LP-relaxation based Distributed Algorithms for Scheduling in Wireless Networks

Chandramani Singh, Angelia Nedić and R. Srikant
Coordinated Science Lab
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
Email: {cksingh, angelia, rsrikant}@illinois.edu

Abstract—**LP relaxations of Maximum Weighted Independent Set (MWIS) problems have been widely studied. A key motivation for this prior work comes from the central role that MWIS plays in designing throughput-optimal algorithms for wireless networks. However, to the best of our knowledge, the actual packet delay performance of these algorithms has not been studied in the context of wireless networks. In this paper, we first present an algorithm for solving the LP relaxation of MWIS which exhibits faster convergence to an optimal solution. Further, we show that one does not have to wait for infinite time for convergence to occur, but a simple rounding technique can be used to identify the ON/OFF states of the wireless links in finite time. As in prior work, such an approach only identifies the optimal MWIS states of some of the links in the network. Therefore, we present a scheme to combine this solution with Q-CSMA. Simulations indicate that the proposed scheme significantly improves the performance of Q-CSMA. Further, the proposed algorithm is shown to perform much better than previously suggested LP relaxation schemes due to its superior convergence properties.**

I. Introduction

In wireless networks with limited resources (spectrum, power etc.), it is important to efficiently utilize the resources, and also to fairly allocate them among the competing wireless nodes. Transmission by a node causes interference to its neighboring receivers. On the other hand, successful reception at a node requires that the *signal to interference plus noise ratio* (SINR) at that node exceed a certain threshold. Thus, when two neighboring nodes transmit simultaneously, their data packets may not be successfully decoded at their corresponding receivers. In this case, we say that these packets have collided with each other. A scheduling algorithm, or a *medium access control* (MAC) protocol, determines which links (or, the corresponding transmitters) can access the medium in each time slot so that no two packets collide with each other. In this paper, we study distributed scheduling algorithms for wireless networks. We want the scheduling algorithm to be distributed as centralized algorithms do not scale well (complexity/overhead wise) as the network grows, and also because in many wireless networks (e.g., in wireless sensor networks) there is no centralized controlling entity.

The second author is also with the Department of Industrial and Enterprise Systems Engineering. The third author is also with the Department of Electrical and Computer Engineering.

978-1-4799-3360-0/14/$31.00 ⓒ 2014 IEEE

Feasible schedules are often modeled as the independent sets in the conflict graph of communication links. We adopt such a model in this paper. Under *maximum weight scheduling* (MWS), the nodes in the conflict graph are assigned weights (typically queue lengths or functions thereof), and at each instant, a *maximum weighted independent set* (MWIS) is scheduled. In a pioneering work, the authors in [1] showed that queue-length based MWS scheduling algorithms are throughput optimal. However, MWS requires solving MWIS problem, a complex combinatorial optimization problem, in each time slot and hence is not implementable in practice.

Maximal scheduling, which chooses a maximal schedule in each time slot, is a low complexity alternative to MWS. In particular, *greedy maximal scheduling* (GMS), also known as *longest queue first scheduling* (LQFS), which factors in queue length information while choosing a maximal schedule, shows good throughput and delay performance for a few classes of network topologies. In fact, LQFS is shown to be throughput optimal if the topology satisfies a *local-pooling* condition [2]. But in general networks, these algorithms achieve only a fraction of the capacity regions [3]. There is a host of other policies that differ in their complexity and the fraction of the maximum throughput region they can achieve [4]–[6].

Another class of MAC protocols are distributed random access protocols named *Carrier sense multiple access* (CSMA) protocols which are widely used due to their simplicity and low overhead. Under a typical CSMA protocol, a transmitter senses whether the channel is free or not. If the channel is busy, the node backs off for a random period of time, otherwise it transmits with some probability. It was shown in [7] that CSMA can be modeled as a continuous time Markov chain. It was also shown that the evolution of the schedules has a product-form stationary distribution. The authors in [8] developed non-adaptive random access algorithms that achieve the same performance as *weighted fair maximal* (WFM) scheduling schemes. An adaptive CSMA algorithm was developed in [9], in which the links adjust the protocol parameters based on the local traffic measurements. It was shown that the proposed algorithm is *throughput optimal* under a time scale separation assumption whereby the CSMA Markov chain converges to its stationary distribution instantaneously compared to the time scale of adaptation of the CSMA parameters. Later the throughput optimality of the adaptive CSMA algorithm was established without the time-scale separation assumption by choosing the CSMA parameters to be slowly increasing functions of the queue lengths (of the form $\log\log(Q)$) [10].

In essence, these CSMA algorithms execute the so-called *Glauber dynamics* that samples over independent sets of the conflict graph. Performance of these algorithms depends on how fast the Glauber dynamics converges to the stationary distribution of the CSMA Markov chain. Although the CSMA parameters (i.e., the weights) of the form $\log\log(Q)$ stabilize the network, the resulting dynamics reacts very slowly to changes in the queue lengths, which leads to poor delay performance. It was shown in [11] that throughput optimality of adaptive CSMA continues to hold even with weight functions of the form $\log(Q)/g(Q)$, where $g(Q)$ can be an arbitrarily slowly increasing function. This variant of CSMA algorithms exhibits much better delay performance. In an enhancement of Glauber dynamics, developed in [12], the authors allowed multiple links to update their states in the same slot. They termed the algorithm as Q-CSMA and also proposed *hybrid* Q-CSMA, combining GMS and Q-CSMA algorithms. In another work [13], the authors refined Q-CSMA based on nodes' degrees to further improve its delay performance. A parametric version of Glauber dynamics (parameterized by $\alpha \in [0, 1]$) was proposed in [14], which ranges from the standard Glauber dynamics ($\alpha = 0$) to the Metropolis algorithm ($\alpha = 1$). It was shown that the algorithms are throughput optimal for all $\alpha \in [0, 1]$ but larger values of α achieve smaller delay.

In spite of these modifications, the delay performance of CSMA protocols is not satisfactory. A potential solution to this problem is combining the MWIS and CSMA based approaches. More precisely, the LP relaxation of MWIS problem is solved to settle the states of a fraction of the nodes, and the rest execute Glauber dynamics. The rationale behind this approach is the observation that mixing time of Glauber dynamics is exponential in the size of the graph. Hence executing Glauber dynamics over a smaller graph leads to a faster mixing, and hence to smaller average queue lengths. Nonetheless, a distributed implementation of the overall algorithm entails distributed solution of MWIS problem. In the following we briefly review a few recent approaches to accomplish this.

The authors in [15] showed that MWIS problem can be modeled as a maximum *a posteriori* (MAP) estimation problem, and they proposed a belief propagation algorithm. This algorithm, when it converges, solves the LP relaxation of the MWIS problem. The authors also showed that a simple modification of this algorithm becomes a coordinate descent algorithm for the dual problem. The latter algorithm yields a solution of the LP relaxation, and thus also gives solution of the MWIS problem when the graph is bipartite and has unique MWIS. Recently, a message passing based two-phase algorithm was proposed to estimate a solution of MWIS problem [16]. The first phase is a gradient projection algorithm that solves the LP relaxation of MWIS problem. The second phase uses the LP solution to yield a MWIS in case of bipartite graphs and a maximal schedule in general.

A. Our Contribution

The following is a preview of our contribution.

