On the practical complexity of solving the maximum weighted independent set problem for optimal scheduling in wireless networks

Peng Wang
School of Electrical and
Computer Engineering
University of Delaware
Email: pwangee@udel.edu

Stephan Bohacek
School of Electrical and
Computer Engineering
University of Delaware
Email: bohacek@udel.edu

***Abstract*— It is well known that the maximum weighted independent set (MWIS) problem is NP-complete. Moreover, optimal scheduling in wireless networks requires solving a MWIS problem. Consequently, it is widely believed that optimal scheduling cannot be solved in practical networks. However, there are many cases where there is a significant difference between worst-case complexity and practical complexity. This paper examines the practical complexity of the MWIS problem through extensive computational experimentation. In all, over 10000 topologies are examined. It is found that the MWIS problem can be solved quickly, for example, for a 2048 node topology, it can be solved in approximately one second. Moreover, it appears that the average computational complexity grows polynomially with the number of nodes and linearly with the mean degree of the conflict graph.**

I. Introduction

It is well known that optimal throughput in wireless networks can be achieved by scheduling transmissions [1]. In 1984, [2] claimed that optimal scheduling is NP-complete. Since then, several authors have made similar claims. On the other hand, it has been shown that under the assumption that co-channel interference does not arise, the problem is polynomial (e.g., see [3], [4], [5]). Unfortunately, co-channel interference does arise in typical wireless networks. For this reason, it is believed that optimal scheduling, with co-channel interference, is NP-complete, and that optimal scheduling is computational impossible except for trivial networks. In contrast, we have found that in practical scenarios, optimal schedules can be quickly computed.

This paper examines the computational complexity of optimal scheduling in practical wireless networks. Specifically, optimal scheduling requires solving a graph theoretic problem known as the maximum weighted independent set (MWIS) problem. Thus, the complexity of optimal scheduling is tied to the complexity of MWIS problem. In general, the MWIS problem is NP-complete [6]. Moreover, it is NP-complete to approximate the MWIS with an approximation ratio of $n^{1-\varepsilon}$, for $\varepsilon > 0$, where n is the number of vertices in graph [7]. On the other hand, there are many classes of graphs where the MWIS problem has polynomial complexity. For example, MWIS can be found in polynomial time of perfect graphs[1][8], disk graphs [9], circle graphs [10], trees [11] and as well as many families of graphs that are free of particular subgraphs [12], [13]. The MWIS problem is also solvable in polynomial time on line graphs. In wireless scheduling, line graphs arise when there is no co-channel interference.

Beside restrictive the case where the network where there is no co-channel interference, it is unknown whether the MWIS problems that arise in practical wireless scheduling have any special properties that make them solvable in polynomial time. Nonetheless, through extensive computational experiments we have found that the MWIS that arises in practical wireless scheduling can be solved quickly. As shown in Section VI, the MWIS that arises when computing the optimal schedule for a wireless network with 2048 nodes can be computed in approximately one second. Moreover, computational evidence indicates that the computation time grows polynomially with the size of the network. This paper will also demonstrate that the number of nodes and the average degree of the conflict graph (defined in Section III-B) are good predictors of the computation time. Other factors such as node density and bit-rate impact the average degree of the conflict graph, and hence do not need to be considered beyond their impact on the degree of the conflict graph.

To investigate this behavior, we consider three approaches. First, Section II examines problems with worst-case exponential complexity that have been found to be easily solved in practice. Section IV reconsiders the proofs of NP-completeness of the wireless scheduling problem and indicates that the current proofs can only be applied to the special networks that do not arise in typical deployments. Finally, Section VI examines the complexity of the MWIS that arises in practical wireless networks. Three propagation environments are considered, namely, the two-ray model, the two-ray model with lognormal shadow fading, and ray-tracing on downtown Chicago. Under these propagation models, topologies of dif-

[1]A perfect graph is one that does not include and chordless cycle of odd length greater or equal to five.

ferent sizes and densities are generated. In all, approximately 10000 topologies are generated. Optimal schedules are computed on these topologies, and hence the MWIS problem is also solved for these topologies.

II. Worst-Case and Average Complexity

In our recent work, we have found that optimal scheduling is practical. Our findings are not in contradiction with earlier proofs of NP-completeness, but rather is well aligned with the practical computational complexity in a wide range of other NP-complete or exponential problems. For example, consider linear programming. In [14] it was shown that in the worst-case, the computational complexity of the simplex algorithm is exponential in the size of problem. On the other hand, there is an abundance of evidence that in practice, the computational complexity of the simplex algorithm is $m \times n$ where m is the number of constraints and n is the number of variables [15]. Moreover, interior point methods have been developed that have polynomial complexity. However, despite the fact that interior point methods have a better worst-case performance, state-of-the-art solvers such as CPLEX [16] and XPress [17] use the simplex method. In summary, there may be a substantial difference between worst-case computational complexity and practical computational complexity.

While there are many ways to quantify practical computational complexity, one common approach is to use the average complexity [18]. By this definition of complexity, several problems that are NP-complete in the worst case, are polynomial on average. For example, in graph theoretic problems, average complexity is the complexity averaged over solving the problem over random graphs. A random graph is one where an edge between any two vertices exists with probability p. Under this definition, it has been shown that the average complexity of finding Hamilton Cycles and solving the edge coloring problem are polynomial [19], [20]. In the case of the maximum independent set (MIS) problem, there exists an algorithm with average complexity of $\sum_{k=1}^{n}\binom{n}{k}2^{-k(k-1)/2} = O\left(n^{\log(n)}\right)$ on random graphs with $p = 1/2$, where there are n nodes in the graph [21]. It is important to point out that for networks with thousands of nodes, $\log(n)$ is relatively small. In contrast, there exist approximations of the MIS with complexity n^{100} [22], which, while polynomial, is more complex than $n^{\log(n)}$ for any practical wireless network.

The SAT problem, which is NP-complete in general, is another example of a significant difference between practical computational complexity and worst-case computation complexity. While we are not aware of any proofs on average complexity, several researchers have developed algorithms that can quickly solve randomly generated problems [23], [24], [25], [26]. Due to the relevance in artificial intelligence research, the average complexity of the SAT problem has been extensively studied. One finding is that the distribution of the problems plays an important role in the average complexity of SAT problem [27].

The impact of the distribution is troublesome since it indicates that the complexity averaged over a specific distribution of problems might be significantly different from the average complexity when averaged over problems that appear in practice. This paper investigates the practical complexity of the MWIS problems that arise in computing optimal schedules for wireless networks.

III. Optimal Scheduling and the MWIS Problem

Maximizing throughput or maximizing network utility often directly or indirectly includes solving a MWIS problem. This section briefly explains how particular approach to maximize throughput gives rise to a MWIS problem.

