x_i = 0$ for all $i \notin \mathscr{T}$. The *independent set polytope* $\mathscr{P}_I(\mathscr{G})$ of a graph $\mathscr{G}$ is the convex hull of the incidence vectors of the independent sets in $\mathscr{G}$. Next we consider what we will call the *clique polytope* $\mathscr{P}_C(\mathscr{G})$ determined by

$$\begin{array}{ll} x_i \geq 0, & \forall i \in \mathscr{V}, \\ \sum_{i \in \mathscr{C}_j} x_i \leq 1, & \forall \mathscr{C}_j \in \mathscr{S}. \end{array} \tag{3}$$

The second inequalities in Eq. (3) are called *clique inequalities*. Since the integer feasible points in $\mathscr{P}_C(\mathscr{G})$ are exactly the incidence vectors of independent sets, it follows that $\mathscr{P}_I(\mathscr{G})$ is the integer hull of $\mathscr{P}_C(\mathscr{G})$ (the convex full of its integer points).

A graph $\mathscr{G}$ is called *perfect* if its clique and coloring numbers are identical for all induced subgraphs of $\mathscr{G}$ (i.e., all subgraphs with the same vertex set and edge set being a subset of $\mathscr{E}$). An induced (chordless) cycle of odd length of at least 5 is called an *odd hole*, and an induced subgraph that is the complement of an odd hole is called an *odd antihole*. By the strong perfect graph theorem [22], a graph $\mathscr{G}$ is perfect if and only if $\mathscr{G}$ contains no odd hole and no odd antihole. In perfect graphs, the independent set polytope is equal to the clique polytope. Perfect graphs include many types of graphs such as bipartite graphs, line graphs of bipartite graphs, interval graphs, chordal graphs, distance-hereditary graphs, permutation graphs, trapezoid graphs, split graphs, and others [23]. Checking whether a graph is perfect can be done in polynomial time [24].

Let us now turn our attention to the MWIS problem. Assign to every node i in $\mathscr{G}$ a weight w_i. Without loss of generality we assume $w_i > 0$ for all i, since any nodes with non-positive weights can simply be removed from the graph in the context of the MWIS. Denote by d_i the degree of node i, i.e., $d_i \triangleq |\mathscr{N}_i|$, where here, $\mathscr{N}_i$ denotes the set of neighbors to i (i.e., all j with $(i,j) \in \mathscr{E}$). Let $D \triangleq \max_{i\in\mathscr{V}} d_i$ be the degree of $\mathscr{G}$. For convenience we assume that D is common knowledge to all nodes in $\mathscr{V}$. As we will see later, this assumption can be easily relaxed. Last, we assume that every node in $\mathscr{V}$ is unique, namely, each one has a distinct identifier not shared with any other nodes in the graph. For communication networks, for instance, MAC addresses of networking devices can be used in place of the identifiers.

The MWIS problem can be written as the following *Integer Linear Program (ILP)*:

$$\begin{array}{ll} \max & \sum_{i\in\mathscr{V}} w_i x_i \\ \text{s.t.} & x_i + x_j \le 1, \quad \forall (i,j) \in \mathscr{E}, \\ & x_i \in \{0,1\}, \quad \forall i \in \mathscr{V}, \end{array} \tag{4}$$

where the decision variables are the indicators $x_1, \ldots, x_N$ of whether a node a selected in the MWIS. The *Linear Programming (LP)* relaxation of (4) is simply the LP formed from (4) by relaxing the integer constraints $x_i \in \{0,1\}$ as $0 \le x_i \le 1$. We will call this LP the *edge relaxation* of MWIS.

Another formulation for the MWIS involves the clique inequalities:

$$\begin{array}{ll} \max & \sum_{i\in\mathscr{V}} w_i x_i \\ \text{s.t.} & \sum_{i\in\mathscr{C}_j} x_i \le 1, \quad \forall \mathscr{C}_j \in \mathscr{S}, \\ & x_i \in \{0,1\}, \quad \forall i \in \mathscr{V}. \end{array} \tag{5}$$

We will call the LP relaxation of (5) the *clique relaxation* of MWIS. Compared with (4), (5) describes the same feasible set for binary decision variables. However, the clique relaxation is tighter than the edge relaxation and this is important in our work. The price for this benefit is that there are more maximal cliques than edges. As we will see, the computational cost of the algorithm we propose increases (locally) linearly with the number of cliques. Yet, if the degree D is fixed and small, which is a reasonable assumption in wireless networks, the number of clique inequalities increases only linearly with the size of graph.

IV. Gradient projection phase

Consider the clique relaxation of MWIS

$$\begin{array}{ll} \max & \sum_{i\in\mathscr{V}} w_i x_i \\ \text{s.t.} & \sum_{i\in\mathscr{C}_j} x_i \le 1, \quad \forall j : \mathscr{C}_j \in \mathscr{S}, \\ & x_i \in [0,1], \quad \forall i \in \mathscr{V}. \end{array} \tag{6}$$

We will develop a distributed algorithm for solving this problem. Even though we defined $\mathscr{S}$ to contain only the maximal cliques in order to reduce the size of (6), the algorithm will be applicable for any set of cliques $\mathscr{S}$. Thus, the edge relaxation (i.e., when $\mathscr{S}$ contains only the 2-cliques) becomes a special case and the exact same algorithm can be used.

Let us add a logarithmic barrier function to problem (6), that is,

$$\begin{array}{ll} \max & \sum_{i\in\mathscr{V}} w_i x_i + \epsilon \sum_{i\in\mathscr{V}} (\log x_i + \log(1-x_i)) \\ \text{s.t.} & \sum_{i\in\mathscr{C}_j} x_i \le 1, \quad \forall j : \mathscr{C}_j \in \mathscr{S}, \\ & x_i \in [0,1], \quad \forall i \in \mathscr{V}, \end{array} \tag{7}$$

where ϵ is a small positive constant. As ϵ approaches 0, the objective value of (7) converges to that of problem (6). From now on we will call problem (7) the primal problem. Let $\boldsymbol{\theta} = (\theta_j; j : \mathscr{C}_j \in \mathscr{S})$ be the dual variables corresponding to the clique inequalities. The dual function is

$$\begin{aligned} q(\boldsymbol{\theta}) &= \max_{\mathbf{x}\in[0,1]^N} \{\textstyle\sum_{i\in\mathscr{V}} w_i x_i \\ &\quad + \epsilon \textstyle\sum_{i\in\mathscr{V}} (\log x_i + \log(1-x_i)) \\ &\quad + \textstyle\sum_{j:\mathscr{C}_j\in\mathscr{S}} \theta_j (1 - \sum_{i\in\mathscr{C}_j} x_i)\} \\ &= \max_{\mathbf{x}\in(0,1)^N} \{\textstyle\sum_{i\in\mathscr{V}} w_i x_i \\ &\quad + \epsilon \textstyle\sum_{i\in\mathscr{V}} (\log x_i + \log(1-x_i)) \\ &\quad - \textstyle\sum_{i\in\mathscr{V}} \sum_{j:\mathscr{C}_j\in\mathscr{S}, i\in\mathscr{C}_j} \theta_j x_i + \sum_{j:\mathscr{C}_j\in\mathscr{S}} \theta_j\} \\ &= \max_{\mathbf{x}\in(0,1)^N} \{\textstyle\sum_{i\in\mathscr{V}} [(w_i - \sum_{j:\mathscr{C}_j\in\mathscr{S}_j, i\in\mathscr{C}_j} \theta_j) x_i \\ &\quad + \epsilon(\log x_i + \log(1-x_i))] + \textstyle\sum_{j:\mathscr{C}_j\in\mathscr{S}} \theta_j\} \\ &= \textstyle\sum_{i\in\mathscr{V}} \max_{0<x_i<1} g_i(x_i) + \sum_{j:\mathscr{C}_j\in\mathscr{S}} \theta_j, \end{aligned}$$

where $g_i(x) \triangleq (w_i - \sum_{j:\mathscr{C}_j\in\mathscr{S}, i\in\mathscr{C}_j} \theta_j) x + \epsilon(\log x + \log(1-x))$. The following lemma establishes some useful properties of the solution $x_i(\boldsymbol{\theta})$ that maximizes $g_i(x)$. The proof of Lemma IV.1 is in Sec. VIII-A.

Lemma IV.1 *For all $i \in \mathscr{V}$, the unique maximizer $x_i(\boldsymbol{\theta}) \in (0,1)$ of $g_i(x)$ is given by*

$$x_i(\boldsymbol{\theta}) = \begin{cases} \dfrac{1 - \frac{2\epsilon}{a_i(\boldsymbol{\theta})} + \sqrt{\frac{4\epsilon^2}{(a_i(\boldsymbol{\theta}))^2} + 1}}{2}, & \text{if } a_i(\boldsymbol{\theta}) > 0, \\ \dfrac{1 - \frac{2\epsilon}{a_i(\boldsymbol{\theta})} - \sqrt{\frac{4\epsilon^2}{(a_i(\boldsymbol{\theta}))^2} + 1}}{2}, & \text{if } a_i(\boldsymbol{\theta}) < 0, \\ \dfrac{1}{2}, & \text{if } a_i(\boldsymbol{\theta}) = 0, \end{cases} \tag{8}$$

where $a_i(\boldsymbol{\theta}) = w_i - \sum_{j:i\in\mathscr{C}_j, \mathscr{C}_j\in\mathscr{S}} \theta_j$.