1) We present an algorithm to solve the LP relaxation of MWIS problem, which lends itself to a local message passing based implementation. The algorithm is significantly faster than the constant step size gradient projection methods (e.g., the one used in [16]).
2) We propose a new scheduling algorithm combining the LP relaxation solution and Glauber dynamics. In the proposed algorithm, the solution of the LP relaxation identifies the optimal states of a subset of the nodes, and Glauber dynamics is run over the remaining nodes, which results in faster mixing, and hence better delay performance of the hybrid algorithm.
3) We show that one does not need to solve the LP relaxation of MWIS problem exactly in order to estimate the integral part of the solution which is required in our proposed algorithm. To show this, we use the structure of fractional independent set polytopes and finiteness of the conflict graphs under consideration.
4) We also propose an algorithm for bipartite graphs, that *almost surely* extracts a solution of the MWIS problem from a solution of its LP relaxation. Our algorithm is probabilistic, and also yields an alternative proof of correctness of the coloring based deterministic estimation algorithm of [16].

II. Wireless Scheduling Problem

A. Network Model

We model a wireless network via a set of links $\mathcal{V}$ and their interference constraints. Each link is a transmitter-receiver pair in which the receiver can successfully decode the corresponding transmitter's signal in the absence of interference from other transmitters. Each link $i \in \mathcal{V}$ is associated with a set $\mathcal{N}_i$ of interfering (or, conflicting) links defined as follows; transmission over link i cannot be successfully decoded (at its recceiver) if there a simultaneous transmission over any link $j \in \mathcal{N}_i$. We assume that the interference constraints are symmetric: for any $i, j \in \mathcal{V}$, if $i \in \mathcal{N}_j$ then $j \in \mathcal{N}_i$. The links and their interference relations together define the so called conflict graph $(\mathcal{V}, \mathcal{E})$, where, for $i, j \in \mathcal{V}$, $(i, j) \in \mathcal{E}$ if and only if $j \in \mathcal{N}_i$ (or equivalently, $i \in \mathcal{N}_j$). *From here onwards we refer to the links as the nodes of the conflict graph.*

A schedule is a set of nodes whose transmitters transmit simultaneously and the corresponding receivers can still successfully decode the respective signals. Clearly, any schedule is an independent set in the conflict graph $(\mathcal{V}, \mathcal{E})$. A schedule can be represented by a vector $x \in \{0, 1\}^{|\mathcal{V}|}$ where $x_i = 1$ if and only if i belongs to the schedule. In the following, we refer to x itself as a schedule.

We consider a time-slotted system with a discrete time packet arrival process at each node. Let $a_i(t)$ denote the number of packets arriving at node i in time-slot t. We assume, for each node $i \in \mathcal{V}$, that $\{a_i(t)\}$ is an independent and identically distributed process over time and also that $\mathbb{E}[a_i(t)] = \lambda_i$. We use $x(t) \in \{0, 1\}^{|\mathcal{V}|}$ to denote the schedule used in time slot t, and we assume that one packet could be transmitted over each scheduled link. Finally, let $q_i(t), i \in \mathcal{V}$ be the queue lengths of the nodes at the end of slot t. The queue lengths evolve as

$$q_i(t) = [q_i(t-1) - x_i(t)]_+ + a_i(t) \tag{1}$$

for all $i \in \mathcal{V}$ and t, where $[z]_+ = \max(z, 0)$ for any $z \in \mathbb{R}$.

B. Problem Formulation

A scheduling algorithm is a policy to determine which schedule to use in each time slot. The capacity region of the network is the set of all arrival rates λ for which there exists a scheduling algorithm that can stabilize the queues. We are interested in distributed algorithms that are throughput optimal and also result in small queue lengths. Due to space limitations, we do not define terms such as throughput optimality and queue stability here. These are well-established terms, e.g., see [11], [12].

III. LP-BASED SCHEDULING ALGORITHMS

A. LP Relaxation of the MWIS Problem

Let us consider the conflict graph $(\mathcal{V}, \mathcal{E})$, and let us associate a positive weight w_i with each node $i \in \mathcal{V}$. The MWIS problem is the one where we want to find an independent set of $(\mathcal{V}, \mathcal{E})$ that is optimum with respect to sum of the weights of the constituent nodes. Formally, the MWIS problem is the following binary integer program.

$$\begin{aligned} \text{maximize} \quad & U(y) = \sum_{i \in \mathcal{V}} w_i y_i, \\ \text{subject to} \quad & y_i + y_j \leq 1 \text{ for all } (i,j) \in \mathcal{E}, \\ & y_i \in \{0, 1\} \text{ for all } i \in \mathcal{V}. \end{aligned} \tag{2}$$

We propose an algorithm to solve the LP relaxation of the MWIS problem, which is obtained by replacing the integrality constraints with the constraints $0 \leq y_i \leq 1, i \in \mathcal{V}$.

$$\begin{aligned} \text{maximize} \quad & U(y), \\ \text{subject to} \quad & y_i + y_j \leq 1 \text{ for all } (i,j) \in \mathcal{E}, \\ & 0 \leq y_i \leq 1 \text{ for all } i \in \mathcal{V}. \end{aligned} \tag{3}$$

Remark 3.1: Problem 3 has the following properties (see [17, Chapter 64] for proof).

1) The LP polytope defined by the constraints of (3) always has half-integral vertices (i.e., the vertices have components 0,1 or $\frac{1}{2}$), and has integral vertices in the case of bipartite graphs. Hence, if the conflict graph is bipartite and Problem (3) has a unique solution, it is also the solution of the MWIS problem. However, this uniqueness does not always hold, and we do not need this in our proofs although it has been used in other works (e.g., [15]).
2) For any graph and for any optimal solution $\tilde{y}$ of (3), there exists a solution y^* of the MWIS problem such that $y_i^* = \tilde{y}_i$ for all i with $\tilde{y}_i \in \{0, 1\}$. We use this property in proving optimality of the proposed hybrid policy.

We solve Problem (3) using the penalty function method. Towards this, we add a quadratic penalty function to the objective function. This yields the following concave optimization problem.

$$\begin{aligned} \text{maximize} \quad & U_\epsilon(y) = \sum_{i \in \mathcal{V}} w_i y_i - \frac{\epsilon}{2} \sum_{i \in \mathcal{V}} y_i^2, \\ \text{subject to} \quad & y_i + y_j \leq 1 \text{ for all } (i,j) \in \mathcal{E}, \\ & 0 \leq y_i \leq 1 \text{ for all } i \in \mathcal{V}, \end{aligned} \tag{4}$$

where $\epsilon > 0$ is a penalty (or perturbation) parameter. Its dual objective can be written as follows. For $\theta \in \mathbb{R}_+^{|\mathcal{E}|}$,

$$\begin{aligned} V_\epsilon(\theta) &= \max_{y \in [0,1]^{\mathcal{V}}} \sum_{i \in \mathcal{V}} w_i y_i - \frac{\epsilon}{2} \sum_{i \in \mathcal{V}} y_i^2 + \sum_{(i,j) \in \mathcal{E}} \theta_{ij} (1 - y_i - y_j) \\ &= \sum_{i \in \mathcal{V}} \max_{y_i \in [0,1]} g_i(y_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}, \end{aligned}$$

where

$$g_i(y_i) = y_i \left(w_i - \sum_{j \in \mathcal{N}_i} \theta_{ij} \right) - \frac{\epsilon}{2} y_i^2.$$

The following lemma characterizes the solutions $y_i(\theta)$ that optimize $g_i(y_i)$ for all $i \in \mathcal{V}$.