A. Throughput Maximization

Let ϕ denote a particular connection, with Φ denoting the set of all such connections. The data rate along with connection ϕ is denoted by f_{ϕ} and the path followed by connection ϕ is denoted by $P(\phi)$, that is, $P(\phi)$ is the set of links used by connection ϕ, and the total data rate sent over link x is $\sum_{\{(\phi)|x\in P(\phi)\}} f_{\phi}$, where $\{(\phi)\,|x \in P(\phi)\}$ is the set of flows that cross link x. All links are directional.

We define an *assignment* to be a vector $v = \left[\begin{array}{ccc} v_1 & \cdots & v_L \end{array}\right]$, where there are L links in the network and where $v_x \in \{0,1\}$ with $v_x = 1$ implying that link x is transmitting during assignment v. The *set of considered assignments* is denoted by $\mathcal{V}$, while the *set of all assignments* is denoted $\overline{\mathcal{V}}$. Since $v_x \in \{0,1\}$, $\overline{\mathcal{V}}$ contains 2^L assignments. The size of $\overline{\mathcal{V}}$ is the main challenging facing throughput maximization. Thus, typically, $\mathcal{V} \subsetneqq \overline{\mathcal{V}}$.

The data rate across link x during assignment v is denoted by $R(v,x)$. In general, $R(v,x)$ is a complicated function. However, here a simple binary relationship is used to define $R(v,x)$. Specifically,

$$R(v,x) = \begin{cases} R_x \text{ if } v_y = 0 \text{ for all } y \in \chi(x) \\ 0 \text{ otherwise,} \end{cases} \qquad (1)$$

where $\chi(x)$ is a set of links that conflict with x, i.e., $y \in \chi(x)$ if simultaneous transmissions over x and y are not possible. R_x is the nominal data rate over link x. Note that this definition of $R(v,x)$ neglects the possibility of transmission errors due to the aggregate interference from several links not in $\chi(x)$. However, as discussed in [28], such problems can easily be addressed. All computations in this paper use this technique, and hence the computed capacities account for multiple interferers.

The set of conflicting links, $\chi(x)$, depends on the communication model. Arguably, the $SINR$ binary communication model is the most relevant of the binary models[2] and is the model that is used in this paper. Let $SINR(x_{\text{Rec}}, y_{\text{Trans}})$ be the SINR at the receiver of link x when link y is also transmitting. Then, the SINR binary communication model specifies that $y \in \chi(x)$ if $SINR(x_{\text{Rec}}, y_{\text{Trans}}) < T(x)$ or $SINR(y_{\text{Rec}}, x_{\text{Trans}}) < T(y)$, where $T(x)$ and $T(y)$ are thresholds that depend on the modulation schemes. If link layer ACKs are used, then $y \in \chi(x)$ if $SINR(x,y) < T(x)$ or $SINR(y,x) < T(y)$

[2] A binary communication model is one that satisfies (1) for some χ.

Algorithm 1 Computing an Optimal Schedule

1: Select an initial set of assignments $V(0)$, set $k = 0$.
2: Solve (2) for $\mathcal{V} = \mathcal{V}(k)$ and compute $\mu(k)$ and $\lambda(k)$, the Lagrange multipliers associated with constraints (2c) and (2d), respectively.
3: Search for an assignment $v^{+} \notin \mathcal{V}(k)$ such that

$$\sum_{x=1}^{L} \mu_x(k) R(v^{+}, x) > \lambda(k). \tag{2f}$$

4: **if** such an assignment is found **then**
  set $\mathcal{V}(k+1) = V(k) \cup v^{+}$, set $k = k+1$, and go to Step 2.
5: **else**
  if no such assignment *exists*, then stop, the optimal schedule has been found.
6: **end if**

In this paper, it is assumed that all channel gains are constant. Since the focus of this paper is on the communication over the mesh infrastructure (which is not moving), such an assumption is reasonable.

A schedule is a convex combination of assignments. Specifically, a schedule is a set $\{\alpha_v : v \in \mathcal{V}\}$ where $\sum_{v\in\mathcal{V}} \alpha_v \leq 1$ and $\alpha_v \geq 0$. With this notation, the total data rate that the schedule $\boldsymbol{\alpha}$ provides over link x is $\sum_{v\in\mathcal{V}} \alpha_v R_x v_x$. Finally, the capacity optimization problem is

$$\max_{\boldsymbol{\alpha},\mathbf{f},F} F \tag{2a}$$

subject to:

$$F \leq f_\phi \text{ for all } \phi \in \Phi \tag{2b}$$

$$\sum_{\{\phi | x\in P(\phi)\}} f_\phi \leq \sum_{v\in\mathcal{V}} \alpha_v R(v,x) \text{ for each link } x \tag{2c}$$

$$\sum_{v\in\mathcal{V}} \alpha_v \leq 1 \tag{2d}$$

$$0 \leq \alpha_v \text{ for each } v \in \mathcal{V}, \tag{2e}$$

where $\mathbf{f}$ is the vector of flow rates.

As mentioned, the challenge in solving this problem is that optimality can be achieved if $\mathcal{V} = \overline{\mathcal{V}}$, but $\overline{\mathcal{V}}$ has 2^L elements. Alternatively, the set $\mathcal{V}$ can be constructed iteratively following Algorithm 1. It can be proved that Algorithm 1 will converge to the optimal solution. However, Step 3 in Algorithm 1 requires searching for a new assignment. As explained in the next section, this step is the same as solving the MWIS problem.

B. MWIS

The search for a new assignment required in Step 3 of Algorithm 1 can be accomplished by solving

$$\max_{v} \sum_{x=1}^{L} R(v,x)\,\mu_x, \tag{3}$$

where μ_x is the Lagrange multiplier associated with constraint (2c). As will be shown next, solving this maximization is equivalent to finding the maximum weighted independent set of the weighted conflict graph.

The utility of the conflict graph for finding schedules has been demonstrated in several previous works (e.g., [2], [29]). A wireless network induces a conflict graph as follows. Each link in the network induces a vertex in the conflict graph. Thus, a link x in the network is associated with a vertex in the conflict graph; this vertex is denoted with x, where whether x refers to a link in the network or a vertex in the conflict graph is clear from the context. There is an edge between vertices x and y if $y \in \chi(x)$, where, as discussed in Section III-A, x and links in $\chi(x)$ cannot simultaneously transmit. The weighted conflict graph is constructed by assigning the weight $R_x\mu_x$ to vertex x, where R_x is the nominal data rate across link x.

An independent set (or stable set) of a graph is a set of vertices where no two vertices in the set are neighbors. Thus, an independent set of the conflict graph is a set of links that are not in conflict and hence, are able to transmit simultaneously. Letting I be an independent set, the weight of I is the sum of the weights of the vertices in I, i.e., $\sum_{x\in I} R_x\mu_x$. Since I is an independent set, $\sum_{x\in I} R_x\mu_x = \sum_{x=1}^{L} R(v(I), x)\,\mu_x$. Thus, solving (3) is equivalent to finding the maximum weighted independent set.

While there are several techniques available to compute MWIS, this paper used a technique based on integer linear programming presented in [30].

