

UNIVERSITY OF CAMBRIDGE

TECHNICAL REPORT

**On the Transmission Scheduling of
Wireless Networks under SINR Constraints**

Themistoklis Charalambous, Evelina Klerides and Wolfram Wiesemann

CUED/F-INFENG/TR.649

February , 2010

© Cambridge University Engineering Department

On the Transmission Scheduling of Wireless Networks under SINR Constraints

Themistoklis Charalambous, Evelina Klerides and Wolfram Wiesemann

Abstract—In this paper we study a joint transmission scheduling and power control problem that arises in wireless networks. The goal is to assign time slots (or channels) and transmitting powers to communication links such that all communication requests are processed correctly, specified Quality-of-Service (QoS) requirements are met, and the number of required time slots is minimized. The first main result proves that the problem, also known as the wireless scheduling problem, is NP-hard. We then solve a mixed-integer linear programming (MILP) formulation of the problem with Branch & Bound (B&B) and Cutting Plane (CP) approaches. We enhance the computational performance of these schemes with heuristic procedures that provide tighter upper and lower bounds. We close with an extensive computational study, which shows that despite the complexity of the problem, the proposed methodology scales to problems of nontrivial size. Our algorithms can therefore serve as a benchmark for the performance evaluation of heuristic or distributed algorithms that aim to find near-optimal solutions without information about the whole network.

Index Terms—transmission scheduling, SINR, complexity, power control

I. INTRODUCTION

IN wireless networks, communicating nodes are equipped with half-duplex transceivers. This implies that at any point in time each node is limited to transmit only on a single channel of fixed range of frequency spectrum. In large networks there exist inevitably many transmissions and it is therefore unlikely that a channel will be used only by a single wireless node. Therefore, methods that provide bandwidth guarantees to wireless connections as well as optimize transmission scheduling are essential, and the design of multiple access mechanisms to efficiently control channel access is of vital importance. This problem has triggered numerous attempts for channel spatial reuse.

Each transmission corresponds to a spatiotemporal propagation of radio waves that are received by all nodes in proximity utilizing the same channel at the time. As a result, nodes interfere with each other when they use the same channel simultaneously. The level of interference depends on the received signals from all transmitters in the channel at the time. The higher the transmitting power from irrelevant nodes in the same channel, the higher the interference experienced at a receiver.

Nevertheless, wireless technology standards provide a radio-frequency (RF) spectrum with a set of many non-overlapping

channels, and a node has the option to choose on which channel to transmit. Likewise, in cases where only a single channel is available, it is possible to divide time into frames, and then frames can be divided into time slots, such that at each frame a node has the option to choose on which time slot to transmit. In the latter case, synchronization of the wireless nodes in the network is necessitated. If synchronization is not considered, however, choosing a channel or a time slot in the network becomes the same problem.

If many nodes transmit simultaneously in the same channel, the interference caused by these transmissions will prevent an intended receiver from receiving the signal successfully and the message is lost. On the other hand, if too few nodes transmit simultaneously in the same channel, valuable bandwidth is wasted, and the overall throughput of the network may suffer. Hence, the problem faced by any Media Access Control (MAC) layer is to find the subtle balance, in which a large number of devices transmit simultaneously and yet, the network is feasible, i.e. the resulting network interferences are kept below specified thresholds. Therefore, it is of vital importance to orchestrate channel access in order to fully exploit spatial reuse and hence, minimize both the number of time slots required to successfully complete all requests and the total power being dissipated in the network for the given number of time slots.

When studying wireless networks, the choice of model is crucial. Not only must the chosen model facilitate the design of protocols, but it also has to truthfully reflect the nature of the real network. The models more commonly used are classified into *graph-based* and *fading-channel* models, which we discuss in turn.