The dual of (7) is given by

$$\begin{array}{ll} \min & q(\boldsymbol{\theta}) \\ \text{s.t.} & \theta_j \ge 0, \quad \forall j : \mathscr{C}_j \in \mathscr{S}, \end{array} \tag{9}$$

$$q(\boldsymbol{\theta}) = \sum_{i\in\mathscr{V}} g_i(x_i(\boldsymbol{\theta})) + \sum_{j:\mathscr{C}_j\in\mathscr{S}} \theta_j,$$

$$g_i(x_i(\boldsymbol{\theta})) = \begin{cases} \dfrac{a_i - 2\epsilon + \sqrt{4\epsilon^2 + a_i^2}}{2} + \epsilon \log \dfrac{1 - \frac{2\epsilon}{a_i} + \sqrt{\frac{4\epsilon^2}{a_i^2} + 1}}{2} + \epsilon \log \dfrac{1 + \frac{2\epsilon}{a_i} - \sqrt{\frac{4\epsilon^2}{a_i^2} + 1}}{2}, & \text{if } a_i > 0, \\ 2\epsilon \log \dfrac{1}{2}, & \text{if } a_i = 0, \\ \dfrac{a_i - 2\epsilon + \sqrt{4\epsilon^2 + a_i^2}}{2} + \epsilon \log \dfrac{1 - \frac{2\epsilon}{a_i} - \sqrt{\frac{4\epsilon^2}{a_i^2} + 1}}{2} + \epsilon \log \dfrac{1 + \frac{2\epsilon}{a_i} + \sqrt{\frac{4\epsilon^2}{a_i^2} + 1}}{2}, & \text{if } a_i < 0. \end{cases} \tag{10}$$

where, after some algebra, the dual function gets simplified as in (10). (For economy of space we have suppressed the dependence of $a_i(\boldsymbol{\theta})$ on $\boldsymbol{\theta}$.) Furthermore, $g_i(x_i(\boldsymbol{\theta}))$ is continuously differentiable with respect to $\boldsymbol{\theta}$ and its partial derivative can be written as the following lemma suggests. The proof of this result is in Sec. VIII-B.

Lemma IV.2 *For all $i \in \mathscr{V}$, the partial derivative of $g_i(x_i(\boldsymbol{\theta}))$ with respect to θ_j is given by*

$$\frac{\partial g_i(x_i(\boldsymbol{\theta}))}{\partial \theta_j} = \begin{cases} -x_i(\boldsymbol{\theta}), & \textit{if } i \in \mathscr{C}_j, \\ 0, & \textit{otherwise.} \end{cases}$$

Based on this lemma it follows that for any $j : \mathscr{C}_j \in \mathscr{S}$,

$$\begin{aligned} \frac{\partial q(\boldsymbol{\theta})}{\partial \theta_j} &= \frac{\partial(\sum_{k\in\mathscr{V}} g_k(x_k(\boldsymbol{\theta})))}{\partial \theta_j} + 1 \\ &= \sum_{i:\mathscr{C}_j\in\mathscr{S}, i\in\mathscr{C}_j} \frac{\partial g_i(x_i(\boldsymbol{\theta}))}{\partial \theta_j} + 1 \\ &= 1 - \sum_{i:\mathscr{C}_j\in\mathscr{S}, i\in\mathscr{C}_j} x_i(\boldsymbol{\theta}). \end{aligned}$$

Note that the continuous differentiability of $q(\boldsymbol{\theta})$ is actually expected since $x_i(\boldsymbol{\theta})$ is the unique maximizer of $g_i(x)$. We can now solve the primal problem (7) by using the gradient projection method to solve the dual problem (9). For each iteration n, let $\mathbf{x}^{(n)}$ and $\boldsymbol{\theta}^{(n)}$ be the values of the vectors $\mathbf{x}$ and $\boldsymbol{\theta}$ at iteration n, and γ a pre-specified stepsize. The algorithm is given in Fig. 1. Notice that it only involves message-passing among neighboring nodes and uses local information. In particular, node i needs to maintain θ_j's only for the cliques in which i participates. Theorem IV.3 establishes the convergence of the algorithm for a sufficiently small stepsize γ (the proof can be found in Sec. VIII-C).

1) Initialization: set $\theta_j^{(0)} := \max_{i:i\in\mathscr{C}_j}\{w_i\}$ for all $j : \mathscr{C}_j \in \mathscr{S}$. Calculate $x_i^{(0)}$ according to Equation (8) for all $i \in \mathscr{V}$, and set $n := 1$.
2) At iteration n for all $i \in \mathscr{V}$,
   a) node i sends a message to all its neighbors $\mathscr{N}_i$, with the message being $x_i^{(n-1)}$;
   b) node i calculates $\theta_j^{(n)} := [\theta_j^{(n-1)} - \gamma(1 - \sum_{k:\mathscr{C}_j\in\mathscr{S},k\in\mathscr{C}_j} x_k^{(n-1)})]_+$, $\forall j : \mathscr{C}_j \in \mathscr{S}$, $i \in \mathscr{C}_j$;
   c) node i calculates $x_i^{(n)}$ according to equation (8) using $\theta_j^{(n)}$, $\forall j : \mathscr{C}_j \in \mathscr{S}$, $i \in \mathscr{C}_j$.
3) Set $n := n + 1$ and go to Step 2.

Fig. 1. Gradient projection algorithm for solving problem (7).

Theorem IV.3 *For any γ such that*

$$0 < \gamma < \frac{\epsilon}{2^{\frac{3D+5}{2}}(D+1)^{\frac{3}{2}}},$$

the algorithm in Fig. 1 converges to the optimal primal-dual pair $(\mathbf{x}^*, \boldsymbol{\theta}^*)$ that solves problems (7) and (9). Moreover, and to reach a dual solution $\boldsymbol{\theta}^{(n)}$ satisfying $|q(\boldsymbol{\theta}^{(n)}) - q(\boldsymbol{\theta}^*)| \le \sigma$, the algorithm requires $O(1/\sigma)$ iterations.

The algorithm in Fig. 1 is an infinite loop, while in implementations we almost always want to select a stopping criterion to terminate the algorithm once the solution we obtain is "close enough" to the theoretical optimal solution. One of the possible criteria is to simply run the loop for a pre-defined number of iterations. Another possible criterion is as follows: select a small constant $\delta > 0$, and stop the iterations if $|\theta_j^{(n)} - \theta_j^{(n-1)}| \le \delta$ for all $j : \mathscr{C}_j \in \mathscr{S}$. Although checking this criterion at each iteration seems to require global coordination among the nodes in graph $\mathscr{G}$, it can in fact be done in a distributed fashion using the competition algorithm we present later in Fig. 3. In particular, at each iteration n that is a multiple of $2D$, that is, $n = 2Dk$ for some positive integer k, each node $i \in \mathscr{V}$ individually decides if the iteration should continue, namely, sets a flag $f_i^{(k)}$ to 1 if

$|\theta_j^{(n)} - \theta_j^{(n-1)}| \leq \delta \; \forall j : \mathscr{C}_j \in \mathscr{S}, \, i \in \mathscr{C}_j$, and sets $f_i^{(k)}$ to 0 otherwise. Then, checking the stopping criterion amounts to verifying whether or not $\max_{i \in \mathscr{V}} f_i^{(k)} = 1$. To that end, the competition algorithm in Fig. 3 can be utilized to accomplish this in a distributed manner with the weight for node i set to $f_i^{(k)}$, $\forall i \in \mathscr{V}$, and this can be done concurrently with the gradient projection algorithm since they do not interfere with each other. Note that the choice of updating times is due to Theorem V.3 (also presented later), which guarantees that we will finish checking the stopping criterion before $f_i^{(k)}$ changes, $\forall i \in \mathscr{V}$.

We noted earlier that as ϵ in (7) approaches zero, the objective of the *barrier problem* in (7) converges to that of (6). There is a systematic way, called the *barrier method* (see [25, Chap. 4]), to ensure that the output $\mathbf{x}^*$ of the gradient method converges to the optimal solution of (6). To that end, we start with some ϵ_0, run the algorithm in Fig. 1 until it converges and then reduce ϵ (e.g., by multiplying it with some constant $\zeta \in (0, 1)$). In this manner, ϵ approaches zero geometrically and for any given required accuracy ϵ_d we need a polynomial number of iterations in $\log(1/\epsilon_d)$ to converge to an approximate optimal solution of (6). Considering that according to Thm. IV.3, each iteration of the barrier method takes a pseudo-polynomial [2] number of iterations in the desired dual function accuracy σ, we can conclude that the proposed gradient projection approach converges pseudo-polynomially to an approximate optimal solution of the MWIS clique relaxation (6). In practice, we can terminate the barrier iterations when the outputs of the algorithm in Fig. 1 for different ϵ are close enough. To implement this process we can store a fixed sequence of ϵ's at the nodes of $\mathscr{V}$ and program them to move to the next ϵ once convergence of the previous phase has been reached.

We note that the complexity conclusions reached above, consider σ as part of the input to the algorithm. The dependence of the running time on the number of nodes is polynomial.

A key result, summarized in Thm. IV.4 establishes the optimality of our algorithm for perfect graphs; the proof is in Sec VIII-D.

Theorem IV.4 *The optimal solution $\mathbf{x}^*$ of (6) obtained by the gradient projection algorithm is optimal for the MWIS problem (5) when $\mathscr{G}$ is perfect and the optimal solution of problem (6) is unique.*

The uniqueness assumption in Thm. IV.4 can be restrictive but it can be mitigated by randomly perturbing the node weights. In particular, since (6) is an LP problem over a polytope (bounded polyhedron) its optimal solution $\mathbf{x}^*$ is not unique if and only if $\mathbf{x}^*$ lies on a facet of the feasible set. Then, there always exist a small enough perturbation of the weights w_i that can render the optimal solution unique.

We note that [26] has proposed a message passing algorithm to solve the Maximum a Posteriori estimation problem (MAP) on perfect graphs. Since every MWIS problem is trivially a MAP (see [2]), the algorithm in [26] could, in principle, be used to solve the MWIS for perfect graphs as well. We say in principle because the application of the algorithm in [26] does not necessarily lead to a practical algorithm in our wireless network context. In particular, the algorithm of [26] operates on a graph which includes a node for every clique and every configuration (i.e., variable selection) of that clique. This can be a large number of nodes and it is not clear how the algorithm operations can be mapped to nodes of the wireless network. Our algorithm is different as it solves the MWIS directly without first transforming it to a MAP. Message-passing takes place on the conflict graph $\mathscr{G}$ whose nodes are links of the wireless network. Thus, the algorithm operations for each wireless link can be easily performed by one of the wireless devices incident to that link.

Next we make a connection between perfect graphs and wireless networks. Thm. IV.4 provides an optimality guarantee if the conflict graph $\mathscr{G}$ is perfect. While the conflict graph of an arbitrary wireless network $\mathscr{W} = (\mathscr{D}, \mathscr{L})$ is not necessarily perfect, Thm. IV.5 (the proof is in Sec. VIII-E) shows that if $\mathscr{W}$ is a tree and under the node-exclusive interference model, the corresponding conflict graph is indeed perfect. The node-exclusive interference model prescribes that link $i \in \mathscr{L}$ interferes with link $j \in \mathscr{L}$ if and only if they are sharing at least one common device in $\mathscr{D}$. We note that tree topologies are common, especially for wireless sensor networks.