Lemma 3.1: For all $i \in \mathcal{V}$, the unique maximizer $y_i(\theta)$ of $g_i(y_i)$ is given by[1]

$$y_i(\theta) = \begin{cases} 1 & \text{if } \sum_{j \in \mathcal{N}_i} \theta_{ij} \leq w_i - \epsilon \\ \frac{w_i - \sum_{j \in \mathcal{N}_i} \theta_{ij}}{\epsilon} & \text{if } w_i - \epsilon < \sum_{j \in \mathcal{N}_i} \theta_{ij} < w_i \\ 0 & \text{if } \sum_{j \in \mathcal{N}_i} \theta_{ij} \geq w_i. \end{cases} \tag{5}$$

We do not provide a proof of this Lemma as it is straightforward to verify. Following some simple algebra, we obtain

$$g_i(y_i(\theta)) = \begin{cases} w_i - \sum_{j \in \mathcal{N}_i} \theta_{ij} - \frac{\epsilon}{2} & \text{if } \sum_{j \in \mathcal{N}_i} \theta_{ij} \leq w_i - \epsilon \\ \frac{(w_i - \sum_{j \in \mathcal{N}_i} \theta_{ij})^2}{2\epsilon} & \text{if } w_i - \epsilon < \sum_{j \in \mathcal{N}_i} \theta_{ij} < w_i \\ 0 & \text{if } \sum_{j \in \mathcal{N}_i} \theta_{ij} \geq w_i. \end{cases}$$

Further, the dual optimization problem can be written as

$$\begin{aligned} \text{maximize} \quad & V_\epsilon(\theta) = \sum_{i \in \mathcal{V}} g_i(y_i(\theta)) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}, \\ \text{subject to} \quad & \theta_{ij} \geq 0 \text{ for all } (i,j) \in \mathcal{E}. \end{aligned} \tag{6}$$

The partial derivative of $g_i(y_i(\theta))$ with respect to θ_{jk} is

$$\frac{\partial g_i(y_i(\theta))}{\partial \theta_{jk}} = \begin{cases} -y_i(\theta) & \text{if } j = i, k \in \mathcal{N}_i \text{or if } k = i, j \in \mathcal{N}_i \\ 0 & \text{otherwise.} \end{cases}$$

Thus we conclude that

$$\frac{\partial V_\epsilon(\theta)}{\partial \theta_{ij}} = 1 - y_i(\theta) - y_j(\theta). \tag{7}$$

In particular, the function $V_\epsilon(\theta)$ is continuously differentiable. By [18, Lemma 3.1], it turns out that $\nabla V_\epsilon(\theta)$ is also Lipschitz continuous over $\mathbb{R}_+^{|\mathcal{E}|}$ with a Lipschitz constant $\tilde{L}_\epsilon = \frac{2 \max_{i \in \mathcal{V}} d_i}{\epsilon}$. One can use the optimal gradient method of [19] to solve Problem (6). It would require knowledge of the Lipschitz constant. We instead employ the optimal scaled gradient projection (OGP) algorithm proposed in [18,

[1] If $w_i - \epsilon < 0$, the first case in the right hand side of (5) does not arise.

Section 3.2].[2] We present the exact algorithm at node i as Algorithm OGP$(\epsilon, t^0, \theta^i)$ - here θ^i and η^i denote $(\theta_{ij}, j \in \mathcal{N}_i)$ and $(\eta_{ij}, j \in \mathcal{N}_i)$ respectively.[3]

Algorithm OGP $(\epsilon, t^0, \theta^i)$

initialize $t = t^0, \beta(t) = 1, \theta^i(t) = \theta^i, \eta^i(t+1) = \theta^i(t),$
repeat
$t = t + 1,$
$\beta(t) = \frac{1+\sqrt{1+4\beta^2(t-1)}}{2},$
$\theta_{ij}(t) = \left[\eta_{ij}(t) - \frac{\epsilon(1-y_i(\eta(t))-y_j(\eta(t)))}{d_i+d_j}\right]_+$ for all $j \in \mathcal{N}_i,$
$\eta_{ij}(t+1) = \theta_{ij}(t) + \left(\frac{\beta(t-1)-1}{\beta(t)}\right)(\theta_{ij}(t) - \theta_{ij}(t-1))$ for all $j \in \mathcal{N}_i,$
until $\max_{(i,j)\in\mathcal{E}} |\theta_{ij}(t) - \theta_{ij}(t-1)| < \delta.$

The algorithm terminates when, for all $(i, j) \in \mathcal{E}$, the successive iterates of θ_{ij} are within δ distance for some small $\delta > 0$. Although checking this criterion seems to require global coordination among all the nodes, it can be be done using a distributed consensus algorithm (see [20, Page 24]).

Observe the subtle difference with respect to the standard gradient projection algorithms that, here, the updates of $\{\eta_{ij}(t)\}$ are obtained as linear combinations of the *two* previous iterates; setting $\beta(t) = 1$ for all t would give the standard algorithm. Moreover, we perform a scaled gradient descent step in each iteration. See Section IV-A for a detailed analysis of this algorithm.

Remark 3.2: Algorithm OGP$(\epsilon, t^0, \theta^i)$ relies on message passing among neighboring nodes for computation of the iterates $\theta_{ij}(t)$. In particular, neighboring nodes share their degrees (d_is), and also exchange their $y_i(\eta(t))$ values in each slot. Let us also observe that for any $(i, j) \in \mathcal{E}$, nodes i's and j's updates of $\eta_{ij}(t)$ and $\theta_{ij}(t)$ are always consistent.

After Algorithm OGP$(\epsilon, t^0, \theta^i)$ terminates, we reduce the value of ϵ and restart the algorithm using the current value of θ^i. More precisely, we start with $\epsilon = \epsilon^0$ and repetitively run Algorithm OGP$(\epsilon, t^0, \theta^i)$, setting $\epsilon = \alpha\epsilon$ (for an $\alpha \in (0, 1)$) after each run. We will show in Section IV that we only require to reduce ϵ up to a certain value ϵ^* for accurate estimation of integral components in a solution of Problem (3).

B. The Proposed Solution

Our proposed algorithm is based on combining the solution of the LP relaxation of the MWIS problem (3) with Q-CSMA algorithms. We occasionally solve the LP relaxation of the MWIS problem with nodes' weights being functions of respective queue-lengths; $w_i(t) = f(q_i(t))$ for all $i \in \mathcal{V}$.[4]

[2] While the Lipschitz constant can be broadcast to all the nodes using a consensus algorithm, the optimal scaled gradient projection method, apart from bypassing this requirement, also exhibits a better convergence rate (see [18]). This algorithm is called optimal because it yields best convergence rate for the class of problems under consideration (see [19]).

[3] Observe from (5) that $y_i(\eta)$ depends on η through η^i only.

[4] We use the same weight functions in the MWIS problem and the Q-CSMA algorithm, which are motivated by convergence properties of Glauber dynamics.

An LP solution y consists of integer values for a subset of nodes and fractional values for the rest; say $y_i \in \{0, 1\}$ for all $i \in \mathcal{V}_I$ and $y_i \in (0, 1)$ for all $i \in \mathcal{V} \setminus \mathcal{V}_I$. We propose that the nodes in $\mathcal{V}_I$ act in accordance with the LP solution while the nodes in $\mathcal{V} \setminus \mathcal{V}_I$ execute the Q-CSMA algorithm until the next solution of the MWIS problem is obtained. In the following, we first describe the Q-CSMA algorithm. We then formally describe the proposed algorithm.

1) The Q-CSMA Algorithm: The Q-CSMA algorithm executes Glauber dynamics which samples over the collection of independent sets $\mathcal{X}$ of the conflict graph. In each slot t, a non-conflicting set of nodes is chosen to update their states. A chosen node i's state transition probability depends on its weight $w_i(t) = f(q_i(t))$. The steady state of this so called parallel Glauber dynamics favors schedules with larger aggregate weights. A formal description of parallel Glauber dynamics is presented in Algorithm PGD.