Graph-based models (such as the Protocol model [1] and Ideal model [2]) represent the wireless network as a graph and model transmission interferences by some graph property, which neglects the aggregated interference of nodes located further away. However, as shown in [3] graph-based scheduling algorithms are very conservative. they do not consider that the effect of simultaneous transmissions is cumulative, and hence they do not exploit the full potential for spatial reuse. For example, graph-based algorithms are unable to identify overlapping links that can be scheduled in the same time slot.

Fading-channel models depict real-world phenomena in wireless communications. These phenomena include multipath fading, shadowing and attenuation with distance. The most common fading-channel model being used is the physical model, which is thoroughly described in Section III. On a finer granularity, one distinguishes between the *geometric* and the *abstract* physical model. In the geometric physical model the channel gain between two nodes is solely determined

T. Charalambous is with the Department of Engineering, University of Cambridge, CB2 1PZ, UK e-mail: tc257@cam.ac.uk.

E. Klerides is with the Operational Research Group, Business School, Imperial College London, SW7 2AZ, UK e-mail: e.klerides@imperial.ac.uk.

W. Wiesemann is with the Department of Computing, Imperial College London, SW7 2AZ, UK e-mail: wwiesema@imperial.ac.uk.

by their spatial distance. Hence, simplifying assumptions are incorporated into this model, for example, the radios are perfectly isotropic and there are no obstructions [4]. In the abstract physical model the channel gain between two nodes incorporates all the real-world phenomena and hence no information can be extracted about the geometry of the network.

In this work, we consider the abstract physical model and determine a feasible network with a minimal number of employed time slots. Contrary to existing models, our formulation includes the choice of optimal transmitting powers. We prove in this paper that the transmission scheduling problem is also NP-hard for variable power levels. This setting has been described as an open problem in [5]. Indeed, it was unknown until now if there exists a tractable solution when the power levels of the interfering nodes are allowed to vary, despite the vast amount of attempts in the literature to solve the transmission scheduling problem in polynomial time. Then, we formulate the problem as a Mixed Integer Linear Program (MILP) and solve it using Branch-and-Bound (B&B) and Cutting Plane (CP) approaches in order to illustrate the complexity of the problem. Moreover, we refine the B&B and CP procedures with heuristics that determine tight upper and lower bounds on the problem and thereby enhance the performance of these methodologies.

The rest of the paper is organized as follows. In Section II we revise some previous approaches and comment on related work. In Section III the employed system model is described. Section IV presents the conditions for obtaining feasible networks, whereas Section V employs these conditions to formulate the transmission scheduling problem for variable power levels as an MILP. Section VI proves NP-hardness of the transmission scheduling problem for variable power levels, whereas Section VII suggests B&B and CP solution techniques to the scheduling problem. In Section VIII the validity of our formulation and the performance of our methodology are evaluated. Finally, in Section IX, conclusions are drawn and directions for future work are also given.

II. RELATED WORK

Many different optimization approaches to a wide variety of scheduling problems have been developed in the literature within the general framework of wireless networks, i.e., research for solutions targeting different optimization goals has been conducted, such as power dissipation, frame length and number of channels (or time slots) minimization.

Extensive research in the transmission scheduling area using graph-based models has resulted in a large number of centralized and decentralized algorithms, such as [6], [7] and [8]. However, for the sake of brevity we do not discuss previous works considering graph-based models, but instead we concentrate on those adopting fading channels and in particular, the physical model.

Power is a valuable resource in wireless networks, since the batteries of the wireless nodes have limited lifetime. As a result, power control has been a prominent research area for all kinds of wireless communication networks (e.g. [9]–[14]). More specifically, power control has been extensively

employed for MAC in multi-hop wireless networks (for example in [15]–[19]). Some of them aim to minimize power dissipation. For example, [16] proposes a two-phase method for the joint scheduling and power control which aims to find an admissible set of links along with their transmission power levels. [20] study the same problem as in [16], but it focuses on minimizing the scheduling length. Others, such as [21] and [17], aim to maximize throughput at the cost of increased power dissipation by allowing many simultaneous interference-limited transmissions. However, only some of the MAC schemes with power control consider transmission scheduling. In such schemes either time is divided into fixed-length slots or there exist many channels and the wireless nodes have to choose on which one to transmit. The rest of the section concentrates on the related work that considers transmission scheduling under the (geometric and abstract) physical model only.

