

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

Peng Wang Stephan Bohacek
Department of Electrical and Computer Engineering
University of Delaware

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.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—Wireless Communication

Keywords

Maximum Weighted Independent Set; Optimal Scheduling; Wireless Mesh Network

1. 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.

*This work was prepared through collaborative participation in the Collaborative Technology Alliance for Communications and Networks sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAD19-01-2-0011. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WICON'08, November 17-19, 2008, Maui, Hawaii, USA.
Copyright 2008 ICST 978-963-9799-36-3.

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 $n^{1-\epsilon}$, for $\epsilon > 0$, 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 graphs¹ [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 the restrictive case where there is no co-channel interference and the case of networks restricted to one-dimensions (e.g., roads), 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 5, 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 3.2) 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.

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

The remainder of the paper proceeds as follows. The next section provides a brief overview of problems with worst-case exponential complexity that have been found to be easily solved in practice. Section 3 gives a brief overview of a way to compute optimal schedules, and, more specifically, how the MWIS problem arises in computing optimal schedules. Then, Section 4 details how the topologies used in this study were constructed. Sections 5 - 7 present the results of computational experiments involving over 10000 topologies. Finally Section 8 provides concluding remarks.

2. WORST-CASE AND AVERAGE COMPLEXITY

The results of the computational experiments presented below indicate that optimal scheduling is practical. In particular, the MWIS problem that arises in wireless scheduling can be quickly solved in practice. These results are not in contradiction with earlier proofs of NP-completeness, but rather are 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 $\sum_{k=1}^n \binom{n}{k} 2^{-k(k-1)/2} = O(n^{\log(n)})$ on random graphs with $p = 1/2$, where there are n nodes in the graph [21].

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 the authors are not aware of any proofs on average complexity, several researchers have developed algorithms that can quickly solve randomly generated problems [22, 23, 24, 25]. Due to the importance of the SAT problem, 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 [26].

The impact of the distribution is troublesome since it indicates that the complexity averaged over a specific distrib-

ution of problems might be significantly different from the average complexity when averaged over problems that appear in practice. For this reason, this paper investigates the practical complexity of the MWIS problems that arise in computing optimal schedules for wireless networks.

3. 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.

3.1 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 $\mathbf{v} = [v_1 \ \cdots \ v_L]$, 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 \mathbf{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, $\mathcal{V} \subsetneq \overline{\mathcal{V}}$.

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

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

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 $R(\mathbf{v}, x)$ neglects the possibility of transmission errors due to the aggregate interference from several links not in $\chi(x)$. However, as discussed in [27], such problems can easily be addressed. All computations in this paper use this technique, and hence the computed throughputs 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 models² 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 the transmitter of 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 receivers of data frames will transmit ACKs. Thus, in this case, $y \in \chi(x)$ if $SINR(x_{\text{Rec}}, y_{\text{Trans}}) < T(x)$, $SINR(x_{\text{Rec}}, y_{\text{Rec}}) < T(x)$, $SINR(y_{\text{Rec}}, x_{\text{Trans}}) < T(y)$, or $SINR(y_{\text{Rec}}, x_{\text{Rec}}) < T(y)$. See [28] for more details.

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

Algorithm 1 Computing an Optimal Schedule

- 1: Select an initial set of assignments $\mathcal{V}(0)$, select a level of accuracy $\rho \geq 0$, and set $n = 0$.
- 2: Solve (2) with $\mathcal{V} = \mathcal{V}(n)$, and, hence, compute the flow rates $\mathbf{f}(n)$ and the Lagrange multipliers $\boldsymbol{\mu}(n)$ and $\lambda(n)$ associated with constraints (2c) and (2d), respectively.
- 3: Search for an assignment $\mathbf{v}(n)$ such that

$$\mathbf{v}(n) = \arg \max_{\mathbf{v} \in \overline{\mathcal{V}}} \sum_{x=1}^L R(\mathbf{v}, x) \mu_x(n) - \lambda(n). \quad (3)$$