Algorithm PGD (in time slot t)

Choose $b_i(t)$ from Uniform$([0, 1])$,
if $b_i(t) > b_j(t)$ for all $j \in \mathcal{N}_i$ **then**
if $x_j(t-1) = 0$ for all $j \in \mathcal{N}_i$ **then**
$x_i(t) = 1$ w.p. $\frac{\exp(w_i(t))}{1+\exp(w_i(t))}$ and 0 w.p. $\frac{1}{1+\exp(w_i(t))}$
else
$x_i(t) = 0$
end if
else
$x_i(t) = x_i(t-1).$
end if

Remark 3.3: Local message exchanges are needed for implementation of Algorithm PGD. In particular, the neighboring nodes exchange their $b_i(t)$ values in each slot, which facilitates comparison of these values as required to determine which nodes may update their states.

2) The proposed Algorithm: The proposed solution is formally presented in Algorithm OGP-GD. Here, we divide the available bandwidth into a *control channel* and a *data channel*. The LP relaxation of MWIS problem is solved over the control channel. More precisely, nodes pass messages required in the course of the optimal gradient projection algorithm over the control channel. After several runs of this algorithm for a chosen sequence of ϵs, a subset of nodes (those with $run\_gd_i = false$) act in accordance with its output. On the other hand, the nodes with $run\_gd_i = true$ execute Glauber dynamics over the data channel.[5]

Remark 3.4: Consider a node i for which $run\_gd_i = true$ in a time slot t. Then, in the time slot t, $run\_gd_i = true$ for all $j \in \mathcal{N}_i$ also. Thus, i and all $j \in \mathcal{N}_i$ generate $b_i(t)$ and $b_j(t)$ respectively, in time slot t, as required for the subsequent comparison for generation of decision schedule (see Algorithm PGD).

[5] Alternatively, we may divide each time slot into a *control slot* and a *data slot*. Then the messages required in the course of the optimal gradient projection algorithm are communicated during the control slot and the Glauber dynamics and the actual data transfer are executed during the data slot.

Algorithm OGP-GD

initialize $t = 0, x_i(0) = 0, run\_gd_i = true,$

// *Over the Data Channel*
loop
  $t = t + 1,$
  if $run\_gd_i = true$ **then**
    run Algorithm PGD
  else
    $x_i(t) = \tilde{y}_i$
  end if
end loop

// *Over the Control Channel*
loop
  $w_i = w_i(t), \epsilon = \epsilon_0$ and $\theta_{ij} = 0$ for all $j \in \mathcal{N}_i,$
  repeat
    run Algorithm OGP $(\epsilon, t^0, \theta^i),$
    $\epsilon = \alpha\epsilon,$ // $\alpha \in (0, 1)$
    $t^0 = t,$
    $\theta_{ij} = \theta_{ij}(t)$ for all $j \in \mathcal{N}_i$
  until $\epsilon < \epsilon^*$
  if $y_i(\theta(t)) \in \{0, 1\}$ **then**
    $run\_gd_i = false,$
    $\tilde{y}_i = y_i(\theta(t))$
  else
    $run\_gd_i = true$
  end if
end loop

Remark 3.5: Notice that we run parallel Glauber dynamics over a smaller network. It significantly reduces the mixing time of the CSMA Markov chain. Consequently, the proposed composite algorithm exhibits better delay performance than the plain parallel Glauber dynamics.

C. Clique Relaxation of the MWIS Problem

For bipartite graphs a significant fraction of components in the solution of LP relaxation (3) are either 0 or 1. In this case, we need to run Glauber dynamics over significantly smaller graphs, which leads to faster mixing of the CSMA Markov chain. On the other hand, simulation indicates that almost all the variables assume a value $1/2$ in solutions of Problem (3) for general (non-bipartite) graphs. A result showing this for *bicritical* graphs with equal node-weights was proved in [21] (also see [22]). Such a solution does not help in improving the performance of Q-CSMA. A potential solution to address this issue is to solve a tighter LP relaxation comprising of constraints corresponding to all the maximal cliques in the graph. Hence, we consider the following LP:

$$\begin{aligned} \text{maximize} \quad & U(y), \\ \text{subject to} \quad & \sum_{i \in C_k} y_i \leq 1 \text{ for all } C_k \in \mathcal{C}, \\ & 0 \leq y_i \leq 1 \text{ for all } i \in \mathcal{V}. \end{aligned} \tag{8}$$

where $\mathcal{C}$ is the set of all the maximal cliques in the network.

Remark 3.6: 1) The LP polytope defined by the constraints of (8) has integral vertices in the case of *perfect graphs*.[6] Hence if the graph is perfect and Problem (8) has a unique solution, it is also the solution of the MWIS problem (see [17, Chapter 65]).
2) In general, the number of maximal cliques can be exponential in the size of the graph. But for graphs with bounded node degrees, the number of maximal cliques increases only linearly with the number of nodes [23].

Problem (8) can also be solved using a dual based optimal gradient projection method. Its development is similar to that in Section III-A. For all $i \in \mathcal{V}$ and $\theta \in \mathbb{R}_+^{|\mathcal{C}|}$, we define

$$y_i(\theta) = \begin{cases} 1 & \text{if } \sum_{k:i \in C_k} \theta_k \leq w_i - \epsilon \\ \frac{w_i - \sum_{k:i \in C_k} \theta_k}{\epsilon} & \text{if } w_i - \epsilon < \sum_{k:i \in C_k} \theta_k < w_i \\ 0 & \text{if } \sum_{k:i \in C_k} \theta_k \geq w_i. \end{cases} \tag{9}$$

For all $i \in \mathcal{V}$, we define $\mathcal{C}_i$ to be the set of maximal cliques containing i and $c_i = |\mathcal{C}_i|$. Then, the exact algorithm run at node i is similar to Algorithm OGP$(\epsilon, t^0, \theta^i)$, but with the iterates $\{\theta^i(t) = (\theta_k(t), k : C_k \in \mathcal{C}_i)\}$ being updated as

$$\theta_k(t) = \left[\eta_k(t) - \frac{\epsilon(1 - \sum_{j \in C_k} y_j(\eta(t)))}{\sum_{j \in C_k} c_j} \right]_+,$$

where $y_j(\cdot)$s are now given by (9).

This algorithm can also be integrated with parallel Glauber dynamics. Observe that if neighboring nodes share their neighbor-lists, each node can identify all the maximal cliques containing it. As before, implementation of this algorithm also requires one-hop message exchanges only. See our technical report [24] for details.

D. Comparison to Existing LP Based Solutions

A widely studied class of scheduling algorithms are MWS algorithms in which the schedule $x(t)$ in each time slot t is a solution of the MWIS problem with weights $w_i(t) = f(q_i(t))$ for all $i \in \mathcal{V}$. The MWS algorithms are throughput optimal under certain constraints on weight functions [25]. Nonetheless, MWS requires solving the MWIS problem in each time slot and hence is not implementable in practice.

The authors in [16] proposed a two-phase algorithm that should be run in each time slot. In the first phase, the LP relaxation of the MWIS problem is solved using logarithmic penalties and a gradient projection method. Suppose that, for an LP solution y, $y_i \in \{0, 1\}$ for all $i \in \mathcal{V}_I$ and $y_i \in (0, 1)$ for all $i \in \mathcal{V} \setminus \mathcal{V}_I$. The nodes in $\mathcal{V}_I$ set their states as per the LP solution. In the second phase, the nodes in $\mathcal{V} \setminus \mathcal{V}_I$ estimate their states using either a coloring algorithm or parallel Glauber dynamics. All these algorithms entail multiple rounds of local message exchanges (to be precise, exponential in the size of the conflict graph) in each time slot, and thus again, cannot be implemented as such. An analogous algorithm based on the clique relaxations (8) was developed in [20, Chapter 4].