In [22] it is proven that strategies in the geometric physical model that use uniform power assignment schemes (same power to all nodes in the network) or linear power assignment schemes (power levels proportional to the minimum power required to reach the receiver node) have a bad scheduling complexity. In addition, they propose a power-assignment algorithm that successfully schedules a network using a poly-logarithmic number of time slots. In [23] an approximation scheduling algorithm is proposed that computes a feasible solution in polynomial time for the geometric model with worst-case approximation guarantees for arbitrary network topologies when the power levels are constant. Then, [24] removed the logarithmic approximation of [23] and they show that constant parameter and model changes modify the result only by a constant. In [25] it is shown that solutions with oblivious power assignments (the power level of a node depends only on the transmitter-receiver distance) cannot compete with solutions using possibly different power levels and channels for a network. However, they are capable of achieving nearly the same performance as solutions restricted to symmetric power and channel assignments.

Despite the vast amount of literature for the geometric physical model, the computation of efficient schedules in the abstract physical model has been studied in a more limited number of papers. Previous attempts to determine the minimum number of time slots required, such that all the communication pairs have at least one time slot to communicate at the required SINR constraints, were focused on heuristics (e.g. [26]), which do not necessarily provide an optimal solution. In [27], the link assignment optimization problem was formulated. The goal is to assign at least one time slot to each communication pair such that the number of time slots is minimized. However, the authors assume that all power levels have been chosen a priori and therefore constitute an input of the optimization problem. In this setting, the authors show that for uniform power levels, the problem is equivalent to the NP-hard edge coloring problem. In [28], we extend this line of research and formulate the transmission scheduling problem for variable power levels. However, in this paper the complexity of the problem is not discussed and the solution relies on standard, non problem-specific Branch-and-Bound (B&B) techniques.

The complexity of the problem is important since it determines whether the transmission scheduling problem is tractable or not, with the view to deciding the direction for future research on efficient centralized and eventually distributed algorithms.

III. MODEL

The system model can be divided into two levels: the network as a whole and the channel. Thus, we have the network model and the channel model. The network model concerns the general topology of the nodes and their characteristics. The channel model describes the assessment of the link quality between communication pairs and the interaction between the nodes in the network.

A. Network Model

In this study, we consider a network where the links are assumed to be unidirectional and each node is supported by an omnidirectional antenna. For a planar network (easier to visualize without loss of generality), this can be represented by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of all nodes and \mathcal{L} is the set of the active links in the network. Each node can be a receiver or a transmitter only at each time instant due to the half-duplex nature of the wireless transceiver. Each transmitter aims to communicate with a single node (receiver) only, which cannot receive from more than one node simultaneously. We denote by \mathcal{T} the set of transmitters and \mathcal{R} the set of receivers in the network.

B. Channel Model

The link quality is measured by the Signal-to-Interference-and-Noise-Ratio (SINR). The channel gain on the link between transmitter i and receiver j is denoted by g_{ij} and incorporates the mean path-loss as a function of distance, shadowing and fading, as well as cross-correlations between signature sequences. All the g_{ij} 's are positive and can take values in the range $(0, 1]$. Without loss of generality, we assume that the intended receiver of transmitter i is also indexed by i . The power level chosen by transmitter i is denoted by p_i . ν_i denotes the variance of thermal noise at the receiver i , which is assumed to be additive Gaussian noise. The interference