Theorem IV.5 *Under the node-exclusive interference model, the conflict graph of a tree wireless network is a perfect graph.*

V. Estimation phase

In this section we describe how to construct a feasible solution to MWIS.

We assume that we have run the gradient projection procedure using the barrier method we outlined in Sec. IV and we have obtained an optimal solution $\mathbf{x}^*$ to the clique relaxation (6). For convenience we further assume that $x_i^* = 0$, for any node $i \in \mathscr{V}$ such that $w_i = 0$, which can be readily achieved by modifying any optimal solution to problem (6) with local information

[2] It would have been a polynomial number of iterations if it was polynomial in $\log(1/\sigma)$.

in case this assumption is violated. We next set to 0 or 1 any x_i, $i \in \mathscr{V}$, if the corresponding x_i^* is equal to 0 or 1, respectively.[3] This is justified by the following result, whose proof is in Sec. VIII-F.

Lemma V.1 *For any $i \in \mathscr{V}$ where $x_i^* \in \{0,1\}$, there is always an optimal solution $\tilde{\mathbf{x}}$ to MWIS (cf. (4) or (5)) such that $\tilde{x}_i = x_i^*$.*

As a result of Lemma V.1, we now have to assign node i to 0 or 1 only for those $i \in \mathscr{V}$ such that x_i^* was not deemed to be either 0 or 1. We next introduce two estimation algorithms that construct a feasible solution to the MWIS using $\mathbf{x}^*$. The first one is a greedy algorithm; the second algorithm is a low-complexity rounding algorithm based on coloring which, as we will see, has some interesting optimality properties.

1) A Greedy Estimation Algorithm: The greedy estimation algorithm we propose works as follows. A node $i \in \mathscr{V}$, takes the value of 0 or 1 if x_i^* is 0 or 1, respectively; otherwise, it sets itself to 0 or 1 in a way that maintains the feasibility of the solution only after all nodes in $\mathscr{N}_i$ whose weight is greater than w_i are processed. The detailed algorithm is depicted in Fig. 4, where χ is a special symbol standing for "undetermined." In case there is a tie in Step 2(b) of the algorithm, i.e., node i finds that it is tied for the largest weight in its neighborhood, then node i and all nodes in $\mathscr{N}_i$ with equal weights can use their unique IDs to break the tie, e.g., the node with the smallest ID can set itself to 1.

1) Initialization: for each node $i \in \mathscr{V}$, set $\hat{x}_i^{(0)} := 1$ if $x_i^* = 1$ and set $\hat{x}_i^{(0)} := 0$ if $x_i^* = 0$ or $w_i = 0$; otherwise set $\hat{x}_i^{(0)} := \chi$. Set $n := 1$.
2) At iteration n for all $i \in \mathscr{V}$,
   a) node i sends a message $(\hat{x}_i^{(n-1)}, w_i)$ to all nodes in $\mathscr{N}_i$;
   b) for any node $i \in \mathscr{V}$ such that $\hat{x}_i^{(n-1)} = \chi$: if $\exists j \in \mathscr{N}_i$ such that $\hat{x}_j^{(n-1)} = 1$, set $\hat{x}_i^{(n)} := 0$; else if $w_i > w_j$ or $\hat{x}_j^{(n-1)} = 0$ for all $j \in \mathscr{N}_i$, set $\hat{x}_i^{(n)} := 1$.
3) If $n = N$, stop and output $\hat{\mathbf{x}} := (\hat{x}_1^{(n)}, \dots, \hat{x}_N^{(n)})$; else set $n := n + 1$ and go to Step 2.

Fig. 2. Greedy estimation algorithm.

Next, we state the correctness of the estimation algorithm in Fig. 2. The proof of Thm. V.2 is in Sec. VIII-G.

[3] In practice, x_i^* may be not be exactly 0 or 1 due to roundoff errors. We can set a certain tolerance level to decide which x_i^* will be assumed to be 0 or 1.

Theorem V.2 *The estimation algorithm in Fig. 2 outputs a feasible solution to the MWIS problem.*

2) An Estimation Algorithm Based on Coloring: Before we present the estimation algorithm we describe how to color the graph $\mathscr{G}$. The objective of the coloring procedure is to label all nodes in $\mathscr{V}$ with colors such that no two adjacent nodes share the same color. To proceed, we first need to identify a node in $\mathscr{V}$ as the special node, i.e., the root, and inform each node in $\mathscr{G}$ whether or not it is the root.

As it will become more clear later on, the selection of the root influences how colors get assigned, which will in turn influence how the remaining conflicts in $\mathscr{G}$ get resolved. One attractive possibility is to select as root the node with the highest weight in $\mathscr{V}$. Other choices, however, are possible and one can experiment with several heuristics, which can affect the quality of the resulting solution especially in cases where a substantial number of conflicts remain in $\mathscr{G}$ after solving the clique relaxation. In fact, this flexibility is an interesting and useful feature of the coloring-based estimation algorithm.

In the remainder we assume that the root is the node with the largest weight. Other choices can be handled with minor modifications. We select the root with an iterative competition algorithm depicted in Fig. 3, where at each iteration n and for each node $i \in \mathscr{V}$, $r_i^{(n)}$ is the root to the best of node i's knowledge up to that point. Note that $r_i^{(n)}$ may not necessarily be the node with the highest weight in $\mathscr{V}$ until the algorithm converges, $\forall i \in \mathscr{V}$. Moreover, we assume that whenever there is a tie in the comparison in Step $2(b)$ of Fig. 3, it is broken by a stable arbitration. Let r be the root of the graph found by any centralized algorithm using the same arbitration mechanism as in Fig. 3. We also use the notation $\mathscr{N}_i^+ = \mathscr{N}_i \cup \{i\}$.

1) Initialization: set $r_i^{(0)} := i$, $s_i^{(0)} := w_i$ and $n := 1$.
2) At iteration n for all $i \in \mathscr{V}$,
   a) node i sends a message to all its neighbors $\mathscr{N}_i$, with the message being $(r_i^{(n-1)}, s_i^{(n-1)})$;
   b) node i updates $r_i^{(n)}$ and $s_i^{(n)}$ as $r_i^{(n)} := r_{j^*}^{(n-1)}$ and $s_i^{(n)} := s_{j^*}^{(n-1)}$, where $j^* = \arg\max_{j \in \mathscr{N}_i^+} s_j^{(n-1)}$.
3) If $n = N - 1$, stop and output $(r_1^{(n)}, \dots, r_N^{(n)})$; else set $n := n + 1$ and go to Step 2(a).

Fig. 3. Competition algorithm for finding the root.

In Theorem V.3 we establish the correctness of the competition algorithm. The proof is in Sec. VIII-H.

Theorem V.3 *The competition algorithm in Fig. 3 outputs r (i.e., $r_i^{(N-1)} = r,\ \forall i$) for any node $i \in \mathscr{V}$.*

In Sec. I we made the assumption that all the nodes in graph $\mathscr{G}$ know D, the degree of graph $\mathscr{G}$. As Theorem V.3 shows, this can be readily achieved in a distributed fashion using the competition algorithm in Fig. 3; the only modification is to let the weight of node i be d_i, $\forall i \in \mathscr{V}$.

With every node having the correct root information, graph $\mathscr{G}$ can be colored using at most $2D$ colors in a number of steps that is polynomial in the size of the graph [27], given that each node in $\mathscr{V}$ has a unique identifier and known by its neighbors. The coloring algorithm in [27] assumes a special node, which can be found by the competition algorithm in Fig. 3. Moreover, the coloring algorithm in [27] is a distributed one that is optimal for bipartite graphs, i.e., it colors any bipartite graph with two colors. In the sequel, for each node $i \in \mathscr{V}$ we denote by $c_i \in \{1, \ldots, 2D\}$ the color of itself as the result of the competition algorithm in Fig. 3 and the coloring algorithm in [27], and let $\mathbf{c} = (c_1, \ldots, c_N)$.

We now turn to the coloring-based estimation algorithm. The detailed algorithm is depicted in Fig. 4. A

---

1) Initialization: for each node $i \in \mathscr{V}$, set $\hat{x}_i^{(0)} := 1$ if $x_i^* = 1$ and set $\hat{x}_i^{(0)} := 0$ if $x_i^* = 0$ or $w_i = 0$; otherwise set $\hat{x}_i^{(0)} := \chi$. Set $n := 1$.
2) At iteration n for all $i \in \mathscr{V}$,
   a) node i sends a message $(\hat{x}_i^{(n-1)}, c_i)$ to all nodes in $\mathscr{N}_i$;
   b) for any node $i \in \mathscr{V}$ such that $\hat{x}_i^{(n-1)} = \chi$: if $\exists j \in \mathscr{N}_i$ such that $\hat{x}_j^{(n-1)} = 1$, set $\hat{x}_i^{(n)} := 0$; else if $c_i < c_j$ or $\hat{x}_j^{(n-1)} = 0$ for all $j \in \mathscr{N}_i$, set $\hat{x}_i^{(n)} := 1$.
3) If $n = 2D$, stop and output $\hat{\mathbf{x}} := (\hat{x}_1^{(n)}, \ldots, \hat{x}_N^{(n)})$; else set $n := n + 1$ and go to Step 2.

Fig. 4. Estimation algorithm based on coloring.

result similar to Thm. V.2 holds for this algorithm as well; we state it in Thm. V.4 and omit the proof because it is similar to that of Thm. V.2.

Theorem V.4 *The estimation algorithm in Fig. 4 outputs a feasible solution to the MWIS problem.*

We next establish an interesting optimality property of the coloring-based estimation algorithm. We will show that when $\mathscr{G}$ is bipartite it outputs an optimal solution to MWIS. More specifically, we first solve the edge relaxation of MWIS using the gradient projection approach of Sec. IV. (In fact, for bipartite graphs the edge and the clique relaxation are identical since 2-cliques are maximal cliques). Maintain the notation $(\mathbf{x}^*, \boldsymbol{\theta}^*)$ for the optimal solution to the edge relaxation and use this $\mathbf{x}^*$ as input to the Algorithm in Fig. 4. The following theorem establishes the bipartite optimality property; the proof is in Sec VIII-I.