- 4: case $G(\mathbf{f}) = \min_{\phi \in \Phi} f_\phi$
 - 5: if $\frac{\sum_{x=1}^L R(\mathbf{v}(n), x) \mu_x(n) - \lambda(n)}{G(\mathbf{f}(n))} < \rho$
 - 6: Stop.
 - 7: end if
 - 8: case $G(\mathbf{f}) = \sum_{\phi \in \Phi} \log(f_\phi)$
 - 9: if $\sum_{x=1}^L R(\mathbf{v}(n), x) \mu_x(n) - \lambda(n) < L \log(1 + \rho)$
 - 10: Stop.
 - 11: end if
 - 12: set $\mathcal{V}(n+1) = \mathcal{V}(n) \cup \mathbf{v}(n)$
 - 13: set $n = n + 1$, and go to Step 2.
-

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 $\{\alpha_{\mathbf{v}} : \mathbf{v} \in \mathcal{V}\}$ where $\sum_{\mathbf{v} \in \mathcal{V}} \alpha_{\mathbf{v}} \leq 1$ and $\alpha_{\mathbf{v}} \geq 0$. With this notation, the total data rate that the schedule $\boldsymbol{\alpha}$ provides over link x is $\sum_{\mathbf{v} \in \mathcal{V}} \alpha_{\mathbf{v}} R_x v_x$. Finally, the capacity optimization problem is

$$\max_{F, \boldsymbol{\alpha}, \mathbf{f}} F \quad (2a)$$

subject to:

$$F \leq f_\phi \text{ for all } \phi \in \Phi \quad (2b)$$

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

$$\sum_{\mathbf{v} \in \mathcal{V}} \alpha_{\mathbf{v}} \leq 1 \quad (2d)$$

$$0 \leq \alpha_{\mathbf{v}} \text{ for each } \mathbf{v} \in \mathcal{V}, \quad (2e)$$

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. The following was proved in [27].

Theorem 1. Let $\mathbf{f}(n)$ be the vector of flow rates found by the n th iteration of Algorithm 1, let $\mathbf{f}(\infty)$ be the optimal flow rates, and let $G(\mathbf{f}) = \min_{\phi \in \Phi} f(\phi)$. Then $(G(\mathbf{f}(\infty)) - G(\mathbf{f}(n+1))) / (G(\mathbf{f}(\infty)) - G(\mathbf{f}(n))) \geq \delta$ for some constant $\delta > 0$, that is, $G(\mathbf{f}(n))$ converges to $G(\mathbf{f}(\infty))$ geometrically fast. Also, if Algorithm 1 terminates after n^* iterations with $\rho > 0$, then $\frac{G(\mathbf{f}(\infty)) - G(\mathbf{f}(n^*))}{G(\mathbf{f}(\infty))} < \rho$.

3.2 MWIS

The search for a new assignment required in Step 3 of Algorithm 1 requires solving

$$\max_{\mathbf{v}} \sum_{x=1}^L R(\mathbf{v}, x) \mu_x, \quad (4)$$

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 3.1, 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, $\sum_{x \in I} R_x \mu_x = \sum_{x=1}^L R(\mathbf{v}(I), x) \mu_x$. Thus, solving (4) is equivalent to finding the maximum weighted independent set.

While there are several techniques available to compute MWIS, this paper used a technique based ILP via

$$\max_{\mathbf{v}} \sum_{x=1}^L R_x \mu_x v_x \quad (5a)$$

$$\text{subject to: } v_x + v_y \leq 1 \text{ if } y \in \chi(x) \quad (5b)$$

$$v_x \in \{0, 1\}. \quad (5c)$$

In large networks, there are many constraints (5b). The computation time can be dramatically improved if a clique³ decomposition is used, where we define a clique decomposition to be a set of cliques $\{Q_i, i = 1, 2, \dots, M\}$ such that if $y \in \chi(x)$, then there is a clique Q_i such that $x \in Q_i$ and $y \in Q_i$. Then, (5) becomes

$$\max_{\mathbf{v}} \sum_{x=1}^L R_x \mu_x v_x \quad (6)$$

$$\text{subject to: } \sum_{x \in Q_i} v_x \leq 1 \text{ for } i = 1, 2, \dots, M$$

$$v_x \in \{0, 1\}.$$

This paper explores the computational complexity of solving (6), which is equivalent to solving a MWIS problem.