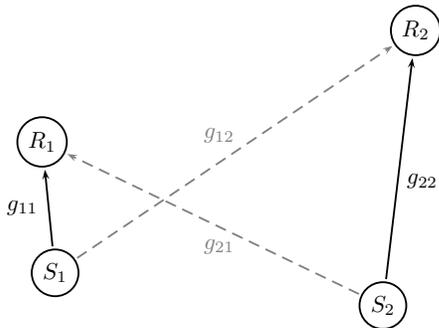


Fig. 1. An example of a network consisting of two communication pairs only. Each pair i consists of a transmitter S_i and a receiver R_i connected with a solid line while the grey dotted arrows indicate the interference that transmitters cause to the neighboring receivers.

power at the i^{th} node, I_i , includes the interference from all

the transmitters in the network and the thermal noise, and is given by

$$I_i = \sum_{j \neq i, j \in \mathcal{T}} g_{ji} p_j + \nu_i. \quad (1)$$

Therefore, the SINR at the receiver i is given by

$$\Gamma_i = \frac{g_{ii} p_i}{\sum_{j \neq i, j \in \mathcal{T}} g_{ji} p_j + \nu_i}. \quad (2)$$

Due to the unreliability of the wireless links, it is necessary to ensure Quality of Service (QoS) in terms of SINR in wireless networks. Hence, independently of nodal distribution and traffic pattern, a transmission from transmitter i to its corresponding receiver is successful (error-free) if the SINR of the receiver is greater or equal to the *capture ratio* γ_i ($\Gamma_i \geq \gamma_i$). The value of γ_i depends on the modulation and coding characteristics of the radio. Therefore,

$$\frac{g_{ii} p_i}{\sum_{j \neq i, j \in \mathcal{T}} g_{ji} p_j + \nu_i} \geq \gamma_i \quad (3)$$

IV. PRELIMINARIES

Inequality (3) depicts the QoS requirement of a communication pair i while transmission takes place. After manipulation it becomes equivalent to the following

$$p_i \geq \gamma_i \left(\sum_{j \neq i, j \in \mathcal{T}} \frac{g_{ji}}{g_{ii}} p_j + \frac{\nu_i}{g_{ii}} \right). \quad (4)$$

In matrix form, for a network consisting of n communication pairs, this can be written as

$$\mathbf{p} \geq \Gamma \mathbf{G} \mathbf{p} + \boldsymbol{\eta} \quad (5)$$

where

$$\begin{aligned} \Gamma &= \text{diag}(\gamma_i) \\ \mathbf{p} &= (p_1 \ p_2 \ \dots \ p_n)^T \\ G_{ij} &= \begin{cases} 0 & , \text{ if } i = j, \\ \frac{g_{ji}}{g_{ii}} & , \text{ if } i \neq j. \end{cases} \\ \eta_i &= \frac{\gamma_i \nu_i}{g_{ii}} \end{aligned}$$

Let $C = \Gamma G$, so that (5) can be written as

$$(I - C) \mathbf{p} \geq \boldsymbol{\eta} \quad (6)$$

The matrix C has nonnegative elements and it is reasonable to assume that is irreducible, since we are not considering totally isolated groups of links that do not interact with each other. By the Perron-Frobenius theorem [29], we have that the spectral radius of the matrix C is a simple eigenvalue, while the corresponding eigenvector is positive componentwise. The necessary and sufficient condition for the existence of a nonnegative solution to inequality (6) for every positive vector $\boldsymbol{\eta}$ is that $(I - C)^{-1}$ exists and is nonnegative. However, $(I - C)^{-1} \geq 0$ if and only if $\rho(C) < 1$ [30] (Theorem 2.5.3), [31], where $\rho(C)$ denotes the spectral radius of C .

Therefore, the necessary and sufficient condition for (6) to have a positive solution \mathbf{p}^* for a positive vector $\boldsymbol{\eta}$ is that the Perron-Frobenius eigenvalue of the matrix C is less than 1. In

brief, a network described by matrix C is feasible if and only if $\rho(C) < 1$.