Theorem V.5 *The solution $\hat{\mathbf{x}}$ given by the coloring-based estimation algorithm is optimal for the MWIS problem when $\mathscr{G}$ is bipartite.*

At first sight it appears that Thm. V.5 is a special case of Thm. IV.4 since bipartite graphs are perfect. There is though a subtle difference. Thm. V.5 does not require the uniqueness assumption of Thm. IV.4. Thus, if one wishes to solve the edge relaxation which is less expensive than the clique relaxation, it can obtain an optimal MWIS solution for bipartite graphs without having to resort into the perturbation strategy we discussed in Sec. IV.

We note that a different message passing algorithm for the edge relaxation was proposed in [2]. To find a feasible MWIS solution, the algorithm in [2] requires the LP relaxation to have a unique solution for general graphs; the same uniqueness condition is required for the algorithm in [2] to be optimal for bipartite graphs.

For completeness, in Fig. 5 we summarize the algorithm we have proposed so far for solving the MWIS problem on general graphs $\mathscr{G}$.

---

1) Phase I (Steps 1(a) and 1(b) are performed concurrently):
   a) use the algorithm in Fig. 1 to obtain the optimal solution $\mathbf{x}^*$ to problem (6).
   b) color the nodes in graph $\mathscr{G}$ using the competition algorithm in Fig. 3 and the coloring algorithm in [27].
2) Phase II: construct a feasible MWIS solution $\hat{\mathbf{x}}$ with either the greedy estimation algorithm of Fig. 2 or the coloring-based estimation algorithm of Fig. 4.

Fig. 5. The distributed algorithm for solving the MWIS problem (5).

Returning to the wireless scheduling application, we establish a sufficient condition for stability when the network is being scheduled using the algorithm in Fig. 5. Key to this result is the following lemma whose proof is in Sec. VIII-J.

Lemma V.6 *The feasible MWIS solution $\hat{\mathbf{x}}$ produced by either the greedy estimation algorithm of Fig. 2 or* *the coloring-based estimation algorithm of Fig. 4 is a maximal independent set.*

Consider now the scheduling policy induced by the MWIS solution $\hat{\mathbf{x}}$. In particular, at each time slot t we construct the conflict graph as described in Sec. I and run the algorithm in Fig. 5. Given Lemma V.6, [28, Theorem 1] guarantees the stability of this scheduling policy. We formalize this result as Theorem V.7; recall the notation $\mathscr{N}_i^+ \triangleq \{i\} \cup \mathscr{N}_i$.

Theorem V.7 *The scheduling policy induced by the algorithm in Fig. 5 is stable in the mean, that is, the process* $\{\mathbf{Q}^{(t)}\}$ *satisfies (2), as long as* $\sum_{j \in \mathscr{N}_i^+} \frac{\lambda_j}{\mu_j} < 1$ *for all* $i = 1, \ldots, N$.

VI. Implementation issues and simulation results

In this section we return to the wireless application we introduced in Sec. I and apply the algorithm of Fig. 5. As we will see, the clique relaxation is quite tight and the scheduling policy we obtain results in a smaller long-run average backlog compared to a CSMA policy and another greedy alternative.

We first explore the tightness of the clique relaxation. We have already seen that for tree networks and under the node-exclusive interference model the clique relaxation is indeed tight and will yield an integer optimal solution. How tight is it though for more general topologies and under different interference models? To answer this question, we consider five randomly generated 20-node wireless networks over a 100×100 area. We adopt a general interference model where links within a given distance to each other interfere. For each network we generate the corresponding conflict graph, then randomly generate the node weights, and solve the corresponding clique relaxation of MWIS. Table I reports the average ratio of fractional elements in the relaxed optimal solution over the total number of nodes averaged over 1,000 runs. These results suggest that the clique relaxation is able to decide how the links of the network should be scheduled with very few exceptions on average. This is quite interesting, perhaps even surprising, as it suggests that for random instances of wireless networks our distributed algorithm yields a near-optimal solution. Motivated by this finding, in the remainder of this section we only use the (simpler and less computationally demanding) greedy estimation algorithm from Sec. V to make decisions for the remaining links corresponding to the non-integer elements in the solution.

TABLE I
THE PERCENTAGE OF NON-INTEGER VARIABLES IN THE OPTIMAL SOLUTION OF THE MWIS CLIQUE RELAXATION.

	Case1	Case2	Case3	Case4	Case5
ratio	0.49%	0.08%	0.58%	0%	0%

Next we describe how we schedule the wireless links based on the solution produced by the algorithm of Fig. 5. As described in Sec. I, at the beginning of each time slot t we construct the conflict graph and set $w_i^{(t)} = \frac{Q_i^{(t-1)}}{\mu_i}$ for all $i = 1, \ldots, N$. At the beginning of each time slot t we make the scheduling decision for any link i as follows: if $\hat{x}_i = 1$, link i is scheduled to transmit during time slot t; otherwise it is not. Alternatively we can simply set $X_i^{(t)} = \hat{x}_i$, $\forall i = 1, \ldots, N$. Clearly, such a scheduling policy is free of interference by construction of $\hat{\mathbf{x}}$. Moreover, this scheduling policy is fairly intuitive, aiming to reduce the packet congestion by giving higher priority to those links with greater "normalized backlogs," that is, queue lengths divided by their corresponding transmission rates.

We provide implementation details of the scheduling policy in Fig. 6. During each time slot we perform just a single iteration of either the gradient projection method or the estimation algorithm. This is done because wireless devices are typically (e.g., as in the case of wireless sensor networks) not powerful enough to perform many iterations during a time slot. Of course, if one deals with a situation where devices can afford more computations it is easy to modify the algorithm and take advantage. Another reason for performing only one iteration per time slot is that we want to have a fair comparison with CSMA and the greedy policy which do the same. In the policy of Fig. 6, we perform only a finite number of iterations from both the gradient projection method and the estimation algorithm. We introduce two quantities, T_{Grad} and T_{Est}, which denote the number of time slots over which we run the gradient projection method and the estimation algorithm, respectively. There is a trade-off in setting these values: the larger they are the better the policy we compute at the expense of updating the policy less frequently. We also note that the algorithm in Fig. 6 implements a modified version of the greedy estimation method. It is clear by construction that the scheduling policy produced by the algorithm is interference-free.

In practice, it is useful to have a conflict graph invariant over time, thus, avoiding having to "reconstruct" the graph every time we wish to solve the MWIS problem. To that end, we can simply assume that the nodes of $\mathscr{G}$ contain all possible wireless links that could potentially be established and the edges of $\mathscr{G}$ indicate all possible interfering pairs of wireless links. At each time slot many of these links will have empty queues and will be assigned zero weights. Before solving the MWIS we can fix decisions $X_i^{(t)} = 0$ for every such link i and proceed

1) Initialization: set $t := 0$, $X_i^{(t)} := 0$ for all data links i, $w_i := \frac{Q_i^{(0)}}{\mu_i}$, $stage := 0$, and $Phase := G$.
2) If $Phase = G$, go to Step 3, if $Phase = E$ go to Step 4.
3) Gradient Projection: if $stage < T_{Grad}$, perform one iteration of the gradient projection method, set $stage := stage + 1$, and go to Step 5. Else if $stage = T_{Grad}$, set $stage := 0$, $Phase := E$, $isSet_i := false$, $X_i^{next} := 0$ for all i, and go to Step 5.
4) Estimation: if $stage < T_{Est}$, for each i do:
   a) If $\exists j \in \mathcal{N}_i$ with $isSet_j = true$ and $X_j^{next} = 1$ then set $isSet_i := true$ and $X_i^{next} := 0$.
   b) Else if $x_j < x_i$ for any $j \in \mathcal{N}_i$ such that $isSet_j = false$, then set $isSet_i := true$ and $X_i^{next} := 1$.
   c) Else if $x_j \leq x_i$ for any $j \in \mathcal{N}_i$ such that $isSet_j = false$ and $ID_i < ID_j$ for all $j \in \mathcal{N}_i$ such that $isSet_j = false$ and $x_j = x_i$, then set $isSet_i := true$ and $X_i^{next} := 1$.

   If $stage = T_{Est}$, then set $stage := 0$, for all i, update schedule $X_i := X_i^{next}$, and set $w_i := Q_i(t)/\mu_i$ and $Phase := G$.
5) t:=t+1, go to Step 2.

Fig. 6. The scheduling algorithm. ID_i denotes the unique ID of link i.

to solve the MWIS for the remaining links. In typical wireless networks deployed over a large area many pairs of nodes may never be able to communicate and many links may never interfere. Thus, the resulting conflict graph will have much fewer than $|\mathscr{D}|(|\mathscr{D}| - 1)$ nodes and will be relatively sparse.

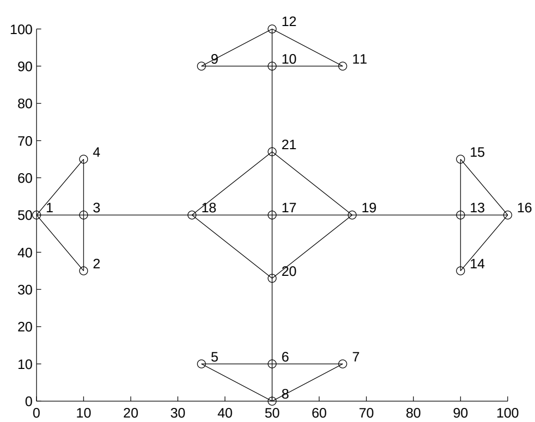


Fig. 7. Star-topology network.

Next, we use an example to illustrate the benefits of the clique MWIS relaxation we have proposed in this paper with the edge relaxation. Consider a three-node wireless network with independent Poisson packet arrivals and rates set to: $\lambda_{1,2} = 0.4286$, $\lambda_{2,3} = \lambda_{3,1} = 0.2857$, and $\lambda_{2,1} = \lambda_{3,2} = \lambda_{1,3} = 0$ packets per time slot. All links have a constant transmission rate $\mu = 1$ per time slot. We schedule the network in two ways: first by using the edge relaxation and then by using the clique relaxation (the maximum clique size is 3 in this example). In both cases, we run the network for $10,000$ time units. The average total queue length achieved by the policy using the clique relaxation is 3.64, whereas the policy using the clique relaxation achieves an average total queue length of 2.68; a difference of 36%. We can clearly see the benefit of introducing the tighter clique relaxation.