³A clique is a set of vertices where there is an edge between each vertex pair in the set.

4. 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.

4.1 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

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

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, $P_{\text{TransmitPower}} = 18$ dBm and $\lambda = 0.125m$, 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

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

where X is a Gaussian random variable with mean 0 and standard deviation 4 dB [30]. We assume that nodes are spaced far enough apart that the random part of the propagation are independent. However, propagation is symmetric [30].

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 of the complexity of optimal scheduling in urban areas, the UDel Models Propagation Simulator [31] was employed. Specifically, for this study, ray-tracing was performed on a 2 km² 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

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.
-

so that within a 15 km² region 5000 nodes were 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.

4.2 Random Topology Generation

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

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

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 \mathcal{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.

Algorithm 3 Greedy Method to Select Single Path

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

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 of a tree and each wireless router is in exactly one tree. While there are several approaches for 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

$$r(x) := \max \{r : P_x - P_{Guard} > RSS(r)\},$$

where P_{Guard} is used as a buffer to reduce sensitivity to interference. This study used $P_{Guard} = 3$ dB. If no such bit-rate exists (i.e., $P_x - P_{Guard} < -90$ 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 3.1.

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

$$\max_{\mathbf{S}, F} F \tag{7a}$$

$$\sum_{\{x: x_t=w\}} S_x - \sum_{\{y: x_r=w\}} S_y + F = 0 \text{ for } w \notin GW \tag{7b}$$

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

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 (7c) 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 (7c) 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 $\mathbf{P}(w)$, a path from some gateway to node w .

5. COMPUTATION TIME AS A FUNCTION OF THE NUMBER OF NODES IN THE NETWORK - THE LOW DEGREE CASE

Figure 1 (a), (b), and (c) show the average time to solve (4) for urban propagation, the two-ray propagation model, and the two-ray with shadowing propagation model, respectively. These computation times⁴ were averaged over each iteration of Algorithm 1 and averaged over 100 randomly generated topologies. For small topologies, the computation time is quite small. Randomness due to memory management and other high priority tasks, small computation times are significantly influenced by noise. Therefore, for topologies with fewer than 256 nodes, each time that the ILP program was to be solved, it was repeatedly solved ten times. The computation time was then the average of these ten times. Figure 1 also shows the 95% confidence intervals, which were found with bootstrapping. 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 1. 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 1, we have

$$\begin{aligned} \text{Time to find a MWIS for } \Delta = 6 & \tag{8} \\ \approx A \times n^B + T_o \text{ sec.} \end{aligned}$$

where (A, B, T_o) is $(10^{-6.7}, 1.97, 0.0095)$, $(10^{-6.7}, 1.85, 0.0095)$, $(10^{-6.1}, 1.75, 0.0095)$ 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 1. However, as will be shown in the following sections, this behavior is unique to networks that have a low degree (i.e., Δ is small). A third conclusion drawn from Figure 1 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 in (8), $T_o = 0.0095$ sec., for all types of propagation. We suspect that it takes approximately 9.5 msec to load the CPLEX optimizer (which is a DLL) and to begin solving the MWIS problem.

6. IMPACT THE MEAN DEGREE OF THE CONFLICT GRAPH

Figure 1 shows a relationship between the computation time and the size of the network under a specific type of topologies. We seek to understand this relationship for a wider range to types of topologies. This section will present

⁴All 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).