[6] A graph is perfect if in every induced subgraph, the chromatic number equals the maximal clique size. Bipartite graphs are perfect.

IV. Analysis of the Proposed Algorithms

A. Convergence of Algorithm OGP(ϵ, t^0, θ^i)

The objective function of the dual optimization problem (Problem (6)) can be written as

$$V_\epsilon(\theta) = \sum_{i \in \mathcal{V}} v_i(\theta^i) + \sum_{(i,j)\in\mathcal{E}} \theta_{ij}, \tag{10}$$

where $\theta^i = (\theta_{ij}, j \in \mathcal{N}_i)$ and $v_i(\theta^i) = g_i(y_i(\theta))$ for all $i \in \mathcal{V}$. Then $\nabla v_i(\theta^i)$ is Lipschitz continuous over $\mathbb{R}_+^{|\mathcal{N}_i|}$ with a Lipschitz constant $l_i = \frac{d_i}{\epsilon}$. Let H be $|\mathcal{E}| \times |\mathcal{E}|$ diagonal matrix with the diagonal element corresponding to edge (i,j) given by $l_i + l_j$. Then, the update equations for $(\theta_{ij}, (i,j) \in \mathcal{E})$ in Algorithm OGP(ϵ, t^0, θ^i) can be compactly written as

$$\theta(t) = \left[\eta(t) - H^{-1}\nabla V_\epsilon(\eta(t))\right]_+ . \tag{11}$$

We are thus executing a scaled gradient projection algorithm. We now study the convergence behavior of this algorithm.

Let $t^0 = 0, \theta(0) = \theta^0$ and $\theta(t), t \geq 0$ be the sequence of iterates generated by Algorithm OGP(ϵ, t^0, θ^i) and let θ^ϵ be an optimal solution of Problem (6). Let y^ϵ be the solution of Problem (4). Also, let $\langle \cdot, \cdot \rangle_H$ and $\|\cdot\|_H$ denote the scaled inner product and scaled norm respectively: $\langle \theta, \eta \rangle_H = \theta^T H \eta$ and $\|\theta\|_H^2 = \theta^T H \theta$ for all $\theta, \eta \in \mathbb{R}^{|\mathcal{E}|}$. The convergence properties of the algorithm are summarized in the following theorem.

Theorem 4.1: 1) $V_\epsilon(\theta(t)) - V_\epsilon(\theta^\epsilon) \leq \dfrac{2\|\theta^0 - \theta^\epsilon\|_H^2}{(t+1)^2}$,

2) $\|y(\theta(t)) - y^\epsilon\| \leq \dfrac{2\|\theta^0 - \theta^\epsilon\|_H}{\sqrt{\epsilon}(t+1)}$.

Proof: See Appendix A. ■

Remark 4.1: Notice that the dual objective function corresponding to Problem (8) can also be written in the form of (10). In this case, θ^i stands for $(\theta_k, k : i \in C_k)$. Also, $\nabla v_i(\theta^i)$ is Lipschitz continuous over $\mathbb{R}_+^{|\mathcal{C}_i|}$ with a Lipschitz constant $l_i = \frac{c_i}{\epsilon}$. The update equations for $(\theta_k, k \in \mathcal{C})$ can also be expresses as (11), with H a $|\mathcal{C}| \times |\mathcal{C}|$ diagonal matrix with the diagonal element corresponding to the maximal clique C_k given by $h_k = \sum_{j \in C_k} l_j$. With this notation, the convergence claims made in Theorem 4.1 apply to the clique relaxation based algorithm as well.

B. Termination of the LP Solution Algorithm

Our result in this section draws on an idea developed in [15]. Recall that the message passing algorithm over the control channel outputs the set $\mathcal{V}_I(\tilde{y}) = \{i : \tilde{y}_i \in \{0,1\}\}$ and the variables $\tilde{y}_i, i \in \mathcal{V}_I(\tilde{y})$, corresponding to an optimal solution $\tilde{y}$ of (3) (or, of (8) if we use the clique relaxation based algorithm). The proposed scheduling algorithm relies on correct estimation of these quantities. We now show that we need to run Algorithm OGP(ϵ, t^0, θ^i) (or, the one corresponding to clique relaxations) only upto a small enough value ϵ^*.

Since we consider a finite conflicting graph, we can find an $e(\tilde{y}) \in (0,1)$ corresponding to each optimal solution $\tilde{y}$ of (3) such that

$$\tilde{y}_i \in [e(\tilde{y}), 1 - e(\tilde{y})] \text{ for all } i \in \mathcal{V} \setminus \mathcal{V}_I(\tilde{y}).$$

For brevity, we suppress the argument $\tilde{y}$ in the following.

Let y^ϵ denote the unique optimal solution of (4). As $\epsilon \to 0$, y^ϵ converges to an optimal solution $\tilde{y}$ of (3). In particular, there exists an ϵ^* such that

$$|y_i^{\epsilon^*} - \tilde{y}_i| < e/4 \text{ for all } i \in \mathcal{V}.$$

Moreover, during the execution of Algorithm OGP($\epsilon^*, t^0, \theta^i$), there exists a time t^* such that

$$|y_i(\theta(t)) - y_i^{\epsilon^*}| < e/4 \text{ for all } i \in \mathcal{V}$$

for all $t \geq t^*$. Combining these two inequalities, for $t \geq t^*$,

$$|y_i(\theta(t)) - \tilde{y}_i| < e/2 \text{ for all } i \in \mathcal{V}.$$

We can thus set

$$\begin{aligned} \mathcal{V}_I &= \{i : y_i(\theta(t^*)) \in [0, e/2) \cup (1 - e/2, 1]\}, \\ \tilde{y}_i &= 0 \text{ if } y_i(\theta(t^*)) \in [0, e/2), \\ \text{and } \tilde{y}_i &= 1 \text{ if } y_i(\theta(t^*)) \in (1 - e/2, 1]. \end{aligned}$$

We have thus proved the following result.

Theorem 4.2: The estimates based on Algorithm OGP(ϵ, t^0, θ^i) are correct for any $\epsilon \leq \epsilon^*$.

Since ϵ geometrically approaches ϵ^*, we need $O(\log(\epsilon^0/\epsilon^*))$ iterations of Algorithm OGP(ϵ, t^0, θ^i).

Remark 4.2: Observe that we can set $e(\tilde{y}) = 1/2$ in the case of Problem (3) (i.e., for Algorithm OGP(ϵ, t^0, θ^i)) when it has unique solution.

C. A Result for Bipartite Graphs

There has been significant focus on solving MWIS for the special class of bipartite graphs (e.g., see [15], [16]). While a bipartite graph model is not very useful for wireless networks, it is often used as the standard model to illustrate properties of LP relaxation techniques [15], [16]. Hence, for the sake of completeness of the discussion of the relationship of our work to prior literature, we present an algorithm for bipartite graphs, that extracts the solution of MWIS problem from the solution of its LP relaxation with probability 1. Let us denote the graph by $(\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E})$ where, for all $(i,j) \in \mathcal{E}$, either $i \in \mathcal{V}_1$ and $j \in \mathcal{V}_2$ or vice versa. We assume that all the nodes have a common random number generator that generates a number uniformly from the interval $[0,1]$.