We are now ready to compare the performance of several alternative policies. Consider the wireless network in Fig. 7 which we view as typical of many wireless networks and wireless sensor networks. In particular it has the structure of several clusters of nodes with sparse intra-cluster connectivity. The arrivals for each link follow a Poisson process, independent of all other arrival processes. We set $\lambda_{2,1} = \lambda_{5,19} = \lambda_{9,20} = \lambda_{12,18} = \lambda_{16,17} = 0.5$ packets per time unit. All transmission rates μ are set to 1. We generate multiple instances of the problem by scaling each arrival rate as $\alpha\lambda$, where $\alpha \in (0, \infty)$ is a scaling factor. The system evolves in discrete time, so arrivals are assumed to occur at the beginning of each time slot and are distributed according to a Poisson random variable with parameter $\lambda\Delta t$, where Δt is the duration of the time slot.

We measure the performance of a scheduling policy by the aggregate long-run average queue length for $\alpha \in (0, 1)$, that is,

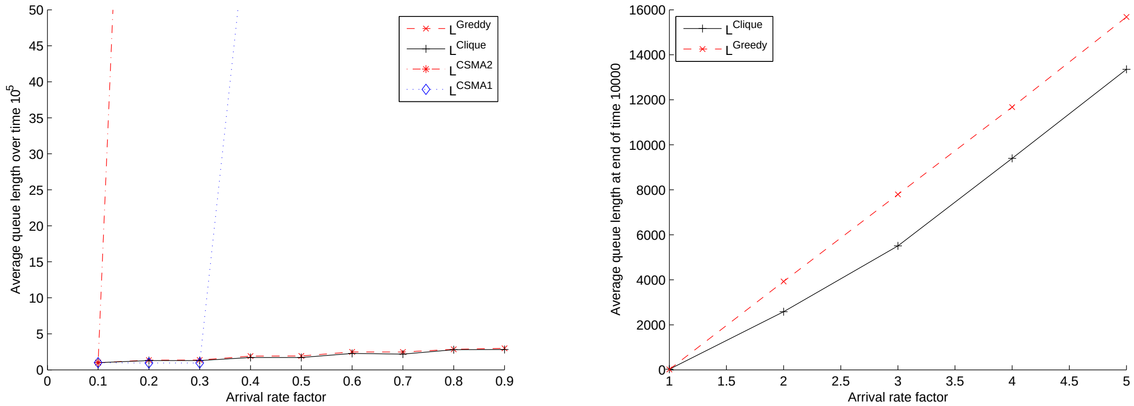
$$L \triangleq \lim_{T\to\infty} \frac{1}{NT} \sum_{t=1}^{T} \sum_{i=1}^{N} Q_i^{(t)}.$$

For $\alpha \in (1, \infty)$ the system becomes unstable and we measure the performance by the average queue length at some (large) finite horizon T, i.e.,

$$L \triangleq \frac{1}{N} \sum_{i=1}^{N} Q_i^{(T)}.$$

Considering such unstable traffic scenarios is interesting as they provide insight on how different scheduling policies behave during periods the network is overloaded.

Let L^{Clique}, L^{CSMA1}, L^{CSMA2} and L^{Greedy} be the average queue lengths achieved by our proposed scheduling algorithm in Fig. 6 (Clique), the two CSMA algorithms (CSMA1 and CSMA2) in [11] and [14], and a simple greedy scheduling algorithm (Greedy) that just



(a) Stable network.

(b) Overloaded network.

Fig. 8. Comparing the performance of various policies.

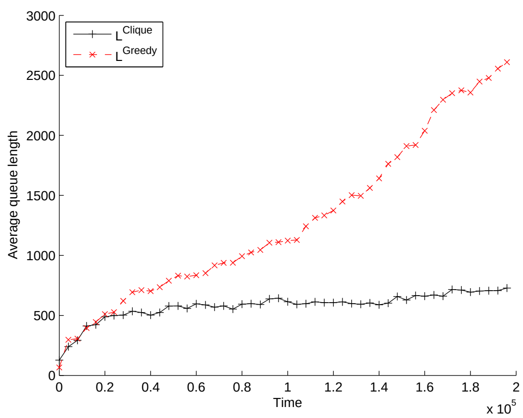


Fig. 9. Comparing our policy and greedy in an 8-node circular network.

uses the greedy estimation algorithm of Fig. 2. Simulations were run using various arrival rate parameters α and the results are depicted in Fig. 8 (a) (stable network) and Fig. 8 (b) (overloaded network).

The results suggest that our policy and Greedy substantially outperform both CSMA variants. In overloaded traffic scenarios our algorithm outperforms Greedy, by as much as 29% based on the results of Fig. 8 (b). Still, it is interesting that Greedy does so well, thus confirming earlier empirical evidence. Yet, this is not universally true. Using insight from [29], we considered a wireless network which forms an 8-node cycle, i.e., for $i = 0, \ldots, 7$ node i can only "talk" to nodes $i - 1 \mod 8$ and $i + 1 \mod 8$. The interference model is the node-exclusive model, that is, two links sharing a node can not transmit simultaneously. With this interference model, two links incident to the same nodes but different transmit directions can be considered as one link and the conflict graph becomes an 8-node cycle as well. The arrival rate is set to $\lambda = 0.5$ per time unit for all links and the transmission rate μ is set to 1. Fig. 9 compares the performance of our policy with the Greedy when $\alpha = 0.998$ (i.e., heavy traffic). It is clear that the greedy policy fails to stabilize the network, while our algorithm maintains stability.

VII. Conclusions

We considered the problem of scheduling wireless networks and converted it into the MWIS problem. We devised a low-complexity, two-phase distributed algorithm to approximately solve the MWIS. The first phase solves a relaxation of the MWIS and the second phase utilizes the output from the first one to construct a feasible MWIS solution. The MWIS clique relaxation we solve is substantially tighter than the standard edge MWIS relaxation earlier work in the literature has considered. This allows us to establish that the relaxation yields an optimal MWIS solution for perfect graphs, generalizing again earlier work that focused on bipartite graphs.

We applied this approach to several instances of typical wireless networks. First, we showed that for tree networks under the node-exclusive interference model our approach is optimal. Numerical results indicate that for more general wireless networks and interference models, the gradient projection phase can produce an almost optimal solution for MWIS with a very small percentage of non-integer variables. Yet, it may require a non-trivial number of iterations to converge. To make the approach practical in settings where the wireless devices can not afford a heavy computational overhead, we devised variants where only a moderate number of iterations gets performed between any two scheduling policy updates. Coupled with a simple estimation algorithm that guarantees an interference-free schedule, our approach is shown to substantially outperform CSMA-type policies across all traffic scenarios (from light to heavy) and also outperform a greedy policy in heavy traffic scenarios (substantially in some instances).

We close by mentioning that even though we considered a synchronous implementation of our algorithms, asynchronous versions of both the gradient projection method and the two estimation algorithms are possible and can also be used in settings where maintaining a common clock is undesirable.

VIII. Proofs

A. Proof of Lemma IV.1

Proof: The first order derivative of $g_i(x)$ is:

$$\frac{dg_i(x)}{dx} = a_i(\boldsymbol{\theta}) + \epsilon(\frac{1}{x} + \frac{1}{x-1}).$$

Solving $\frac{dg_i(x)}{dx} = 0$ yields:

$$x_i(\boldsymbol{\theta}) = \begin{cases} \dfrac{1 - \frac{2\epsilon}{a_i} \pm \sqrt{\frac{4\epsilon^2}{a_i^2} + 1}}{2}, & \text{if} \quad a_i \neq 0, \\ \dfrac{1}{2}, & \text{otherwise,} \end{cases}$$

where we suppress the dependence of $a_i(\boldsymbol{\theta})$ on $\boldsymbol{\theta}$. It is not difficult to verify that:

$$\begin{aligned} &0 < x_1 < 1,\ x_2 < 0, \quad \text{if} \quad a_i > 0, \\ &x_1 > 1,\ 0 < x_2 < 1, \quad \text{if} \quad a_i < 0. \end{aligned}$$

Thus, we have

$$x_i(\boldsymbol{\theta}) = \begin{cases} \dfrac{1 - \frac{2\epsilon}{a_i} + \sqrt{\frac{4\epsilon^2}{a_i^2} + 1}}{2}, & \text{if } a_i > 0, \\ \dfrac{1 - \frac{2\epsilon}{a_i} - \sqrt{\frac{4\epsilon^2}{a_i^2} + 1}}{2}, & \text{if } a_i < 0, \\ \dfrac{1}{2}, & \text{if } a_i = 0. \end{cases}$$

By using L'Hospital's rule, we can verify that $x_i(\boldsymbol{\theta})$ is continuous at $a_i(\boldsymbol{\theta}) = 0$, thus, $x_i(\boldsymbol{\theta})$ is a continuous function over $(-\infty, \infty)$. The second order derivative of $g_i(x)$ is:

$$\frac{d^2 g_i(x)}{dx^2} = -\epsilon(\frac{1}{x^2} + \frac{1}{(x-1)^2}) < 0.$$

Thus, $g_i(x)$ is strictly concave over $x \in (0,1)$ which establishes that $x_i(\boldsymbol{\theta})$ is the unique maximizer. ∎