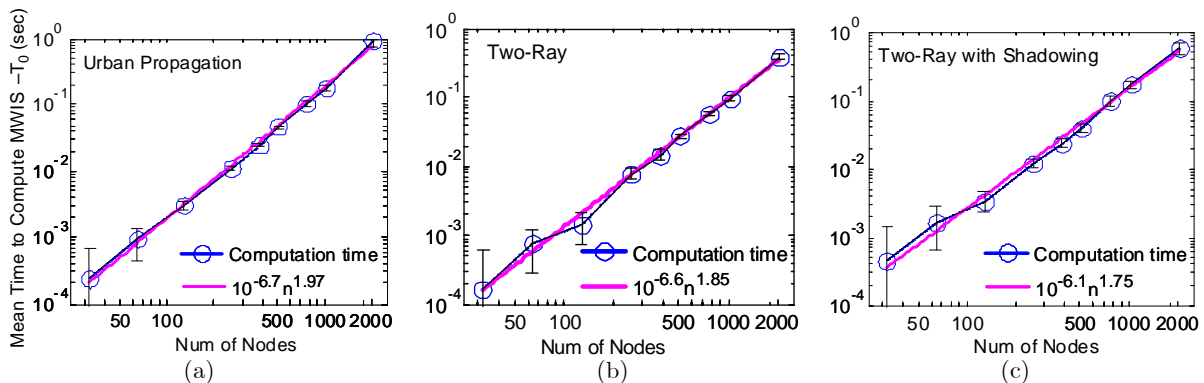


Figure 1: The time to compute the MWIS versus the number of nodes in the network. (a), (b), and (c) show this relationship when the propagation is the urban propagation, the two-ray model, and the two-ray with shadow fading model, respectively. In all cases, the maximum number of neighbors is 24, the target bit-rate is 24 Mbps, and the number of gateways is the number of nodes divided by 16.

results from computational experiments that indicate that

$$\frac{\text{Time to find a MWIS} - T_o}{\text{Mean degree of the conflict graph}} \approx \alpha \times n^\beta, \quad (9)$$

where α and β only depend on the propagation environment. This result implies that in terms of the time to find a MWIS, the mean degree of the conflict graph encapsulates many of the parameters used for generating topologies, namely, the number of gateways, the target bit-rate, r^* , and the maximum number of neighbors, Δ .

We take two steps to show that (9) is a good model for the computation time. First, we fix n , in which case (9) implies that

$$\begin{aligned} \text{Time to find a MWIS} - T_o \\ \approx K \times \text{Mean degree of the conflict graph}, \end{aligned} \quad (10)$$

where the constant K is dependent of the number of nodes, but K is independent of the other parameters of the topology, namely, the number of gateways, the target bit-rate, r^* , and the maximum number of neighbors, Δ . The next two subsections will explore the relationship between K and these other topology parameters.

In the second step to demonstrating (9), we will show that K is polynomial in n , that is, $K = \alpha \times n^\beta$ for some α and β , where α and β depend on the propagation model. This relationship is explored in Section 6.3.

6.1 The Mean Degree of the Conflict Graph, the Number of Gateways, and the Number of Neighbors

Figure 2 shows the relationship between the mean degree of the conflict graph and the computation time for different numbers of gateways and different numbers of maximum neighbors, Δ , but with the target bit-rate fixed at 24 Mbps. Here, urban propagation is used and, as above, each point is averaged over 100 topologies (both the mean degree of the conflict graph and mean computation time are averaged over 100 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 given by (10),

where K is 0.00018, 0.0012, and 0.008 for 128, 512, and 1024 node networks, respectively. These models are also shown in Figure 2.

Figure 2 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 2 (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 a larger number of gateways tends to be slightly below the linear fit, while a smaller number of gateways 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.

6.2 The Mean Degree of the Conflict Graph, the Target Bit-Rate, and the Number of Neighbors

The topologies shown in Figure 2 used a target bit-rate of 24Mbps. Figure 3 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 and different numbers of gateways. 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 that the mean degree of the conflict graph increases with the target bit-rate 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 maximum 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.