Let $\tilde{y}$ be a solution of the LP (3). Having $\tilde{y}$, a random number γ is generated and then a candidate solution $y^\gamma \in \{0,1\}^{|\mathcal{V}_1 \cup \mathcal{V}_2|}$ of the MWIS problem is constructed as follows:

$$y_i^\gamma = \begin{cases} 1 & \text{if } i \in \mathcal{V}_1 \text{ and } \tilde{y}_i > \gamma \\ & \text{or if } i \in \mathcal{V}_2 \text{ and } \tilde{y}_i > 1 - \gamma \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Theorem 4.3: With probability 1, the vector y^γ is a MWIS.

Proof: We first argue that y^γ is always an independent set for any $0 \leq \gamma \leq 1$. To see this, consider any $(i,j) \in \mathcal{E}$ with $i \in \mathcal{V}_1$ and $j \in \mathcal{V}_2$ without any loss of generality. Then $\tilde{y}_i + \tilde{y}_j \leq 1 = \gamma + 1 - \gamma$. Thus we cannot have $y_i^\gamma = y_j^\gamma = 1$.

Next, let us observe that

$$\mathbb{E}\left(\sum_{i \in \mathcal{V}_1 \cup \mathcal{V}_2} w_i y_i^\gamma\right)$$

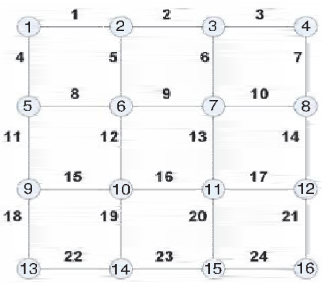

Fig. 1. A grid network with 24 links

$$= \mathbb{E}\left(\sum_{i\in\mathcal{V}_1} w_i \ \ (\tilde{y}_i > \gamma)\right) + \mathbb{E}\left(\sum_{i\in\mathcal{V}_2} w_i \ \ (\tilde{y}_i > 1-\gamma)\right)$$
$$= \sum_{i\in\mathcal{V}_1\cup\mathcal{V}_2} w_i\tilde{y}_i,$$

which is the optimal value of the LP relaxation and thus also the optimal value of the binary MWIS problem. This implies that, for almost all γ (i.e., all γ in a set of probability 1), $\sum_{i\in\mathcal{V}_1\cup\mathcal{V}_2} w_i y_i^\gamma$ equals the optimal value of the MWIS problem. This proves the claim. ■

A node coloring based algorithm was proposed in [16] to estimate a solution of MWIS problem from the solution of its LP relaxation. It was shown that, in the case of bipartite graphs, the estimated solution satisfies the complementary slackness conditions associated with LP (3) and its dual, and thus it is an optimal solution of the MWIS problem. In [24], we have shown that correctness of their algorithm for bipartite graphs can be established as a corollary to Theorem 4.3.

D. Throughput Optimality of the Proposed Algorithm

It is possible to show that our algorithm is throughput-optimal under a time-scale separation assumption. The arguments here are standard, and we refer the reader to the technical report [24] for proof.

V. Numerical Results

In this section, we evaluate the performance of the proposed algorithm via simulation. We compare it to pure Q-CSMA and also to another scheduling algorithm that combines LP solution and Q-CSMA but uses the gradient project approach suggested in [16] to solve the LP relaxation. We refer to the latter algorithm as GP-GD in this section. The reason for choosing GP-GD for comparison is that it has been shown to be superior to other previous algorithms (see [16]). In our simulation, pure Q-CSMA and the proposed algorithm are referred as GD and OGP-GD respectively.

A. A 24 Link Grid Network

We first consider a grid network with 16 nodes and 24 links as shown in Figure 1. We assume one-hop interference constraints whereby simultaneous transmissions at adjacent links (i.e., at any two links that have a common incident node) fail. Consider the following four maximal schedules.

$$M_1 = \{1,3,8,10,15,17,22,24\},$$
$$M_2 = \{4,5,6,7,18,19,20,21\},$$

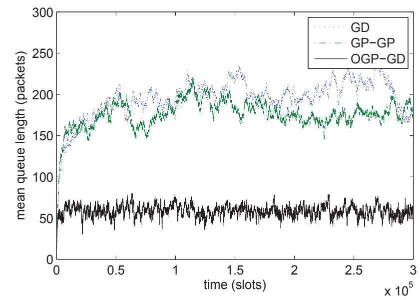

Fig. 2. Sample paths of average queue lengths for $\rho = 0.85$

$$M_3 = \{1,3,9,11,14,16,22,24\},$$
$$M_4 = \{2,4,7,12,13,18,21,23\}.$$

Let $x^1, x^2, x^3, x^4 \in \mathbb{R}_+^{24}$ be the vectors representing the schedules M_1, M_2, M_3 and M_4 respectively. We consider arrival rates that are convex combinations of these maximal schedules scaled by $0 < \rho < 1$, i.e., $\lambda(\rho) = \rho\sum_{j=1}^4 c_j x^j$, where $c = [0.2, 0.3, 0.2, 0.3]$. More specifically, packet arrivals at each link i follow an i.i.d. Bernoulli process with rate λ_i.

Note that, for all $0 < \rho < 1$, $\lambda(\rho)$ lies in the capacity region of the network, and as $\rho \to 1$, $\lambda(\rho)$ approaches a point on the boundary of the capacity region. Further, observe that we have a non-bipartite graph at our disposal, and hence we use the clique relaxation based algorithm for solving LP relaxation of MWIS. We have 16 maximal cliques in this example, each identified by a node of the network and consisting of all the links incident to this node. Finally, we use link weight functions $f_i(q_i) = \frac{\log(1+q_i)}{\log(\exp(1)+\log(1+q_i))}$ as suggested in [11].

Figure 2 shows the average queue-length evolution (averaged over all the links) for $\rho = 0.85$. We run each algorithm for 3×10^5 time slots. It is evident from the figure that the proposed algorithm leads to significantly smaller queue lengths (and hence mean delays) than the other two. To further illustrate, we perform 10 iterations of each algorithm for several values of $\rho \in (0, 1)$. In Figure 3, we plot the long term average per link queue lengths as functions of ρ. We observe that while Algorithms GD and GP-GD incur large delays for higher values of ρ, the proposed algorithm (OGP-GD) has excellent delay performance all through. We would like to point out that Algorithm GP-GD suffers from the slow convergence associated with standard gradient projection algorithm. Thus in using it we use link states computed with weights that may have changed significantly since the last reinitialization of the algorithm. This means that the time scale separation assumption of Section IV-D may not hold. Hence, the algorithm fails to bring much benefit over pure Q-CSMA.

B. A 16 Link Bipartite Conflict Graph

We now consider a network with 16 links that are seen as nodes in Figure 1. Any two links (i.e., nodes in Figure 1) that are connected, interfere with each other. We thus have a bipartite conflict graph now. Here we illustrate that we can use the edge relaxation based LP solution (Algorithm OGP$(\epsilon, t^0, \theta^i)$) and can still get significant performance improvement. To-

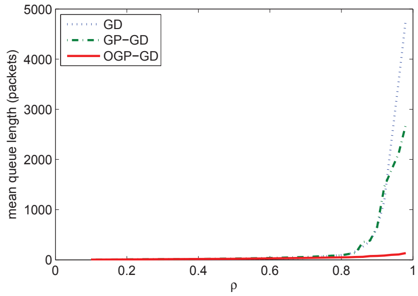

Fig. 3. Long term average per link queue lengths for the grid network as functions of ρ

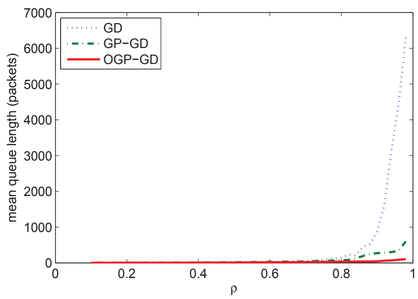

Fig. 4. Long term average per link queue lengths for the bipartite conflict graph as functions of ρ

wards this, we consider the two maximal schedules,

$$M_1 = \{1, 3, 6, 8, 9, 11, 14, 16\},$$
$$M_2 = \{2, 4, 5, 7, 10, 12, 13, 15\},$$

also represented by vectors $x^1, x^2 \in \mathbb{R}_+^{16}$ respectively. We consider arrival rates of the form $\lambda(\rho) = \rho \sum_{j=1}^{2} c_j x^j$, where $c = [0.5, 0.5]$ and $0 < \rho < 1$.