B. Proof of Lemma IV.2

Proof: If $i \notin \mathscr{C}_j$, it is trivial to see that $\frac{\partial g_i(x_i(\boldsymbol{\theta}))}{\partial \theta_j} = 0$. We only consider the other case: $i \in \mathscr{C}_j$. If $a_i(\boldsymbol{\theta}) > 0$, let $y_i = \frac{2\epsilon}{a_i(\boldsymbol{\theta})}$ and we have:

$$\begin{aligned}
&\frac{\partial g_i(x_i(\boldsymbol{\theta}))}{\partial \theta_j} = \frac{\partial g_i(x_i(\boldsymbol{\theta}))}{\partial a_i}\frac{\partial a_i}{\partial \theta_j} \\
&= -\frac{\partial g_i(x_i(\boldsymbol{\theta}))}{\partial a_i} \\
&= -\frac{1 + \frac{a_i}{\sqrt{4\epsilon^2 + (a_i)^2}}}{2} \\
&\quad - \frac{\partial(\epsilon \log \frac{1 - y_i + \sqrt{y_i^2 + 1}}{2})}{\partial y_i}\frac{\partial y_i}{\partial a_i} \\
&\quad - \frac{\partial(\epsilon \log \frac{1 + y_i - \sqrt{y_i^2 + 1}}{2})}{\partial y_i}\frac{\partial y_i}{\partial a_i} \\
&= -\frac{1 + \frac{1}{\sqrt{y_i^2 + 1}}}{2} \\
&\quad - \frac{\epsilon}{1 - y_i + \sqrt{y_i^2 + 1}}(-1 + \frac{y_i}{\sqrt{y_i^2 + 1}})(\frac{-2\epsilon}{a_i^2}) \\
&\quad - \frac{\epsilon}{1 + y_i - \sqrt{y_i^2 + 1}}(1 - \frac{y_i}{\sqrt{y_i^2 + 1}})(\frac{-2\epsilon}{a_i^2}) \\
&= -\frac{1 + \frac{1}{\sqrt{y_i^2 + 1}}}{2} \\
&\quad - \frac{-y_i^2}{2(1 - y_i + \sqrt{y_i^2 + 1})}(-1 + \frac{y_i}{\sqrt{y_i^2 + 1}}) \\
&\quad - \frac{-y_i^2}{2(1 + y_i - \sqrt{y_i^2 + 1})}(1 - \frac{y_i}{\sqrt{y_i^2 + 1}}) \\
&= -\frac{\sqrt{y_i^2 + 1} + 1}{2\sqrt{y_i^2 + 1}} \\
&\quad + \frac{y_i^2(y_i - \sqrt{y_i^2 + 1})}{2\sqrt{y_i^2 + 1}}(\frac{1}{1 - y_i + \sqrt{y_i^2 + 1}} \\
&\quad + \frac{(-1)}{1 + y_i - \sqrt{y_i^2 + 1}}) \\
&= -\frac{\sqrt{y_i^2 + 1} + 1}{2\sqrt{y_i^2 + 1}} + \frac{y_i^2(y_i - \sqrt{y_i^2 + 1})}{2\sqrt{y_i^2 + 1}}(\frac{-1}{y_i}) \\
&= \frac{-\sqrt{y_i^2 + 1} - 1 - y_i^2 + y_i\sqrt{y_i^2 + 1}}{2\sqrt{y_i^2 + 1}} \\
&= \frac{-1 - \sqrt{1 + y_i^2} + y_i}{2} = -x_i(\boldsymbol{\theta}),
\end{aligned}$$

where we have suppressed the dependence of $a_i(\boldsymbol{\theta})$ and $y_i(\boldsymbol{\theta})$ on $\boldsymbol{\theta}$. If $a_i(\boldsymbol{\theta}) < 0$, the proof is similar and therefore omitted. ∎

C. Proof of Theorem IV.3

To prove Theorem IV.3, we need the following lemma that characterizes the dual function $q(\cdot)$.

Lemma VIII.1 *The dual function $q(\cdot)$ is lower bounded and $\nabla q(\cdot)$ is Lipschitz continuous, i.e.,*
(1) $q(\boldsymbol{\theta}) \geq \frac{\sum_{i\in\mathscr{V}} w_i}{D+1} + N\epsilon \log \frac{D}{(D+1)^2}, \forall \boldsymbol{\theta} \geq \mathbf{0}$.
(2) $||\nabla q(\boldsymbol{\theta}) - \nabla q(\boldsymbol{\nu})||_2 \leq \frac{2^{\frac{3(D+1)}{2}} 4(D+1)^{\frac{3}{2}}}{\epsilon} ||\boldsymbol{\theta} - \boldsymbol{\nu}||_2$ $\forall \boldsymbol{\theta}, \boldsymbol{\nu} \geq \mathbf{0}$.

Proof: (1) From the definition of $q(\boldsymbol{\theta})$, we have

$$\begin{aligned} q(\boldsymbol{\theta}) = &\max_{\mathbf{x}\in[0,1]^N} \{\textstyle\sum_{i\in\mathscr{V}} w_i x_i \\ &+ \epsilon \sum_{i\in\mathscr{V}} (\log x_i + \log(1-x_i)) \\ &+ \sum_{j:\mathscr{C}_j\in\mathscr{S}} \theta_j (1 - \sum_{i\in\mathscr{C}_j} x_i)\} \\ \geq &\epsilon \sum_{i\in\mathscr{V}} (\log \frac{1}{D+1} + \log \frac{D}{D+1}) \\ &+ \sum_{i\in\mathscr{V}} w_i \frac{1}{D+1} + \sum_{j:\mathscr{C}_j\in\mathscr{S}} \theta_j (1 - \sum_{i\in\mathscr{C}_j} \frac{1}{D+1}) \\ \geq &\textstyle\sum_{i\in\mathscr{V}} w_i \frac{1}{D+1} + \epsilon \sum_{i\in\mathscr{V}} \log \frac{D}{(D+1)^2} \\ = &\frac{\sum_{i\in\mathscr{V}} w_i}{D+1} + N\epsilon \log \frac{D}{(D+1)^2}, \end{aligned}$$

where the first inequality is due to the fact that $x_i = \frac{1}{D+1}, \forall i \in \mathscr{V}$, is a feasible solution in $[0,1]^N$, and the second is due to $1 - \sum_{i\in\mathscr{C}_j} \frac{1}{D+1} \geq 0$ and $\boldsymbol{\theta} > \mathbf{0}$.
(2) It can be seen that

$$\begin{aligned} ||\nabla q(\boldsymbol{\theta}) - \nabla q(\boldsymbol{\nu})||_2 &\leq ||\nabla q(\boldsymbol{\theta}) - \nabla q(\boldsymbol{\nu})||_1 \\ &= \textstyle\sum_{j:\mathscr{C}_j\in\mathscr{S}} \sum_{i:i\in\mathscr{C}_j} |x_i(\boldsymbol{\theta}) - x_i(\boldsymbol{\nu})| \\ &\leq 2^{D+1} \textstyle\sum_{i\in\mathscr{V}} |x_i(\boldsymbol{\theta}) - x_i(\boldsymbol{\nu})|, \end{aligned}$$

where the last inequality is due to the fact that for each node i, the number of maximal cliques containing i is less than 2^{D+1}.

To proceed with the analysis, we next examine the functions $x_i(\boldsymbol{\theta})$. Let us suppress again the dependence of $a_i(\boldsymbol{\theta})$ on $\boldsymbol{\theta}$. After some algebra, for $i \in \mathscr{C}_j, \mathscr{C}_j \in \mathscr{S}$ we obtain

$$\frac{dx_i(a_i)}{da_i} = \begin{cases} \frac{\epsilon}{a_i^2} - \frac{2\epsilon^2}{a_i^2\sqrt{4\epsilon^2 + a_i^2}}, & \text{if } a_i \neq 0, \\ \frac{4}{\epsilon}, & \text{if } a_i = 0. \end{cases} \quad (11)$$

The second derivative of $x_i(a_i)$ ($a_i \neq 0$) is

$$\frac{d^2 x_i(a_i)}{da_i^2} = \frac{2\epsilon(3\epsilon a_i^2 + 8\epsilon^3 - (a_i^2 + 4\epsilon^2)^{\frac{3}{2}})}{a_i^3 (a_i^2 + 4\epsilon^2)^{\frac{3}{2}}}, \quad (12)$$

and since $(3\epsilon a_i^2 + 8\epsilon^3)^2 < (a_i^2 + 4\epsilon^2)^3$, it follows that $\frac{d^2 x_i(a_i)}{da_i^2}$ is positive if and only if $a_i < 0$. In other words, $\frac{dx_i(a_i)}{da_i}$ is monotonically decreasing if $a_i > 0$ and monotonically increasing if $a_i < 0$. Combined with the fact $\lim_{a_i\to\infty} \frac{dx_i(a_i)}{da_i} = \lim_{a_i\to-\infty} \frac{dx_i(a_i)}{da_i} = 0$, we have $|\frac{dx_i(a_i)}{da_i}| \leq \frac{4}{\epsilon}$ for any real number a_i, $\forall i \in \mathscr{V}$. Therefore, by the mean value theorem, we have $|x_i(a_i) - x_i(b_i)| \leq \frac{4}{\epsilon}|a_i - b_i|$, for any real numbers a_i, b_i, $\forall i \in \mathscr{V}$. With this we have

$$\begin{aligned} &|\nabla q(\boldsymbol{\theta}) - \nabla q(\boldsymbol{\nu})||_2 \leq 2^{D+1} \textstyle\sum_{i\in\mathscr{V}} |x_i(\boldsymbol{\theta}) - x_i(\boldsymbol{\nu})| \\ &\leq 2^{D+1} \textstyle\sum_{i\in\mathscr{V}} \frac{4}{\epsilon} (|\sum_{j:i\in\mathscr{C}_j,\mathscr{C}_j\in\mathscr{S}} (\theta_j - \nu_j)|) \\ &\leq \frac{2^{D+3}}{\epsilon} \textstyle\sum_{i\in\mathscr{V}} \sum_{j:i\in\mathscr{C}_j,\mathscr{C}_j\in\mathscr{S}} |\theta_j - \nu_j| \\ &\leq \frac{8(D+1)2^D}{\epsilon} ||\boldsymbol{\theta} - \boldsymbol{\nu}||_1 \\ &\leq \frac{8(D+1)2^D \sqrt{(D+1)2^{D+1}}}{\epsilon} ||\boldsymbol{\theta} - \boldsymbol{\nu}||_2 \\ &\leq \frac{2^{\frac{3(D+1)}{2}} 4(D+1)^{\frac{3}{2}}}{\epsilon} ||\boldsymbol{\theta} - \boldsymbol{\nu}||_2, \end{aligned}$$

where the last inequality follows from the fact that $||\mathbf{a}||_1 \leq \sqrt{n}||\mathbf{a}||_2$, $\forall \mathbf{a} \in \mathbb{R}^n$. ■

Next we prove Theorem IV.3.