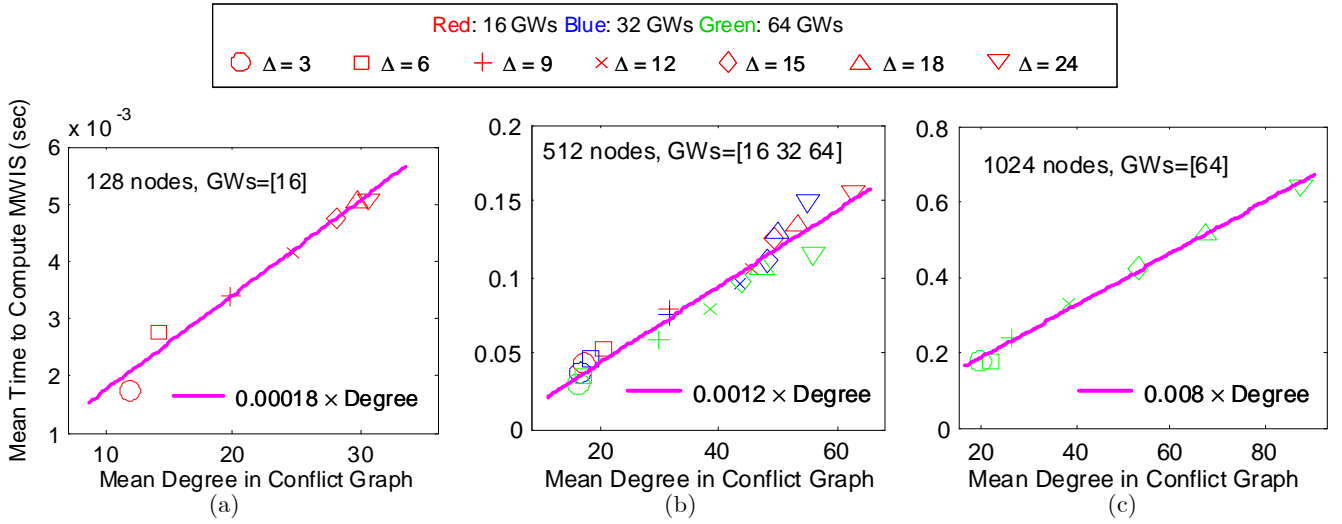


Figure 2: The mean time to compute the MWIS versus the mean degree of the conflict graph for several topologies. (a) Shows the case where the topologies have 128 nodes, 16 gateways, and the maximum number of neighbors, Δ varies from 3 to 24. (b) Shows the case where the topologies have 512 nodes, 16, 32, and 64 gateways, and Δ varies from 3 to 24. (c) Shows the case where the topologies have 1024 nodes, 64 gateways, and Δ varies from 3 to 24.

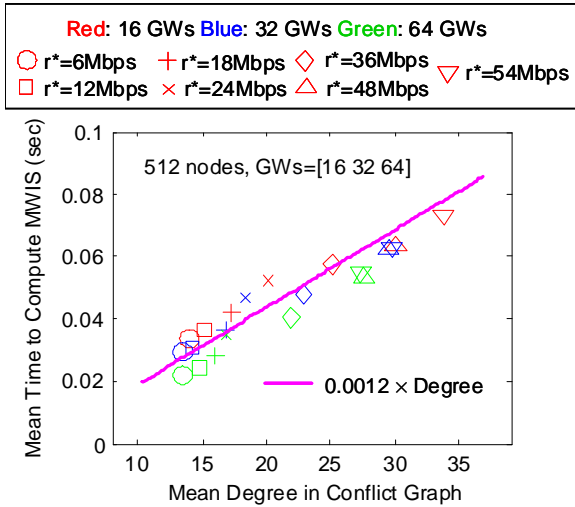


Figure 3: The mean time to compute the MWIS as a function of the mean degree of the conflict graph for different topologies where the topologies are generated with different target bit-rates.