The long term average per link queue lengths are plotted in Figure 4 as functions of ρ. We observe performance benefits similar to those for the grid network of Section V-A.

VI. Conclusion

In this paper we have proposed LP relaxation based distributed algorithms for scheduling in time-slotted wireless networks. More precisely, we combine optimal gradient projection based LP solution algorithms with Glauber dynamics. Our algorithms rely on local message passing. The algorithms are throughput optimal under a time scale separation assumption. We also illustrate via simulation that combining LP solutions with Q-CSMA leads to a significantly better delay performance.

Acknowledgment

The research presented here was supported by ONR grant N00014-13-1-0038, NSF grant CNS-1161404, ARO MURIs W911NF-08-1-0233 and W911NF-07-1-0287, and DTRA grant HDTRA1-13-1-0030.

References

[1] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, December 1991.

[2] 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.

[3] C. Joo, X. Lin, and N. B. Shroff, “Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks,” in *IEEE INFOCOM*, Phoenix, AZ, USA, April 2008, pp. 1777–1785.

[4] X. Lin and S. B. Rasool, “Constant-time distributed scheduling policies for ad hoc wireless networks,” in *IEEE CDC*, San Diego, CA, USA, December 2006, pp. 1258–1263.

[5] S. Sanghavi, L. Bui, and R. Srikant, “Distributed link scheduling with constant overhead,” in *ACM SIGMETRICS*, San Diego, CA, USA, June 2007, pp. 313–324.

[6] S. Sarkar and S. Ray, “Arbitrary throughput versus complexity tradeoffs in wireless networks using graph partitioning,” *IEEE Transactions on Automatic Control*, vol. 53, no. 10, pp. 2307–2323, November 2008.

[7] R. R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin, “Throughput analysis in multihop CSMA packet radio networks,” *IEEE Transactions on Communications*, vol. 35, no. 3, pp. 267–274, March 1987.

[8] A. Proutiere, Y. Yi, and M. Chiang, “Throughput of random access without message passing,” in *CISS*, Princeton, NJ, USA, March 2008, pp. 509–514.

[9] L. Jiang and J. Walrand, “A distributed CSMA algorithm for throughput and utility maximization in wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 960–972, June 2010.

[10] S. Rajagopalan, D. Shah, and J. Shin, “Network adiabatic theorem: an efficient randomized protocol for contention resolution,” in *ACM SIGMETRICS/Performance*, Seattle, WA, USA, June 2009, pp. 133–144.

[11] J. Ghaderi and R. Srikant, “On the design of efficient CSMA algorithms for wireless networks,” in *IEEE CDC*, Atlanta, GA, USA, December 2010, pp. 954–959.

[12] 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,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 825–836, June 2012.

[13] J. Ghaderi and R. Srikant, “Effect of access probabilities on the delay performance of Q-CSMA algorithms,” in *IEEE INFOCOM*, Orlando, FL, USA, March 2012, pp. 2068–2076.

[14] C.-H. Lee, D. Y. Eun, S.-Y. Yun, and Y. Yi, “From glauber dynamics to metropolis algorithm: Smaller delay in optimal CSMA,” in *IEEE ISIT*, Cambridge, MA, USA, July 2012, pp. 2681–2685.

[15] S. Sanghavi, D. Shah, and A. S. Willsky, “Message passing for maximum weight independent set,” *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 4822–4834, November 2009.

[16] I. C. Paschalidis, F. Huang, and W. Lai, “Distributed scheduling of wireless networks: A message passing approach,” in *Mediterranean Conference on Control and Automation*, Platanias-Chania, Crete - Greece, June 2013, pp. 922–929.

[17] A. Schrijver, *Combinatorial Optimization*, ser. Algorithms and Combinatorics. Springer, 2003, vol. 24.

[18] A. Beck, A. Nedić, A. Ozdaglar, and M. Teboulle, “Optimal distributed gradient methods for network resource allocation problems,” Submitted for publication, 2013, http://web.mit.edu/asuman/www/documents/NUM-FGM.pdf.

[19] Y. Nesterov, “A method of solving a convex programming problem with convergence rate $O(1/k^2)$,” *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376, 1983.

[20] F. Huang, “On the maximum weighted independent set problem with applications in wireless sensor networks,” Ph.D. dissertation, Boston University, March 2013.

[21] W. R. Pulleyblank, “Minimum node covers and 2-bicritical graphs,” *Mathematical Programming*, vol. 17, no. 1, pp. 91–103, 1979.

[22] G. Grimmett, “An exact threshold theorem for random graphs and the node-packing problem,” *Journal of Combinatorial Theory, Series B*, vol. 40, no. 2, pp. 187–195, April 1986.

[23] D. R. Wood, “On the maximum number of cliques in a graph,” *Graphs and Combinatorics*, vol. 23, no. 3, pp. 337–352, June 2007.

[24] C. Singh, A. Nedić, and R. Srikant, “LP-relaxation based distributed algorithms for scheduling in wireless networks,” July 2013, http://www.ifp.illinois.edu/~srikant/distributed-scheduling-extended.pdf.

[25] A. Eryilmaz, R. Srikant, and J. R. Perkins, “Stable scheduling policies for fading wireless channels,” *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 411–424, April 2005.

[26] A. Beck and M. Teboulle, *Gradient-based algorithms with applications to signal-recovery problems*. Cambridge University Press, 2009, pp. 42–88.

Appendix
Proof of Theorem 4.1

Proof of Part 1: Our proof is along the lines of [26], but treats a more general scenario with scaled norms. We proceed through a sequence of Lemmas. The first one is a scaled version of the so called *descent lemma*.