Proof: The convergence of the gradient projection algorithm in Fig. 1, as well as the dual optimality of the limit point $\boldsymbol{\theta}^*$, are a direct result of Lemma VIII.1 in [30, Page 214], given that the stepsize γ satisfies the condition

$$0 < \gamma < \frac{2}{\frac{2^{\frac{3(D+1)}{2}} 4(D+1)^{\frac{3}{2}}}{\epsilon}} = \frac{\epsilon}{2^{\frac{3D+5}{2}} (D+1)^{\frac{3}{2}}}.$$

In the rest of the proof we show $\mathbf{x}^*$ is the optimal solution to the primal problem (7). First we note that $\mathbf{x}^*$ is a feasible solution. To see this, note that $0 \leq x_i(\boldsymbol{\theta}^*) \leq 1$ for all $i \in \mathscr{V}$ due to Equation (8). Furthermore, since the algorithm converges with $\boldsymbol{\theta}^* \geq \mathbf{0}$ for some positive γ, i.e.,

$$\theta_j^* = [\theta_j^* - \gamma(1 - \textstyle\sum_{i:i\in\mathscr{C}_j} x_i^*)]_+, \quad (13)$$

it must hold that $\sum_{i:i\in\mathscr{C}_j} x_i^* \leq 1$, and thus $\mathbf{x}^*$ is feasible for problem (7). Also, the above equation indicates that complementary slackness holds, that is, $\theta_j^*(1 - \sum_{i\in\mathscr{C}_j} x_i^*) = 0$ for all $j, \mathscr{C}_j \in \mathscr{S}$.

Next we proceed to prove the primal optimality of $\mathbf{x}^*$. Let $\mathscr{X} \triangleq \{\mathbf{x} \mid \mathbf{x} \in [0,1]^N \text{ and } \sum_{i:i\in\mathscr{C}_j,\mathscr{C}_j\in\mathscr{S}} x_i^* \leq 1\}$ and we have

$$\begin{aligned} &\textstyle\sum_{i\in\mathscr{V}} w_i x_i^* + \epsilon \sum_{i\in\mathscr{V}} (\log x_i^* + \log(1 - x_i^*)) \\ &= \max_{\mathbf{x}\in[0,1]^N} \{\textstyle\sum_{i\in\mathscr{V}} w_i x_i + \epsilon \sum_{i\in\mathscr{V}} (\log x_i + \log(1 - x_i)) \\ &\quad + \textstyle\sum_{j:\mathscr{C}_j\in\mathscr{S}} \theta_j^* (1 - \sum_{i:i\in\mathscr{C}_j} x_i^*)\} \\ &\geq \max_{\mathbf{x}\in\mathscr{X}} \{\textstyle\sum_{i\in\mathscr{V}} w_i x_i + \epsilon \sum_{i\in\mathscr{V}} (\log x_i + \log(1 - x_i)) \end{aligned}$$

$$+\sum_{j:\mathscr{C}_j\in\mathscr{S}}\theta_j^*(1-\sum_{i:i\in\mathscr{C}_j}x_i^*)\}$$
$$\geq\max_{\mathbf{x}\in\mathscr{X}}\{\sum_{i\in\mathscr{V}}w_ix_i+\epsilon\sum_{i\in\mathscr{V}}(\log x_i+\log(1-x_i))\},$$

where the first equality is due to strong duality.

Finally we establish the convergence rate. Since $q(\boldsymbol{\theta})$ is lower bounded and $\nabla q(\cdot)$ is Lipschitz continuous, a result from [31] guarantees that $|q(\boldsymbol{\theta}^{(n)})-q(\boldsymbol{\theta}^*)|=O(1/n)$. Thus, to reach a dual solution satisfying $|q(\boldsymbol{\theta}^{(n)})-q(\boldsymbol{\theta}^*)|\leq\sigma$ we need $O(1/\sigma)$ iterations. ■

D. Proof of Theorem IV.4

Proof: If $\mathscr{G}$ is perfect, then the independent set polytope $\mathscr{P}_I(\mathscr{G})$ is equal to the clique polytope $\mathscr{P}_C(\mathscr{G})$. Thus, every vertex of $\mathscr{P}_C(\mathscr{G})$ is integral. Since problem (6) has a unique optimal solution, that solution must be one of the integral vertices of $\mathscr{P}_C(\mathscr{G})$. Since the clique relaxation (6) is indeed a relaxation of MWIS, it follows that its integral optimal solution is an optimal solution of MWIS. We have already established that the gradient projection method can be used to converge to an optimal solution of (6), hence, an optimal solution of MWIS. ■

E. Proof of Thm. IV.5

Proof: Consider the conflict graph $\mathscr{G}=(\mathscr{V},\mathscr{E})$ of the wireless network $\mathscr{W}=(\mathscr{D},\mathscr{L})$ and recall that $\mathscr{V}$ is identical to the set of links $\mathscr{L}$ of the wireless network. First we establish a property of all the induced graphs of the conflict graph $\mathscr{G}$: if node i and node j of $\mathscr{G}$ are not connected directly in any of the induced graphs, then there is no cycle of $\mathscr{G}$ in which nodes i and j participate.

To that end, suppose there is a cycle in $\mathscr{G}$ to which i and j participate. Since i and j are not connected directly in $\mathscr{G}$, there must be other nodes of $\mathscr{G}$ that help form the cycle, e.g., consider the cycle $i\leftrightarrow k\leftrightarrow j\leftrightarrow l\leftrightarrow i$ using other nodes k and l of $\mathscr{G}$. Now, nodes of $\mathscr{G}$ are links of $\mathscr{W}$ and, under the node-exclusive interference model, links of $\mathscr{W}$ are connected nodes in $\mathscr{G}$ if they are incident to the same node (device) of $\mathscr{W}$. Thus, using the previous example, the cycle $i\leftrightarrow k\leftrightarrow j\leftrightarrow l\leftrightarrow i$ in $\mathscr{G}$ implies a cycle in $\mathscr{W}$: $d_1\overset{k}{\leftrightarrow}d_2\overset{j}{\leftrightarrow}d_3\overset{l}{\leftrightarrow}d_4\overset{i}{\leftrightarrow}d_1$, where $d_1,\dots,d_4$ are wireless nodes of $\mathscr{W}$. It is clear that we have arrived at a contradiction since $\mathscr{W}$ is a tree network and does not contain any cycles. This establishes our desired property.

It now follows that any nodes i and j of $\mathscr{G}$ that are not connected can be colored with the same color. Note that this would not have been the case if we could find an odd-length cycle $i\leftrightarrow k\leftrightarrow j\leftrightarrow l\leftrightarrow m\leftrightarrow i$ in $\mathscr{G}$. In this case, we can not use the same color for any two nodes of $\mathscr{G}$ that are not connected. For instance, if we use color c_1 for i and j and color c_2 for k and l, then we can use neither c_1 nor c_2 for m even though m is not connected with k and j. As a result, and since nodes that are not connected can use the same color, the clique number of $\mathscr{G}$ is equal to the chromatic number of $\mathscr{G}$ and this is true for all induced graphs of $\mathscr{G}$. This implies that $\mathscr{G}$ is a perfect graph. ■

F. Proof of Lemma V.1

Proof: Due to [21, Cor. 64.9a] the 1's in $\mathbf{x}^*$ are correct, namely, there is always an optimal solution $\tilde{\mathbf{x}}$ to MWIS such that for $\tilde{x}_i=x_i^*$, for any $i\in\mathscr{V}$ with $x_i^*=1$. Next we show that the 0's in $\mathbf{x}^*$ are also correct. Suppose we have some $j\in\mathscr{V}$ such that $x_j^*=0$. Given that $w_i\geq 0$ for all $i\in\mathscr{V}$, we distinguish between two cases:
(a) $w_j>0$. In this case there must exist some $k\in\mathscr{N}_j$ with $x_k^*=1$; otherwise we can set x_j^* to 1 and achieve a strictly higher objective value for problem (6) while maintaining feasibility, which violates the fact that $\mathbf{x}^*$ is optimal for (6). As a result $\tilde{x}_j=0$ due to the feasibility constraint.
(b) $w_j=0$. In this case we can simply set $\tilde{x}_j=0$ without changing the objective value and maintain feasibility. ■

G. Proof of Theorem V.2

Proof: At the end of each iteration n, the still undetermined node with the largest weight (or smallest ID in its neighborhood if there is a tie) will necessarily set itself to 1. This implies that the number of undetermined nodes will strictly decrease in every iteration. The feasibility of the solution obtained is clear by construction. ■

H. Proof of Theorem V.3

Proof: Let $\mathscr{S}^{(n)}\subseteq\mathscr{V}$ be the set of all the nodes with the correct root information at the beginning of iteration n, that is, $\mathscr{S}^{(n)}=\{i|r_i^{(n-1)}=r,s_i^{(n-1)}=w_r,\forall i\in\mathscr{V}\}$. Clearly, $\mathscr{S}^{(0)}=\{r\}$. At iteration n, if the algorithm has not stopped, there exists at least one link $(i,j)\in\mathscr{E}$ such that $i\in\mathscr{S}^{(n)}$, $j\in\mathscr{V}\backslash\mathscr{S}^{(n)}$ and $\mathscr{S}^{(n)}\neq\mathscr{V}$. According to the algorithm in Fig. 3, $r_j^{(n)}=r_i^{(n-1)}=r$ and $s_j^{(n)}=s_i^{(n-1)}=w_r$, respectively, and thus we have $\{j\}\cup\mathscr{S}^{(n)}\subseteq\mathscr{S}^{(n+1)}$. As a result, $|\mathscr{S}^{(n)}|$ is strictly increasing in n. Since $|\mathscr{S}^{(0)}|=1$ and the maximum possible value of $|\mathscr{S}^{(n)}|$ is N for any n, in at most $N-1$ iterations $r_i^{(n)}=r$ for any $i\in\mathscr{V}$. ■

I. Proof of Theorem V.5

Proof: We have already argued that $\hat{\mathbf{x}}$ is feasible. Recall that $\boldsymbol{\theta}=(\theta_{ij};\ \forall(i,j)\in\mathscr{E})$ obtained from the gradient projection method is optimal for the dual of the edge relaxation. We now proceed to show $\hat{\mathbf{x}}$ is also optimal by checking the complementary slackness conditions, namely,

$$(x_i + x_j - 1)\theta_{ij} = 0, \quad (14)$$

$$(\textstyle\sum_{j\in\mathscr{N}_i} \theta_{ij} - w_i)x_i = 0, \quad (15)$$

for any $(i,j) \in \mathscr{E}$, where θ_{ij} is the dual variable associated with edge (i,j). Note that $(\mathbf{x}^*, \boldsymbol{\theta}^*)$ is primal-dual optimal for the edge relaxation and thus satisfies complementary slackness. Since we never change $\boldsymbol{\theta}^*$ during the course of the estimation algorithm, what remains to be shown is that $(\hat{\mathbf{x}}, \boldsymbol{\theta}^*)$ also satisfies these conditions.