IV. On Proofs of the NP-Completeness of Optimal Scheduling

The proof that optimal scheduling is NP-complete is due to the fact that optimal scheduling can be posed as a MWIS problem on the conflict graph. In the seminal work on NP-completeness, Karp [6] showed that the maximum independent set problem (MIS) is the same complexity as the 3-SAT problem, which is NP-complete [31]. A wide range of papers have showed that the optimal scheduling in wireless networking requires solving a MWIS problem on the conflict graph, and concluded that optimal scheduling is NP-complete. In this section we will discuss the difficulty with using Karp's proof to prove the complexity of optimal scheduling for practical wireless networks. We begin with reviewing Karp's mapping of 3-SAT problems to MIS problems.

The 3-SAT problem is the problem of determining whether there exists a set of boolean variables $x_1, x_2,...,x_K$ such that

$$\begin{aligned}
&\left(s_{1,1}x_{I(1,1)} \text{ or } s_{1,2}x_{I(1,2)} \text{ or } s_{1,3}x_{I(1,3)}\right) \\
&\text{and} \left(s_{2,1}x_{I(2,1)} \text{ or } s_{2,2}x_{I(2,2)} \text{ or } s_{2,3}x_{I(2,3)}\right) \\
&\cdots \text{and } \left(s_{m,1}x_{I(m,1)} \text{ or } s_{m,2}x_{(m,2)} \text{ or } s_{m,3}x_{I(m,3)}\right) \\
&\cdots \text{and } \left(s_{M,1}x_{I(M,1)} \text{ or } s_{M,2}x_{I(M,2)} \text{ or } s_{M,3}x_{I(M,3)}\right) \\
= \;& \textit{true}
\end{aligned} \tag{4}$$

where $I(m,i) \in \{1, 2, .., K\}$ and $I(m,i) = j$ implies that the ith variable in the mth clause is the jth boolean variable x_j. We require that $I(m,i) \neq I(m,j)$ for $i \neq j$. Furthermore, $s_{m,i} \in \{+,-\}$, with $-x_i$ being the negation of x_i. For

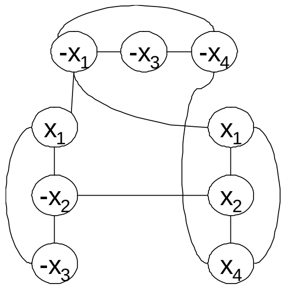

Fig. 1. Graph representation of the 3SAT problem (5).

example,

$$(x_1 \text{or} - x_2 \text{or} - x_3) \text{ and } (x_1 \text{or} x_2 \text{or} x_4) \text{ and } (-x_1 \text{or} - x_3 \text{or} - x_4) \quad (5)$$

is satisfiable; one solution is $(x_1, x_2, x_3, x_4) = (true, true, false, true)$. Note that in this example there are four boolean variables and three clauses, i.e., $K = 4$ and $M = 3$.

The 3-SAT problem can be made into an independent set problem as follows. A graph with vertex set V and edge set E is constructed with

$$\begin{aligned} V &= \{(I(m,i), m) \,|\, i = 1,2,3 \text{ and } m = 1, ..., M\} \quad (6)\\ E &= \{((I(m,i), m), (I(n,j), n)) | \, I(m,i) = I(n,j) \\ &\text{ and } s(m,i) \neq s(n,j)\} \\ &\cup \{((I(m,i), m), (I(m,j), m)) | \;\, i,j = 1,2,3 \text{ and } j \neq i\} \end{aligned}$$

Thus, each of the three variables in a clause is represented as a vertex (so the total number of vertices is $3 \times M$). Vertices from the same clause are all neighbors. Vertices in different clauses are neighbors only if they represent the same boolean variable, but with opposite sign. For example, if x_1 is in one clause, it is a neighbor of $-x_1$ in any other clause. Figure 1 shows the graph representation of the 3-SAT problem (5). Following Karp's proof, it is possible to show that finding an independent set of size M for the graph (6) is equivalent to solving the 3-SAT problem (4). Note that while [32], [33] used the 3-SAT problem to show that a maximum matching problem is NP-complete, the proof is similar to Karp's proof, which shows that the MIS problem is NP-complete. Hence, the discussion below applies to maximum matching problems as well.

The relationship between the 3-SAT problem and optimal scheduling for wireless networks hinges on the graph (6) being one that arises in practical mesh networks. However, the graph (6) has the following properties that greatly limit its applicability to wireless networks

Proposition 1: Each vertex in (6) is in exactly one 3-clique and the largest clique has three vertices.

Proof: A vertex has an edge to vertices that are from the same clause and to vertices from other clauses that represent the same boolean variable of the opposite sign. Since there are three vertices derived from every clause, each vertex has two neighbors, and these three vertices form a 3-clique. We claim that there are no other 3-cliques. To show this, suppose that vertices A, B, and C form a clique and that B is from a different clause. In this case, B must be from the same boolean variable as A, but with a opposite sign. If C is from the same clause as A, then C is from a distinct boolean variable from A, and hence cannot be a neighbor of B. If C is from a different clause than A, then C is from the same boolean as A, but with a different sign from A, and hence the same sign as B. Thus, B and C cannot be neighbors. ■

Proposition 2: The average degree of each vertex is no greater than $\frac{1}{2}\frac{M}{K} + 2$.

Proof: Each vertex has two neighbors from the same clause and is a neighbor of the vertices in different clauses that represent the same boolean variables but with the opposite sign. The average number of clauses in which a boolean variable appears is M/K. Suppose that a of these M/K boolean variables appear with a positive sign and $b = M/K - a$ of these variables appear with a negative sign. Then the average degree is $2ab/(a+b) = 2a(M/K - a)/(M/K)$, which has a maximum of $\frac{1}{2}M/K$. Adding this to the two neighbors from the same clause yields the desired result. ■

Proposition 1 greatly restricts the conflict graphs that can be constructed from 3-SAT problems. This, in turn, restricts the wireless networks that have conflict graphs that can be constructed from 3-SAT problems. The following provides some examples of the restrictions on the wireless networks that have conflict graphs that can be constructed from 3-SAT problems. It is important to note that Propositions 1 and 2 focus on the conflict graph (with vertices and edges), while the proposition below focuses on wireless networks (with nodes and links).

Proposition 3: Suppose that a wireless network has a conflict graph of the form (6), and hence can be represented as a 3-SAT problem. Then the wireless network must have the following properties.

1) Each node communicates with no more than three other nodes. If a node does communicate with three other nodes, then these nodes do not communicate with each other.
2) If ACKs are used, then
   a) the diameter of the wireless network is no greater than three, and
   b) there are no more than 4 nodes in each communicating component of the network[3].
   c) If the 2-hop conflict model is used, then there are no more than 4 nodes in each isolate component of the network[4].
3) For a given link, A, there must be two other links, B and C, that both interfere with A and interfere with each other. While there may be other links D_i with $i = 1, 2, ...$ that interfere with A, for each i, these links, D_i, must not interfere with B, C, or D_j with $i \neq j$.