Furthermore, a sufficient condition to establish feasibility to the system without requiring the knowledge of the whole matrix C , could be $\|C\|_{\infty} < 1$, i.e.

$$\frac{g_{ii}}{\sum_{j \neq i, j \in \mathcal{T}} g_{ji}} > \gamma_i \quad \forall i \in \mathcal{R}, \quad (7)$$

which is equivalent to (3) for constant, equal power levels. Since $\rho(C) \leq \|C\|_{\infty}$, this condition is more conservative. It therefore provides an upper bound on the achievable target SINR levels in a given network, and hence leads to a soft capacity constraint for the underlying system.

V. PROBLEM FORMULATION

In this section, we present the problem of finding the minimum possible number of time slots (or channels) and the corresponding transmitting powers, such that all communication requests are being processed correctly and Quality of Service (QoS) requirements for successful transmissions are satisfied.

Note that to ensure feasibility of our problem we can define the deadline of the network as follows. The maximum number of time slots that may be required is equal to the number of links, $|\mathcal{L}|$. Henceforth, we will assume that the first time slot is at time 1; the latest point in time for which there can be a scheduled transmission is therefore $D = |\mathcal{L}|$. The notation used for the networks in this paper is given below (in Notation 1).

Notation 1 Notation used for the networks:

\mathcal{N}	The set of all nodes in the network
\mathcal{L}	The set of active links in the network
$\mathcal{G} = (\mathcal{N}, \mathcal{L})$	The graph of the network
\mathcal{T}	The set of transmitters in the network
\mathcal{R}	The set of receivers in the network
g_{ij}	The channel gain on the link $i \rightarrow j$
ν_i	The variance of thermal noise at the receiver i
I_i	The interference power at the i^{th} receiver
Γ_i	The SINR at the i^{th} receiver
γ_i	The capture ratio at the i^{th} receiver
D	The deadline of the network

To formulate the optimization problem, we define two sets of decision variables, for each transmitter $i \in \mathcal{T}$ and time $t = 1, \dots, D$; processing-time variables:

$$x_i(t) = \begin{cases} 1, & \text{if transmitter } i \text{ is active at time } t \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

and power level variables: $p_i(t) \in \mathbb{R}_+$.

Since the problem involves both integer and continuous decision variables, the mathematical formulation is classified as a Mixed Integer Program (MIP) and is given in Model 1.

Model 1 Minimum number of time slots

$$\text{minimize}_{t,x,p} \tau = \max_{i \in \mathcal{T}} \sum_{t=1}^D tx_i(t) \quad (9a)$$

subject to

$$\sum_{t=0}^D x_i(t) \geq 1 \quad \forall i \in \mathcal{T}, \quad (9b)$$

$$x_i(t) = 0 \Rightarrow p_i(t) = 0 \quad \forall i \in \mathcal{T}, t = 1, \dots, D, \quad (9c)$$

$$x_i(t) = 1 \Rightarrow g_{ii}p_i(t) \geq \gamma_i \left(\sum_{j \in \mathcal{T}, j \neq i} g_{ji}p_j(t) + \nu_i \right) \quad (9d)$$

$$\forall i \in \mathcal{T}, t = 1, \dots, D,$$

$$x_i(t) \in \{0, 1\} \quad \forall i \in \mathcal{T}, t = 1, \dots, D, \quad (9e)$$

$$p_i(t) \in \mathbb{R}_+ \quad \forall i \in \mathcal{T}, t = 1, \dots, D. \quad (9f)$$

Objective (9a) minimizes the number of time slots needed to schedule all the transmitters in the network. τ is equal to the latest point in time needed for scheduling a transmission. Constraint (9b) ensures that each link in the network is processed at least once in the schedule. Note that there is always an optimal schedule in which each link is processed only once. Constraint (9c) makes sure that if a pair is not processed at a specific time slot, then the power level of the corresponding transmitter is 0 at that time slot. The QoS conditions are guaranteed by constraint (9d). The constraint only affects the optimization if $x_i(t)$ takes the value 1. Finally, the last two constraints (9e) and (9f) define the admissible values for the decision variables.