Lemma A.1: For all $\theta, \eta \in \mathbb{R}_+^{|\mathcal{E}|}$,

$$V_\epsilon(\theta) \le V_\epsilon(\eta) + \langle \theta - \eta, \nabla V_\epsilon(\eta) \rangle + \frac{1}{2}\|\theta - \eta\|_H^2.$$

Proof: Recall that, for each $i \in \mathcal{V}$, $\nabla v_i(\theta^i)$ is Lipschitz continuous over $\mathbb{R}_+^{|\mathcal{N}_i|}$ with a Lipschitz constant $l_i = \frac{d_i}{\epsilon}$. Using the unscaled descent lemma for functions $v_i(\cdot), i \in \mathcal{V}$,

$$v_i(\theta^i) \le v_i(\eta^i) + \langle \theta^i - \eta^i, \nabla v_i(\eta^i) \rangle + \frac{d_i}{2\epsilon}\|\theta^i - \eta^i\|^2$$

for all $\theta^i, \eta^i \in \mathbb{R}_+^{|\mathcal{N}_i|}$. Summing these over all $i \in \mathcal{V}$, we get the desired inequality. ■

We define the mapping $\theta = p(\eta)$ to represent the updates of iterates $\theta(t)$ in Algorithm OGP(ϵ, t^0, θ^i) (see (11)), i.e.,

$$p(\eta) = \left[\eta - H^{-1}\nabla V_\epsilon(\eta)\right]_+ .$$

Note that $p(\eta)$ can be interpreted as the solution of the optimization problem

$$\underset{\xi \in \mathbb{R}_+^{|\mathcal{E}|}}{\text{minimize}} \quad \left\|\xi - \left(\eta - H^{-1}\nabla V_\epsilon(\eta)\right)\right\|_H^2 ,$$

and thus, it satisfies the following optimality condition:

$$\begin{aligned} &\left\langle \theta - p(\eta), p(\eta) - \eta + H^{-1}\nabla V_\epsilon(\eta)\right\rangle_H \ge 0, \\ \text{or } &\langle \theta - p(\eta), \nabla V_\epsilon(\eta)\rangle \ge \langle \theta - p(\eta), \eta - p(\eta)\rangle_H \end{aligned} \tag{13}$$

for any $\theta \in \mathbb{R}_+^{|\mathcal{E}|}, \eta \in \mathbb{R}^{|\mathcal{E}|}$.

Lemma A.2: For every $\theta \in \mathbb{R}_+^{|\mathcal{E}|}, \eta \in \mathbb{R}^{|\mathcal{E}|}$,

$$2\left(V_\epsilon(\theta) - V_\epsilon(p(\eta))\right) \ge \|\theta - p(\eta)\|_H^2 - \|\theta - \eta\|_H^2.$$

Proof: Note that

$$\begin{aligned} &V_\epsilon(\theta) - V_\epsilon(p(\eta)) \\ &\ge V_\epsilon(\theta) - V_\epsilon(\eta) - \langle p(\eta) - \eta, \nabla V_\epsilon(\eta)\rangle - \frac{1}{2}\|p(\eta) - \eta\|_H^2 \\ &\ge \langle \theta - p(\eta), \nabla V_\epsilon(\eta)\rangle - \frac{1}{2}\|p(\eta) - \eta\|_H^2, \\ &\ge \langle \theta - p(\eta), \eta - p(\eta)\rangle_H - \frac{1}{2}\langle \eta - p(\eta), \eta - p(\eta)\rangle_H, \end{aligned}$$

where the first inequality follows from Lemma A.1, the second from convexity of $V_\epsilon(\cdot)$, and the third from (13). Now, simplifying the right hand side,

$$\begin{aligned} V_\epsilon(\theta) - V_\epsilon(p(\eta)) &\ge \frac{1}{2}\langle 2\theta - p(\eta) - \eta, \eta - p(\eta)\rangle_H \\ &= \frac{1}{2}\langle \theta - p(\eta) + \theta - \eta, \theta - p(\eta) - \theta + \eta\rangle_H \\ &= \frac{1}{2}\left(\|\theta - p(\eta)\|_H^2 - \|\theta - \eta\|_H^2\right). \end{aligned}$$

This proves the claim. ■

The next lemma provides the key recursive relation for the sequence $\{V_\epsilon(\theta(t)) - V_\epsilon(\theta^\epsilon)\}$.

Lemma A.3: The sequence of iterates $\{\theta(t)\}$ of Algorithm OGP(ϵ, t^0, θ^i) satisfies, for $t > t^0$,

$$2\left(\beta^2(t)d(t) - \beta^2(t+1)d(t+1)\right) \ge \|e(t+1)\|_H^2 - \|e(t)\|_H^2,$$

where
$$\begin{aligned} d(t) &= V_\epsilon(\theta(t)) - V_\epsilon(\theta^\epsilon), \\ e(t) &= \beta(t)\theta(t) - (\beta(t) - 1)\theta(t-1) - \theta^\epsilon. \end{aligned}$$

Proof: Invoking Lemma A.2 with $\theta = \frac{1}{\beta(t+1)}\theta^\epsilon + (1 - \frac{1}{\beta(t+1)})\theta(t)$ and $\eta = \eta(t+1)$, we have

$$\begin{aligned} &2\left(V_\epsilon\left(\frac{\theta^\epsilon}{\beta(t+1)} + \left(1 - \frac{1}{\beta(t+1)}\right)\theta(t)\right) - V_\epsilon(\theta(t+1))\right) \\ &\ge \frac{1}{\beta^2(t+1)}\left(\|\beta(t+1)\theta(t+1) - (\beta(t+1)-1)\theta(t) - \theta^\epsilon\|_H^2\right. \\ &\quad \left. - \|\beta(t+1)\eta(t+1) - (\beta(t+1) - 1)\theta(t) - \theta^\epsilon\|_H^2\right). \end{aligned} \tag{14}$$

By convexity of $V_\epsilon(\cdot)$,

$$\begin{aligned} &\text{LHS of (14)} \\ &\le 2\left(\frac{V_\epsilon(\theta^\epsilon)}{\beta(t+1)} + \left(1 - \frac{1}{\beta(t+1)}\right)V_\epsilon(\theta(t)) - V_\epsilon(\theta(t+1))\right) \\ &= \frac{2}{\beta^2(t+1)}\left(\beta^2(t)d(t) - \beta^2(t+1)d(t+1)\right), \end{aligned} \tag{15}$$

where the equality follows since $\beta^2(t) = \beta^2(t+1) - \beta(t+1)$. On the other hand,

$$\text{RHS of (14)} = \frac{1}{\beta^2(t+1)}\left(\|e(t+1)\|_H^2 - \|e(t)\|_H^2\right), \tag{16}$$

where we have used the relation

$$\eta(t+1) = \theta(t) + \left(\frac{\beta(t) - 1}{\beta(t+1)}\right)(\theta(t) - \theta(t-1)).$$

Combining (15) and (16), we obtain the desired inequality. ■

We now prove the convergence rate result. Setting $t^0 = 0$, from Lemma A.3,

$$2\beta^2(t+1)d(t+1) + \|e(t+1)\|_H^2 \le 2\beta^2(t)d(t) + \|e(t)\|_H^2$$

for all $t \ge 1$. But, for $t = 1$,

$$\begin{aligned} 2\beta^2(1)d(1) + \|e(1)\|_H^2 &= 2\left(V_\epsilon(\theta(1)) - V_\epsilon(\theta^\epsilon)\right) + \|\theta(1) - \theta^\epsilon\|_H^2 \\ &\le \|\eta(1) - \theta^\epsilon\|_H^2 = \|\theta^0 - \theta^\epsilon\|_H^2, \end{aligned}$$

where, to get the inequality, we have once again invoked Lemma A.2 with $\theta = \theta^\epsilon$ and $\eta = \eta(1)$. Hence we conclude that, for all $t \ge 1$,

$$2\beta^2(t)d(t) + \|e(t)\|_H^2 \le \|\theta^0 - \theta^\epsilon\|_H^2,$$

which implies that

$$\begin{aligned} &2\beta^2(t)d(t) \le \|\theta^0 - \theta^\epsilon\|_H^2, \\ \text{or} \qquad &d(t) \le \frac{\|\theta^0 - \theta^\epsilon\|_H^2}{2\beta^2(t)}. \end{aligned}$$

The desired inequality is obtained after noting that $\beta^2(t) \ge \frac{t+1}{2}$ for all $t \ge 1$ (starting with $\beta(0) = 1$).

Proof of Part 2: This follows from [18, Theorem 3.1] after observing that $U_\epsilon(x)$ is a strongly concave function with parameter ϵ.