[3]A connected component of the network is a set of nodes that can communicated, perhaps through multiple hops.

[4]Two links are isolated if they do not interfer with each other. An isolated component is a set of links that do not interfer with links outside of the set.

Proof:

1) Let node a communicate with nodes b, c, and d. Then, it is not possible to simultaneously communicate across any two of the three links, $\overrightarrow{ab}$, $\overrightarrow{ac}$, and $\overrightarrow{ad}$. Thus, these three links form a 3-cycle in the conflict graph. By Proposition 1, this is the largest possible clique, and hence node a cannot communicate with any more nodes. Now suppose that nodes b and c communicate. In this case, it is not possible to simultaneously communicate across any two of the links $\overrightarrow{ab}$, $\overrightarrow{ac}$, and $\overrightarrow{bc}$. These three links form another 3-clique in the conflict graph. However, link $\overrightarrow{ab}$ is already a member of the 3-clique $\overrightarrow{ab}$, $\overrightarrow{ac}$, and $\overrightarrow{ad}$, and a link cannot be a member of more than one 3-cycle.
2) a) Suppose that there is a four hop path, with nodes a, b, c, d, e, where node a communicates with node b, node b communicates with nodes c and a, and so on. Since ACKs are used, simultaneous communication across links $\overrightarrow{ab}$ and $\overrightarrow{cd}$ is not possible. Thus, the links $\overrightarrow{ab}$, $\overrightarrow{bc}$, and $\overrightarrow{cd}$ form a 3-clique in the conflict graph. Similarly, links $\overrightarrow{bc}$, $\overrightarrow{cd}$, and $\overrightarrow{de}$ form a 3-clique in the conflict graph. Thus, link $\overrightarrow{bc}$ is in two 3-cliques. However, by Proposition 1 a link can only be a member of one 3-clique in the conflict graph.
   b) The only allowable topologies with four nodes are the ones shown in Figure (2). To see this, consider the case where a node communicates with three nodes (i.e., the left-hand side of Figure 2), then the three links shown interfere with each other and hence form a 3-clique. If any of the nodes shown communicate with some other node, then the link to this other node would interfere with the three links shown, and hence, the conflict graph would have a 4-clique, which is forbidden. If no node communicates with three nodes, but some do communicate with two nodes, then by 2.a., the topology on the left in Figure 2 is the largest possible.
   c) In the case of the 2-hop conflict model, if node cannot communicate directly or via a single hop, then the nodes are isolated in the sense that they do not interfere. Hence, the result follows from 2.b.
3) From Proposition 1, a link A must be in exactly one 3-clique in the conflict graph. Thus, each link must interfere with at least two other links and these links must interfere with each other, forming a 3-clique. If the links D_i interfere with A and they interfere with each other, or with B or C , then link A would be a member of multiple 3-cliques, which is forbidden by Proposition 1.

■

Thus, we conclude that wireless networks that have conflict graphs corresponding to 3-SAT problems are special. For example, under some conditions, Proposition 3 shows that only small topologies correspond to 3-SAT problems, in which case, the 3-SAT problem is quite easy. Thus, the relevant question is whether difficult 3-SAT problems correspond to optimal scheduling problems in practical wireless networks. As mentioned in Section II, the 3-SAT problem has been extensively studied and, on average, it has been found to be "easily" solved. More specifically, it has been found that the well-known DP method [34] can easily solve randomly generated 3-SAT problems (see [23] for problem generation) when M/K takes values less than 3 and greater than 5, with 4.5 being the peak of difficulty [27]. From Proposition 2, this implies that if the average degree of the conflict graph is greater than 4.5 and the conflict graph corresponds to a 3-SAT problem, then the 3-SAT problem is easy.

In conclusion, the relationship between the computational complexity of difficult 3-SAT problems and the computational complexity of optimal scheduling for practical wireless networks is not as clear as indicated by previous research. Specifically, a proof of hardness of practical wireless scheduling must show that the scenario in Figure 3 does not arise, but rather that there is overlap between difficult 3-SAT problems and practical scheduling problems. To the best of our knowledge, the existence of such an overlap has not be proved.

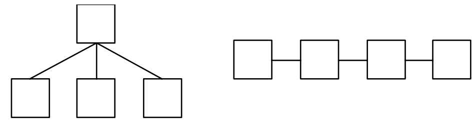

Fig. 2. The topologies shown above are the largest the wireless networks that uses ACKs and that has conflict graphs that can be correspond to a 3-SAT problem.

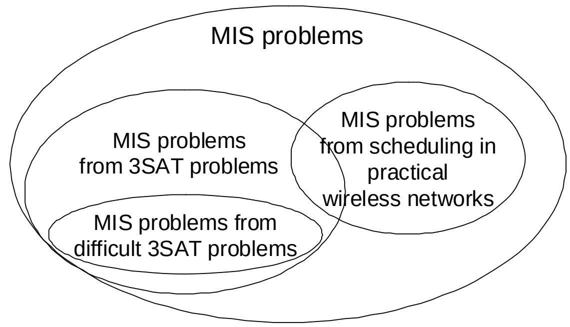

Fig. 3. Possible Venn diagram of maximum independent set (MIS) problems, 3-SAT problems, and scheduling problems.

V. Construction of Random Wireless Networks

In order to investigate the average computational complexity of the MWIS problem for optimal scheduling in practical wireless networks, statistics must be generated from a large number of networks. This investigation focuses on topologies that might arise in large scale wireless mesh networks. Such infrastructure networks are composed of wireless routers and gateways, which have both wired and wireless interfaces. Such networks have densely distributed wireless routers while gateways are more lightly distributed. The routing forms a forest, where gateways are roots of the trees. In this investigation, five parameters are used to characterize a mesh network. Four of these parameters are the number of the nodes, the density of nodes (i.e., how many neighbors a node has), the density of the gateways, and the target bit-rate of links. The propagation environment also plays an important role in the performance of a network. Thus, in order to fully explore the computational complexity, we consider three popular propagation environments, namely, the two-ray model, the two-ray with shadow fading model, and a realistic urban propagation model. Thus, the propagation model is a fifth parameter that controls the topology. The next subsections detail the generation of random topology based on these five parameters.

1) Propagation Models: Propagation is a key aspect of wireless networks. In the two-ray propagation model, the received signal strength (in dB) at a node that is d meters from a transmitting node is

$$P_{\text{2Ray}}(d) = 20\log_{10}\left(\frac{\lambda}{4\pi}\right) + P_{\text{TransmitPower}}\,[\text{dB}] - \begin{cases} 20\log_{10}(d) & \text{for } d \leq C \\ 40\log_{10}(d/C) + 20\log_{10}(C) & \text{for } d > C \end{cases},$$

where C is a parameter that depends on the node height. In the case of hand-held radios, the height is approximately 1.5m, and $C = 225m$. Throughout this paper, $P_{\text{TransmitPower}} = 18$ dB and $\lambda = 0.125$m, as is the case for 802.11b/g. When shadow fading is added, the received signal strength (in dB) at a node that is d meters from a transmitting node is

$$P_{\text{2RayAndShadowing}}(d) = P_{\text{2Ray}}(d) + X$$

where X is a Gaussian random variable with mean 0 and standard deviation 4 dB [35]. We assume that nodes are spaced far enough apart that the random part of the propagation are independent. However, propagation is symmetric [35].