Let us first consider condition (14). For any edge $(i,j) \in \mathscr{V}$, if $\theta^*_{ij} = 0$ this condition is always satisfied for any $\hat{x}_i$ and $\hat{x}_j$. As such we will focus on the case where we have $\theta^*_{ij} > 0$ and by complementary slackness it must hold that $x^*_i + x^*_j = 1$. Let us distinguish between three cases:
(a) $x^*_i = 1$ and $x^*_j = 0$. In this case due to the construction of $\hat{\mathbf{x}}$, we have $\hat{x}_i = 1$ and $\hat{x}_j = 0$, and as a result $(\hat{x}_i + \hat{x}_j - 1)\theta^*_{ij} = 0$.
(b) $x^*_i = 0$ and $x^*_j = 1$. In this case the argument is almost identical with the first case and the same conclusion follows.
(c) x^*_i and $x^*_j \in (0,1)$. Note that in this case we have $x^*_k < 1$ for any node $k \in \mathscr{N}_i \cup \mathscr{N}_j$ since $\mathbf{x}^*$ is a feasible solution. Given that $\mathscr{G}$ is a bipartite graph, without loss of generality let us assume $c_i = 1$ and $c_j = 2$. Following the estimation algorithm in Fig. 4, we set $\hat{x}_i = 1$ during the first iteration and $\hat{x}_j = 0$ during the second, which results in $\hat{x}_i + \hat{x}_j - 1 = 0$ and $(\hat{x}_i + \hat{x}_j - 1)\theta^*_{ij} = 0$.

Next we proceed to show that $(\hat{\mathbf{x}}, \boldsymbol{\theta}^*)$ also satisfies condition (15). Consider any node $i \in \mathscr{V}$ and note that if $\sum_{j\in\mathscr{N}_i} \theta^*_{ij} - w_i = 0$, then this equality always holds for any $\hat{x}_i$. If $\sum_{j\in\mathscr{N}_i} \theta^*_{ij} - w_i \neq 0$, it has to be true that $x^*_i = 0$ by complementary slackness. It follows that $\hat{x}_i = 0$ due to the construction of $\hat{\mathbf{x}}$, and consequently $(\sum_{j\in\mathscr{N}_i} \theta^*_{ij} - w_i)\hat{x}_i = 0$.

We conclude that $\hat{\mathbf{x}}$ is optimal for the edge relaxation and since it is integer (by construction) it must be optimal for the MWIS. ∎

J. Proof of Lemma V.6

Proof: We will use contradiction. Suppose $\hat{\mathbf{x}}$ is not a maximal independent set. Thus, there exists an $i \in \{1,\ldots,N\}$ with $\hat{x}_i = 0$ and such that $\hat{x}_j = 0$ for all links $j \in \mathscr{N}_i$. Only two scenarios could result in this, namely, *(a)* $x^*_i = 0$ and $x^*_j = 0$ for all $j \in \mathscr{N}_i$, or *(b)* $0 < x^*_i < 1$ and the estimation algorithms set both $\hat{x}_i$ to 0 and $\hat{x}_j$ to 0 for all $j \in \mathscr{N}_i$. Next we proceed to prove that neither of these two scenarios could happen.
(a) If $x^*_i = 0$ and $x^*_j = 0$ for all $j \in \mathscr{N}_i$, we can construct another relaxed feasible solution $\tilde{\mathbf{x}}$ by letting $\tilde{x}_i = 1$ and $\tilde{x}_k = x^*_k$ for all $k \in \mathscr{N}_i$. Since $w_i > 0$ (recall we have assumed this for all nodes without loss of generality), clearly $\sum_{k=1}^N w_k \tilde{x}_k > \sum_{k=1}^N w_k x^*_k$ and this contradicts the fact that $\mathbf{x}^*$ is optimal for the relaxation (6).
(b) If $0 < x^*_i < 1$, then according to either estimation Algorithm in Fig. 2 or Fig. 4, $\hat{x}_i$ is set to 0 only if some node $j \in \mathscr{N}_i$ is set to 1, which contradicts the assumption that $\hat{x}_j$ is set to 0 for all $j \in \mathscr{N}_i$. ∎

REFERENCES

[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automat. Contr.*, vol. 37, no. 12, pp. 1936–1948, 1992.
[2] S. Sanghavi, D. Shah, and A. Willsky, "Message passing for maximum weight independent set," *IEEE Trans. Inform. Theory*, vol. 51, no. 11, pp. 4822–4834, 2009.
[3] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *Proceedings of the joint international conference on Measurement and modeling of computer systems (ACM SIGMETRICS/IFIP Performance*, June 2006, pp. 27–38.
[4] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," in *Proceedings of the INFOCOM Conference*, 1998, pp. 533–539.
[5] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," in *Proceedings of the ACM SIGMETRICS*, 2007, pp. 313–324.
[6] X. Lin and S. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," in *Proceedings of the IEEE CDC conference*, San Diego, California, 2006.
[7] S. Sarkar and S. Ray, "Arbitrary throughput versus complexity tradeoffs in wireless networks using graph partitioning," *IEEE Trans. Automat. Contr.*, vol. 53, no. 10, pp. 2307–2323, 2008.
[8] C. Joo, X. Lin, and N. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 17, no. 4, pp. 1132–1145, 2009.
[9] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: Queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks," in *Proceedings of the IEEE INFOCOM*, 2010, pp. 1–5.
[10] A. Gupta, X. Lin, and R. Srikant, "Low-complexity distributed scheduling algorithms for wireless networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1846–1859, 2009.
[11] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Trans. Networking*, vol. 18, no. 3, pp. 960–972, 2010.
[12] M. Leconte, J. Ni, and R. Srikant, "Mixing time of Glauber Dynamics with parallel updates and heterogeneous fugacities," Computing Research Repository, Tech. Rep., 2010, http://arxiv.org/abs/1004.0397.
[13] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: an efficient randomized protocol for contention resolution," in *SIGMETRICS '09: Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, 2009, pp. 133–144.
[14] J. Ghaderi and R. Srikant, "On the design of efficient CSMA algorithms for wireless networks," in *IEEE Conference on Decision and Control*, 2010, pp. 954–959.
[15] I. C. Paschalidis, W. Lai, and X. Song, "Optimized scheduled multiple access control for wireless sensor networks," *IEEE Trans. Automat. Contr.*, vol. 54, no. 11, pp. 2573–2585, 2009.
[16] I. C. Paschalidis, W. Lai, and D. Starobinski, "Asymptotically optimal transmission policies for large-scale low-power wireless sensor networks," *IEEE/ACM Trans. Networking*, vol. 15, no. 1, pp. 105–118, 2007.

[17] A. Kako, T. Ono, T. Hirata, and M. M. Halldsson, "Approximation algorithms for the weighted independent set problem," in *Graph-Theoretic Concepts in Computer Science, 31st International Workshop*, 2005, pp. 341–350.

[18] S. Basagni, "Finding a maximal weighted independent set in wireless networks," *Telecommunication Systems*, vol. 18, pp. 155–168, 2001.

[19] A. Czygrinow and M. Hanckowiak, "Distributed algorithms for weighted problems in sparse graphs," *Journal of Discrete Algorithms*, vol. 4, pp. 588–607, 2004.

[20] B. Gfeller and E. Vicari, "A randomized distributed algorithm for the maximal independent set problem in growth-bounded graphs," in *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, 2007, pp. 53–60.

[21] A. Schrijver, *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.

[22] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, "The strong perfect graph theorem," *Annals of Mathematics*, pp. 51–229, 2006.

[23] C. Berge, "Some classes of perfect graphs," Six Papers on Graph Theory, 1963, indian Statistical Institute, Calcutta.

[24] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, and K. Vušković, "Recognizing Berge graphs," *Combinatorica*, vol. 25, no. 2, pp. 143–186, 2005.

[25] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[26] T. Jebara, "Map estimation, message passing, and perfect graphs," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, ser. UAI '09. Arlington, Virginia, United States: AUAI Press, 2009, pp. 258–267. [Online]. Available: http://dl.acm.org/citation.cfm?id=1795114.1795145

[27] A. Kosowski and L. Kuszner, "Self-stabilizing algorithms for graph coloring with improved performance guarantees," in *Lecture Notes in Computer Science*, 2006, vol. 4029/2006, pp. 1150–1159.

[28] X. Wu, R. Srikant, and J. R. Perkins, "Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 595–605, 2007.

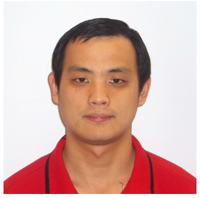
[29] A. Dimakis and J. Walrand, "Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits," *Advances in Applied Probability*, vol. 38, no. 2, pp. 505–521, 2006.

[30] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[31] J. Dunn, "Global and asymptotic convergence rate estimates for a class of projected gradient processes," *SIAM Journal on Control and Optimization*, vol. 19, no. 3, pp. 368–400, 1981.

Ioannis Ch. Paschalidis (M'96–SM'06–F'14) received the M.S. and Ph.D. degrees both in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1993 and 1996, respectively.

In September 1996 he joined Boston University where he has been ever since. He is a Professor and Distinguished Faculty Fellow at Boston University with appointments in the Department of Electrical and Computer Engineering, the Division of Systems Engineering, and the Department of Biomedical Engineering. He is the Director of the Center for Information and Systems Engineering (CISE). He has held visiting appointments with MIT and Columbia University, New York, NY, USA. His current research interests lie in the fields of systems and control, networking, applied probability, optimization, operations research, computational biology, and medical informatics.



Fuzhuo Huang received the B.S. and M.S. degrees from Tsinghua University, Beijing, China, in 2006 and 2008, respectively, and the Ph.D. degree in Systems Engineering from Boston University, Boston, MA, USA, in 2013. He currently is with Google, Inc. focusing on optimization algorithms in Google Advertisement products. His research interests include optimization theory, operations research, algorithms, and wireless sensor networks.

Wei Lai received the B.E. degree in Automatic Control and the M.E. degree in Systems Engineering, both from the Huazhong University of Science and Technology, China, in 1999 and 2001, respectively, and the Ph.D. degree in Systems Engineering from Boston University in 2007. He has been a senior analyst at Ameriprise Financial since 2007. His research is primarily focused on optimization.