The linear fit shown in Figure 3 is the same one shown in Figure 2, i.e., (10) with $K = 0.0012$. This further confirms the linear relationship between the mean degree of the conflict graph and the time to compute the MWIS.

6.3 Time to Compute a MWIS and the Mean Degree of the Conflict Graph

The previous sections provide a strong indication that (10) is a good model for the computation time, where K only depends on the number of nodes. While Figures 2 and 3 only show the case for urban propagation. However, the plots are similar for the two-ray propagation model and the two-ray with lognormal shadowing propagation model.

Figure 4 (a), (b), and (c) show K as a function of the number of nodes for urban propagation model, the two-ray propagation model, and the two-ray with lognormal shadowing propagation model, respectively. These plots also show the model $K = \alpha n^\beta$, where $(\alpha, \beta) = (1.77 \times 10^{-8}, 1.88)$, $(1.09 \times 10^{-7}, 1.64)$, $(7.78 \times 10^{-8}, 1.75)$, for the three types of propagation, respectively. As can be observed, this model provides a high quality of fit. Thus, we conclude that with the computers and algorithms used in this investigation, the time to solve MWIS problem associated with optimal scheduling in practical wireless networks can be modeled with (9).

7. THE MEAN DEGREE OF THE CONFLICT GRAPH

It is important to note that (9) does not imply that the time to find a MWIS grows like n^β . Specifically, as Figures 2 and 3 show, the mean degree of the conflict graph also varies with the number of nodes. Thus, the scaling of the computation time depends on how the network is scaled, or more specifically, it depends on how this scaling impacts the mean degree of the conflict graph. In the case of urban propagation, the time to compute the MWIS grows like

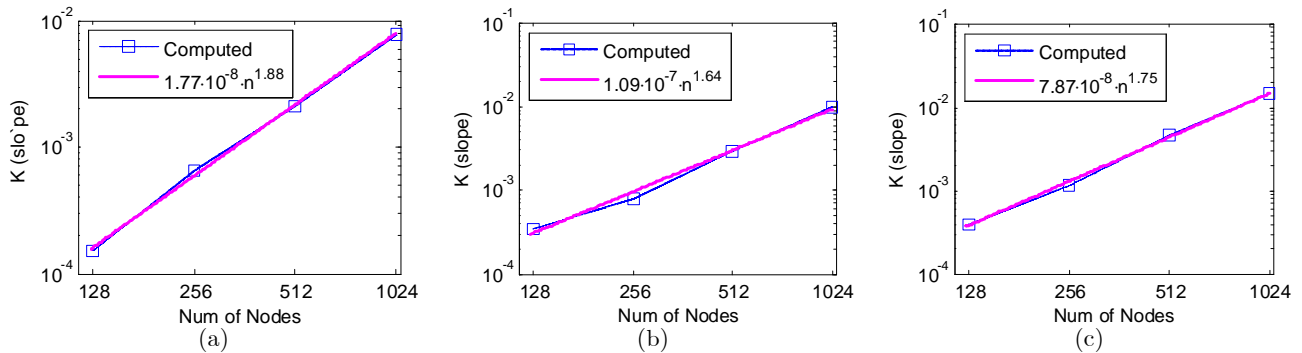


Figure 4: The relationship between K , the parameter in (10), and the number of nodes in the network. (a) is for urban propagation (b) is for the two-ray propagation model, and (3) is for the two-ray propagation model with lognormal shadow fading.

n^β with $\beta = 1.88$ 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 5 shows how, in the case of urban propagation, 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. Thus, the time to find a MWIS increases faster than $n^{1.88}$. However, for other values of Δ , the mean degree of the conflict graph appears to reach a plateau.