Due to the difference between indoor and outdoor propagation, and due to wave guide effects of streets, urban propagation is distinct from the random propagation models. Thus, in order to investigate the performance the complexity of optimal scheduling in urban areas, the UDel Models Propagation Simulator [36] was employed. Specifically, for this study, ray-tracing was performed on a 2 km^2 region of downtown Chicago. This computation provided the received signal strength between any pair of nodes.

Topologies were randomly generated by selecting a subset of nodes from a large *baseline set of nodes*. In the case of the two-ray propagation model and the two-ray with shadowing model, the baseline set of nodes were densely distributed so that within a 15 km^2 region 5000 nodes where distributed. However, for the urban propagation model, nodes were placed to mimic a large infrastructure network. Specifically, outdoors, nodes were placed on lampposts throughout the city, and indoors, enough nodes were placed on each floor so that the entire floor was covered. In all, the baseline set of nodes included over 7000 nodes positioned throughout the city.

Algorithm 2 Selecting the Gateways

1: Let $\mathcal{G}$ be a randomly selected set of NGW nodes from $\mathcal{N}$.
2: Set $\mathcal{G}' = \mathcal{G}$.
3: Remove the node from $\mathcal{G}$ that has been in $\mathcal{G}$ for the most iterations.
4: Set $\mathcal{G} = \mathcal{G} \cup \arg\min_{w \in \mathcal{N} \setminus \mathcal{G}} D(w)$.
5: If $\mathcal{G}' \neq \mathcal{G}$, go to 2, else stop.

2) Random Topology Generation:

a) Node Selection: Beyond the propagation model, four parameters are used to construct a topology, namely, n the number of nodes, r^* the target bit-rates, Δ the maximum number of neighbors, and NGW the number of gateways. The target bit-rate corresponds to a specific received signal strength. Letting $RSS(r)$ be the minimum required received signal strength to decode a transmission at data rate r, then using 802.11g's coding and modulation, typical values of RSS are

$$RSS(6) = -90\text{dBm};\ RSS(12) = -87\text{dBm};$$
$$RSS(18) = -84\text{dBm}; RSS(24) = -81\text{dBm};$$
$$RSS(36) = -78\text{dBm}; RSS(48) = -74\text{dBm};$$
$$RSS(54) = -72\text{dBm},$$

where the data rates are in Mbps. We say that two nodes are neighbors if the propagation model results in a received signal strength that is above $RSS(r^*)$.

Let N denote the set of nodes in the topology. Initially, $\mathcal{N}$ is a single node selected at random. Then, a node is selected at random among all the nodes that satisfy $1.)$ the node has between 1 and Δ neighbors in $\mathcal{N}$, and $2.)$ adding the node to $\mathcal{N}$ will not make any node in $\mathcal{N}$ have more than Δ neighbors in $\mathcal{N}$. If no such node exists, then the process is restarted. If suitable nodes do exist, the process continues until $\mathcal{N}$ has n elements.

Next, gateways are selected. The objective is that the gateways are uniformly spread throughout the network in the sense that the average distance from a node to the closest gateway is minimized. This is formulized as follows. Given a set of gateways, $\mathcal{G}$, a new gateway is added by finding the node, w, that minimizes $D(u)$ where

$$D(u) = \sum_{w \in \mathcal{N}} \min\left(d(u,w), \min_{g \in \mathcal{G}} d(g,w)\right)$$

where $d(u,w)$ is the distance in hops from node u to node w. Thus, the gateways are selected in Algorithm 2.
Note that the above is not a convex optimization and hence the final set of gateways might depend on the initial selection of gateways. Thus, the above algorithm was run ten times and the set of gateways that resulted in the smallest value of $\sum_{w \in \mathcal{N}} \min_{g \in \mathcal{G}} d(g,w)$ was used.

b) Routing: Once the wireless routers and gateways have been selected, the routing was determined. As mentioned above, the routing forms a forest, where each gateway is a root

Algorithm 3 Greedy Method to Select Single Path

1: Let S_x be the optimal flow rates that solve (7) and set $\mathcal{W} = \varnothing$.
2: Randomly select $w \in \mathcal{N}$ and $w \notin \mathcal{W}$.
3: Set $\mathcal{W} = w \cup \mathcal{W}$.
4: $\boldsymbol{P}(w) = \arg\max_{p \in \mathcal{P}} \min_{x \in p} S_x$, i.e., $\boldsymbol{P}(w)$ is the path that results in the highest flow to w.
5: $\boldsymbol{S}(w) = \min\left(F, \min_{x \in P(w)} S_x\right)$.
6: Set $S_x = S_x - \boldsymbol{S}(w)$ for each $x \in \boldsymbol{P}(w)$.
7: If $\mathcal{W} \neq \mathcal{N}$ go to 2, else stop.

of a tree and each wireless router is in exactly one tree. While there are several approaches to routing, this investigation used a max-flow-based, interference aware routing.

The first step in forming routes is to identify the set of potential links, their bit-rates, and the links that they interfere with. Let x denote a link with transmitter x_t and receiver x_r and let P_x be the received signal strength at the receiver. The bit-rate used by link x is denoted $r(x)$ and is given by

$$r(x) := \max\left\{r : P_x - PGuard > RSS(r)\right\},$$

where $PGuard$ is used as a buffer to reduce sensitivity to interference. This study used $PGuard = 3$ dB. If no such bit-rate exists (i.e., $P_x - PGuard < -90$dBm), then the link is removed from consideration. Given the bit-rates, for each link x, the set of conflicting links, $\chi(x)$ can be found, as described in Section III-A.

Interference aware, multi-path max-flow routing is found by solving

$$\max_{\mathbf{S},F} F$$

$$\sum_{\{x:x_t=w\}} S_x - \sum_{\{y:x_r=w\}} S_y + F = 0 \text{ for } w \notin GW \quad (7a)$$

$$\frac{S_x}{r(x)} + \sum_{y \in \chi(x)} \frac{S_y}{r(y)} \leq 1 \text{ for all } x, \quad (7b)$$

where S_x is the flow over link x. Note this optimization problem approximates the impact of interference. Specifically, $\frac{S_x}{r(x)}$ is the fraction of time that link x transmits, and hence (7b) ensures that the fraction of time that link x transmits and the fraction of times that all links that interfere with link x transmit sum to no more than one. Of course, it is possible that some links that interfere with x can transmit simultaneously. But (7b) does not account for this possibility. Thus, (7) provides a lower bound on the throughput. It should be pointed out that while problem (7) is polynomial, solving (7) was, by far, the computational bottleneck of this investigation.

Problem (7) results in multipath routing. Single path routing can be formed by quantization as follows. Define $\mathcal{P}(w)$ to be the set of paths from some gateway to node w. Then the greedy algorithm shown in Algorithm 3 is used to construct $\boldsymbol{P}(w)$, a path from some gateway to node w.

VI. Results from Computational Experiments

A. Computation time as a function of the number of nodes in the network

Figure 4 (a), (b), and (c) show the average time to solve (3) for urban propagation, the two-ray propagation model, and the two-ray with shadowing propagation model, respectively. These computation times[5] were averaged over each iteration of Algorithm 1 and averaged over 40 randomly generated topologies. When generating these topologies, the maximum number of neighbors, Δ, was equal to 6, the target bit-rate was set to 24Mbps, and the number of gateways was the number of nodes in the network divided by 16.

Three conclusions can be made from Figure 4. First, the time to solve the MWIS is quite small, with 2048 node topologies taking approximately one second. Clearly, the statement that the MWIS can only be solved for trivial networks is incorrect. Second, it appears that in practice, the time to compute the MWIS grows polynomially with the size of the network. Specifically, for the topologies shown in Figure 4, we have

$$\text{Computation time} \approx A \times n^B \text{ sec.}$$

where (A, B) is $\left(10^{-6.7}, 1.97\right)$, $\left(10^{-6.7}, 1.85\right)$, $\left(10^{-6.1}, 1.75\right)$ for urban propagation, the two-ray model, and the two-ray with shadowing model, respectively. This relationship between computation time and topology size is also shown in Figure 4. A third conclusion drawn from Figure 4 is that the computational complexity does not greatly depend on the propagation model. Specifically, the computation times for different propagation models are within 10%.

Note that the y-axis in Figure 4 shows the computation time minus 0.0095 msec. We estimate that this was the time to load the CPLEX optimizer (which is a DLL) and to begin solving the MWIS problem. Also, since the MWIS can be solved so quickly, computation times are difficult to estimate accurately for small topologies. Thus, for topologies with less than 256 nodes, every time Algorithm 1 required the MWIS problem to be solved, it was solved ten times. For these cases, the computation time was taken as one tenth of the elapsed CPU time between when the first time the MWIS about to be solved and just after the tenth time the MWIS was solved. Nonetheless, as the 95 percentile confidence intervals indicate, the estimated computation time for small topologies shows a high variance.

B. Impact the Mean Degree of the Conflict Graph

The topology generation method described in Section V makes use of several parameters, namely, the propagation model, the number of nodes, the number of gateways, the target bit-rate, r^*, and the maximum number of neighbors, Δ. Figure 4 shows the impact of the number of nodes

[5]All computations were run on a machine with two Intel E5440 CPUs and 16GB RAM using Matlab v 7.2, and CPLEX v 10 with the Tomlab interface to CPLEX. However, at all times, 8 computations were solved simultaneously. Hence, the computation times shown correspond to the time on a single core (i.e., the MWIS problem was not parallelized across cores).

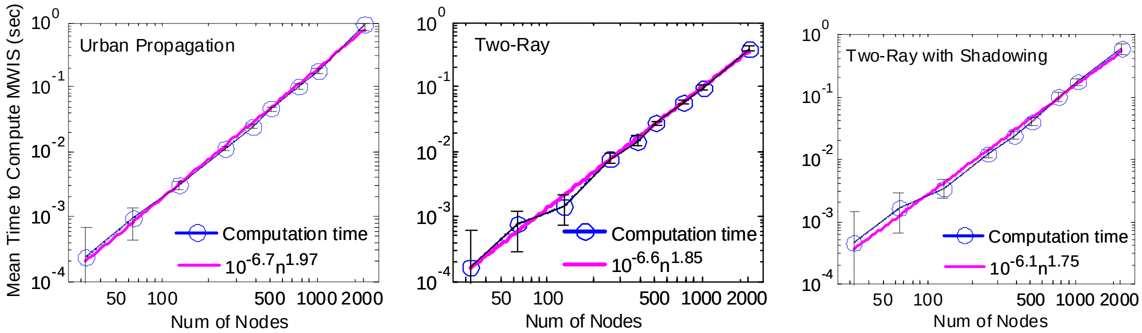

Fig. 4. The time to compute the MWIS versus the number of nodes in the network. (a), (b), and (c) show this relationship when the propagation is the urban propagation, the two-ray model, and the two-ray with shadow fading model, respectively. In all cases, the maximum number of neighbors is 24, the target bit-rate is 24 Mbps, and the number of gateways is the number of nodes divided by 16.

and the propagation environment. We have found that the other topology parameters are best understood as impacting the mean degree of the conflict graph, which impacts the computation time. Figure 5 shows the relationship between the mean degree of the conflict graph and the computation time for a wide range of topology parameters, but with the target bit-rate fixed at 24 Mbps. Here, urban propagation is used and each point is averaged over 40 topologies (both the mean degree of the conflict graph and mean computation time are averaged over 40 topologies). As can be observed, there is an approximately linear relationship between the computation time and the mean degree of the conflict graph. Specifically, the computation time is approximately

$$\text{computation time} \approx C \times \text{mean degree of conflict graph} \quad (8)$$

where C is 0.00018, 0.0012, and 0.008 for 128, 512, and 1024 node networks, respectively. These lines are also shown in Figure 5.

Figure 5 also shows the computation time for particular values of the topology parameters. As expected, as Δ, the maximum number of neighbors in the wireless network, increases, the mean degree of the conflict graph increases. Figure 5 (b) shows the computation time for different numbers of gateways. As can be observed, as the number of gateways increases, the mean degree of the conflict graph slightly decreases, leading to a slight reduction in the computation time. Moreover, notice that the computation times for 64 gateways tends to be slightly below the linear fit, while the 16 gateway cases tend to be slightly above the linear fit. Nonetheless, the linear relationship between the mean degree of the conflict graph and the computation time provides a reasonable approximation for a wide range of gateways.

The topologies shown in Figure 5 used a target bit-rate of 24Mbps. Figure 6 shows the relationship between the mean degree of the conflict graph and the time to compute the MWIS for a wide range of target bit-rates. In this case, urban propagation was used, there were 512 nodes in the topology, and the maximum number of neighbors was set to 6. This figure shows that the mean degree (and computation time) increases with the target bit-rate. The reason for this behavior

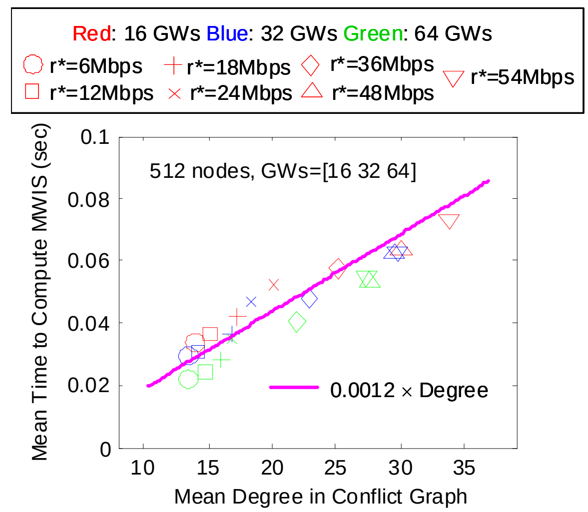

Fig. 6. The mean time to compute the MWIS as a function of the mean degree of the conflict graph for different topologies where the topologies are generated with different target bit-rates.

is that higher bit-rates are more susceptible to interference. For example, 54 Mbps requires 23 dB of SINR, while 6 Mbps only requires 5 dB of SINR. Consequently, transmissions that are several hops away will interfere with high bit-rate transmission, whereas only nearby transmissions will impact low bit-rate transmissions. Since the mean number of neighbors is held fixed, links in topologies where a high target bit-rate is used will interfere with considerably more links than do links in topologies with low target bit-rates.