Note that the models shown in Figure 1 do not have the same exponent as the one given in the previous section. These can be reconciled by recognizing that the mean degree of the conflict graph can vary with n . For example, in the urban propagation case shown in Figure 1, the computation time grows like $n^{1.97}$, thus in order for (9) to hold with the parameters given above, we must have that for the topologies used to generate Figure 1 (a)

$$\frac{A \times n^{1.97}}{\text{mean degree of the conflict graph}} = 1.77 \times 10^{-8} n^{1.88},$$

which implies that the mean degree of the conflict graph must grow like $n^{0.09}$. Similarly, for the two-ray model and the two-ray with shadowing model, the models shown in Figure 1 (b) and (c) can be reconciled with (9) if the mean degree of the conflict graph is approximately $O(n^{-0.21})$ and $O(n^{0.0})$, respectively. Considering Figure 5, such a variation is plausible, especially, when Δ is small, as it is in the case of Figure 1.

8. 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 problem is solvable in

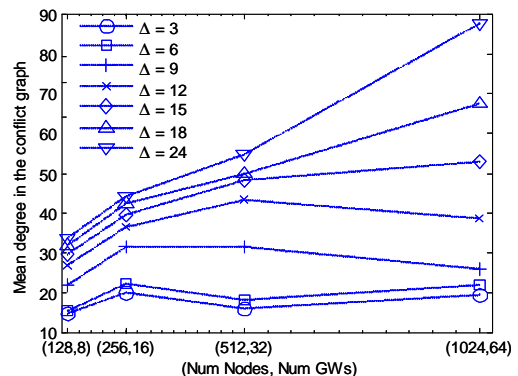


Figure 5: Relationships between the number of nodes and the mean degree of the conflict graphs.

many practical wireless scheduling problems. Specifically, 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. Moreover, 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, it is well known that in many cases the 3-SAT problem can be quickly solved [22, 23, 24, 25]. Moreover, the MWIS problem that arises in practical scheduling is particular subset of MWIS problems. The relationship between practical scheduling and the 3-SAT problem is not clear. Figure 6 show a possible Venn diagram of the set of problems. Note that it is unknown if there is any overlap between difficult 3-SAT problems and the MWIS problems that arise in practical wireless networks. The computational experiments in this paper indicate that there is not a significant overlap.

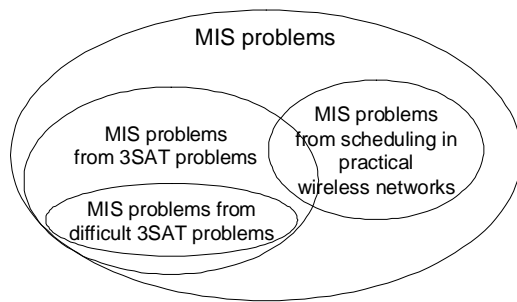


Figure 6: Possible Venn diagram of maximum independent set (MIS) problems, 3-SAT problems, and scheduling problems.

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.

Disclaimer

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U. S. Government.

9. REFERENCES

- [1] L. Tassioulas 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 scheduling in 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. Grottschel, 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, *Algorithms and Complexity*. AK Peters, Ltd, 2002, no. available at: <http://www.physics.it/lectures/AlgorithmComplexity.pdf>.
- [22] A. Goldberg, "On the complexity of the satisfiability problem," New York University, Tech. Rep. Courant Computer Science Report 16, 1979.
- [23] 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.
- [24] J. N. Hooker, "Resolution vs. cutting plane solution of interference problems: Some computational experience," *Operations Research Letters*, vol. 7, pp. 1–7, 1988.
- [25] 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.
- [26] 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
- [27] S. Bohacek and P. Wang, "Practical computation of optimal schedules in multihop wireless networks," submitted 2007, available at <http://udelmodels.eecis.udel.edu>.
- [28] P. Wang and S. Bohacek, "Communication models for capacity optimization in mesh networks," in *ACM PE-WASUN 2008*, Vancouver, Canada, 2008.
- [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] T. Rappaport, *Wireless Communications - Principles and Practice*. Prentice Hall, 2002.
- [31] S. Bohacek, V. Sridhara, and J. Kim, "UDel Models," available at: <http://udelmodels.eecis.udel.edu/>.