The linear fit shown in Figure 6 is the same one shown in Figure 5, i.e., (8) with $C = 0.0012$. This further confirms the linear relationship between the mean degree of the conflict graph and the time to compute the MWIS.

Figure 7 shows the relationship between the slope of the lines in Figure 5 and the number of nodes in the network. Figure 7 also shows the curve $Slope = 3.8 \times 10^{-8} n^{1.76}$. As can be observed, this model provides a high quality of fit. Thus, we conclude that with the computers and algorithms used in

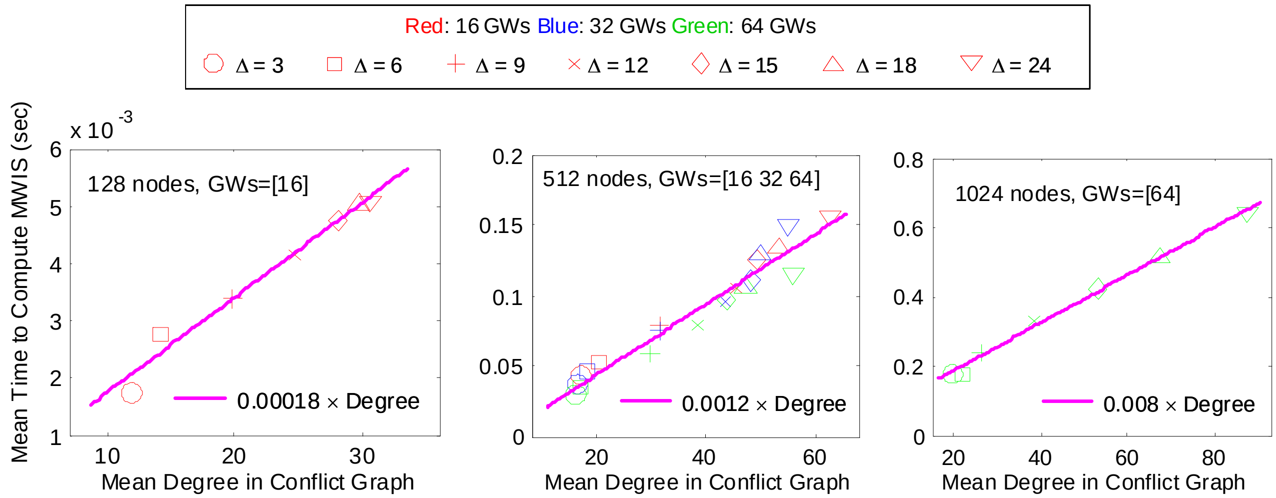

Fig. 5. The mean time to compute the MWIS versus the mean degree of the conflict graph for several topologies. (a) Shows the case where the topologies have 128 nodes, 16 gateways, and the maximum number of neighbors, Δ vaires from 3 to 24. (b) Shows the case where the topologies have 512 nodes, 16, 32, and 64 gateways, and Δ vaires from 3 to 24. (c) Shows the case where the topologies have 1024 nodes, 64 gateways, and Δ vaires from 3 to 24.

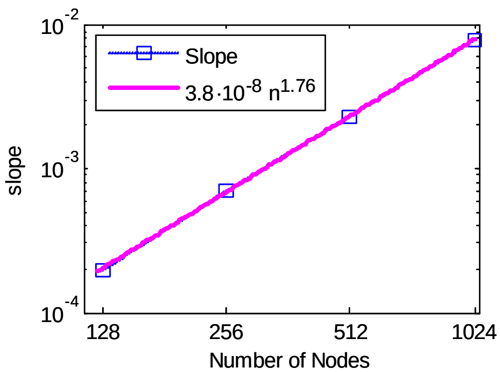

Fig. 7. The slope of the lines shown in Figure 5 versus the number of nodes.

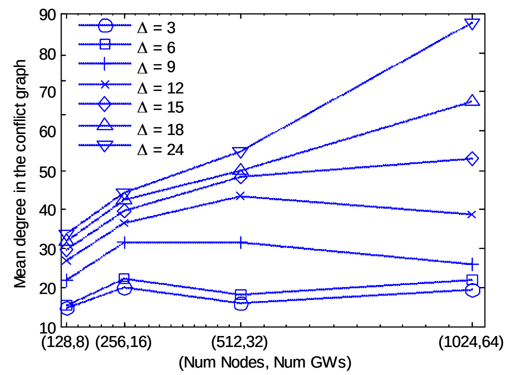

Fig. 8. Relationships between the number of nodes and the mean degree of the conflict graphs.

this investigation, the time to solve MWIS problem associated with optimal scheduling in practical wireless networks can be modeled as

$$\begin{aligned} & \text{Time to solve the MWIS} \\ \approx\; & 3.8 \times 10^{-8} \times n^{1.76} \times \text{mean degree of the conflict graph.} \end{aligned} \tag{9}$$

It is important to note that (9) does not imply that the time to solve the MWIS grows like $n^{1.76}$. Specifically, as Figures 6 and 5 show, the mean degree also varies with the number of nodes. Thus, the scaling of the computation time depends on how the network is scaled, or more specifically, how this scaling impacts the mean degree of the conflict graph. The time to compute the MWIS grows like $n^{1.76}$ only if the mean degree of the conflict graph is somehow held constant as the size of the network grows. However, the mean degree of the conflict graph varies in a complicated way, and hence there does not appear to be any simple relationships between the number of nodes and the mean degree of the conflict graph. Figure 8 shows how the mean degree of the conflict graph varies as the number of nodes increases, but the gateway density is held constant and Δ, the maximum number of neighbors, is held constant. For this type of scaling of the topology, it is difficult to draw any definitive conclusions about the relation between the number of nodes and the mean degree of the conflict graph. For example, for $\Delta = 24$ and $\Delta = 18$, the mean degree clearly increases with the number of nodes. However, for other values of Δ, the mean degree of the conflict graph appears to reach a plateau. Referring back to Figure 4, we see that in the topologies used there, the computation time grows like $n^{1.98}$, $n^{1.85}$, and $n^{1.75}$ for the urban propagation, the two-ray model, and the two-ray model with shadow fading, respectively. Thus, in order for (9) to hold, the mean degree of the conflict graph must grows like $n^{0.22}$, $n^{0.09}$, and $n^{-.01}$, for urban propagation, the two-ray model, and the two-ray with shadowing model, respectively. Considering Figure 8, such a variation is plausible. Nonetheless, further research into the mean degree of the conflict graph is required.

VII. Conclusions

This paper studied the practical computational complexity of the maximum weighted independent set (MWIS) problem that arises in optimal scheduling in wireless networks. In contrast to folklore, the MWIS is solvable in practical wireless scheduling problems. By examining over 10000 randomly generated topologies, it was found that the time to compute the MWIS grows polynomially with the number of nodes and linearly with the mean degree of the conflict graph. More specifically, the mean time to solve the MWIS problem for networks with 2048 nodes was approximately one second. While this result might appear to be in conflict with prior research on the complexity of scheduling, it is not. First, there are a wide range of problems that have a worst-case complexity that grows exponentially with the size of the problem, and yet in practice grow polynomially with the size of the problem. Second, prior research on the complexity of scheduling relied on the relationship between the 3-SAT problem and the MWIS problem. However, the MWIS problem that arises in scheduling is particular subset of MWIS problems, and hence the relationship between scheduling and the 3-SAT problem is not clear.

An important consequence of this paper is that the ability to quickly solve MWIS problems allows optimal schedules to be quickly found. In previous work, the perceived practical intractability had been circumvented by using suboptimal methods or by making strong assumptions about interference.

References

[1] L. Tassiulas and A. Ephremides, "Jointly optimal routing and scheduling in packet radio networks," *IEEE Trans. on Info. Theory*, vol. 38, no. 1, pp. 165–168, Jan 1992.

[2] E. Arikan, "Some complexity results about packet radio networks," *IEEE Trans. on Info. Theory*, vol. 30, no. 4, pp. 681–685, Jul 1984.

[3] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu, "Joint congestion control and distributed schedulingin multihop wireless networks with a node-exclusive interference model," in *Infocom*, 2006.

[4] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 302–315, April 2006.

[5] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Infocom*, 2006.

[6] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. Plenum, 1972, pp. 85–103.

[7] J. Hastad, "Clique is hard to approximate within $n^{1-\epsilon}$." in *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, 1996, pp. 627–636.

[8] M. Grotschel, L. Lovasz, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. Berlin: Springer-Verlag, 1993.

[9] T. Matsui, "Approximation algorithms for maximum independent set problems and fractional coloring problems on unit disk graphs," in *JCDCG*, 1998, pp. 194–200.

[10] G. Valiente, *A New Simple Algorithm for the Maximum-Weight Independent Set Problem on Circle Graphs*. Springer, 2003, vol. 2906, pp. 129–137.

[11] G. H. Chen, M. T. Kuo, and J. P. Sheu, "An optimal time algorithm for finding a maximum weight independent set in a tree," *BIT*, vol. 23, pp. 353–356, 1988.

[12] G. Minty, "On maximal independent sets of vertices in claw-free graphs," *J. Combinatorial Theory*, vol. B, no. 28, pp. 284–304, 1980.

[13] V. E. Alekseev and V. V. Lozin, "Augmenting graphs for independent sets," Rutgers Center of Operations Research, Tech. Rep. 59-2000, 2000.

[14] V. Klee and G. J. Minty, "How good is the simplex algorithm," in *Inequalities III*, O. Shisha, Ed. New York: Academic Press, 1972, pp. 159–175.

[15] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2000.

[16] ILOG, "CPLEX," 2008.

[17] Dash Optimization, "Xpress optimizer," 2007.

[18] V. Damerow, "Average and smoothed complexity of geometric structures," Ph.D. dissertation, Universitat at Paderborn, 2005.

[19] B. Bollobas, T. I. Fenner, and A. M. Frieze, "An algorithm for finding hamilton cycles in a random graph," in *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*. ACM, 1985, pp. 430–439.

[20] L. Perkovic, "Edge coloring, polyhedra and probability," Ph.D. dissertation, Carnegie Mellon University, 1998.

[21] H. S. Wilf, *Algortihms and Complexity*. AK Peters, Ltd, 2002, no. available at: http://www.physics.it/lectures/AlgorithmComplexity.pdf.

[22] M. M. Halldorsson and J. Radhakrishnan, "Improved approximations of independent sets in bound-degree graphs," in *SWAT'94*, 1994.

[23] A. Goldberg, "On the complexity of the satisfiability problem," New York University, Tech. Rep. Courant Computer Science Report 16, 1979.

[24] A. Goldberg, P. W. Purdom, and C. A. Brown, "Average time analysis of simplified davis-putnam procedures," *Information Processing Letters*, vol. 15, pp. 72–75, 1982.

[25] J. N. Hooker, "Resolution vs. cutting plane solution of interference problems: Some computational experience," *Operations Research Letters*, vol. 7, pp. 1–7, 1988.

[26] A. P. Kamath, N. K. Karmarker, K. G. Ramakrishnan, and M. G. C. Resende, "Computational experience with an interior point algorithm on the satisfiability problem," in *Proceedings of the Conference on Integer Programming and Combinatorial Optimization*. Mathematical Programming Society, 1990, pp. 333–349.

[27] D. G. Mitchell, B. Selman, and H. J. Levesque, "Hard and easy distributions for SAT problems," in *Proceedings of the Tenth National Conference on Artificial Intelligence*, P. Rosenbloom and P. Szolovits, Eds. Menlo Park, California: AAAI Press, 1992, pp. 459–465. [Online]. Available: citeseer.ist.psu.edu/mitchell92hard.html

[28] P. Wang and S. Bohacek, "Capacity optimization of mesh networks," *Submitted. Available at http://udelmodels.eecis.udel.edu*, 2007.

[29] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proceedings of ACM MobiCom*, San Diego, CA, September 2003, pp. 66–80.

[30] P. Wang and S. Bohacek, "Computationally practical optimal scheduling and routing for mesh networks," in *Submitted - Available at http://udelmodels.eecis.udel.edu*, 2007.

[31] S. Cook, "The complexity of theorem proving procedures," in *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing*, 1971, pp. 151–158.

[32] G. Sharma, R. Mazumdar, and N. Shroff, "Maximum weighted matching with interference constraints," in *IEEE FAWN*, 2006.

[33] ——, "On the complexity of scheduling in wireless networks," in *ACM Mobicom*, 2006.

[34] M. Davis and H. Putnam, "A computing procedure for quantification theory," *Journal of the ACM*, vol. 7, pp. 201–215, 1960.

[35] T. Rappaport, *Wireless Communications - Principles and Practice*. Prentice Hall, 2002.

[36] S. Bohacek, V. Sridhara, and J. Kim, "UDel Models," available at: http://udelmodels.eecis.udel.edu/.