VI. COMPUTATIONAL COMPLEXITY

Theorem 1. *Problem (9) is NP-hard.*

Proof: The statement of Theorem 1 is equivalent to saying: deciding whether the optimal value of (9) does not exceed a given value $\kappa \in \mathbb{N}$ is NP-hard. We construct a polynomial-time reduction of the Graph Coloring problem, which is well known to be NP-hard [32]. Given an undirected graph $G = (V, E)$ with nodes $V = \{1, \dots, n\}$ and edges $E \subseteq \{\{i, j\} : i, j \in V, i \neq j\}$, as well as a scalar $k \in \mathbb{N}$, the Graph Coloring problem asks whether there is an assignment $f : V \mapsto \{1, \dots, k\}$ of nodes to k colors such that $f(i) \neq f(j)$ for all $\{i, j\} \in E$, that is, neighbouring nodes must have different colors. Our reduction takes as input a Graph Coloring instance and generates an instance of (9) such that the optimal value of problem (9) does not exceed $\kappa = k$ if and only if the answer to the Graph Coloring problem is affirmative. Towards this end, we set $\mathcal{T} := V$, $D := |V| = n$, $\mathcal{L} := E$, $\gamma = (1, \dots, 1)^T$, $\nu_i = 0 \quad \forall i \in \mathcal{R}$ and

$$g_{ij} := \begin{cases} 1 & \text{if } \{i, j\} \in E, \\ 1/2 & \text{if } i = j, \\ 1/(2n) & \text{otherwise.} \end{cases}$$

The size of this reduction is polynomial in the size of the Graph Coloring instance. Hence, if we show that the optimal value of (9) does not exceed κ if and only if the answer to the Graph Coloring instance is affirmative, we have proven

that the solution of (9) is NP-hard. We proceed in two steps. Firstly, we show that if there is a graph coloring that uses ζ colors, then the optimal value of (9) is smaller or equal to ζ . Secondly, we show that if there is a feasible solution for (9) of value ζ , then we can construct an admissible ζ -coloring for the graph coloring instance. The assertion follows from the combination of both arguments and the fact that we consider a minimization objective.

As for the first step, assume that there exists a coloring $f : V \mapsto \{1, \dots, \zeta\}$. Given this coloring, we construct a feasible solution (x, p) of objective value ζ . Towards this end, we set $x_i(t) := 1$ if $f(i) = t$ and $x_i(t) := 0$ otherwise, $\forall i \in \mathcal{T}$. Likewise, set $p_i(t) := \hat{p}_i$ ($\hat{p}_i \in \mathbb{R}_+$) if $f(i) = t$ and $p_i(t) := 0$ otherwise, for all $i \in \mathcal{T}$ and $t = 1, \dots, D$. By construction, constraints (9b), (9c), (9e) and (9f) are satisfied, for any value \hat{p}_i , $\hat{p}_i \in \mathbb{R}_+$. For $i \in \mathcal{T}$ and $t \in \{1, \dots, D\}$ with $x_i(t) = 1$, constraint (9d) requires that

$$p_i(t) \geq \left(2 \sum_{\{i,j\} \in E} p_j(t) \right) + \frac{1}{n} \sum_{\substack{j \in V, j \neq i, \\ \{i,j\} \notin E}} p_j(t).$$

Since f constitutes a valid coloring, the first term on the right hand-side must evaluate to zero, because otherwise, the spectral radius ρ of the matrix that is constituted by $\{i, j\} \in E$ is greater than 1 and hence the network would be infeasible, as described in Section IV. On the other hand, when the first term of the right hand-side is zero, the second term fulfills the inequality, since

$$\sum_{\substack{j \in V, j \neq i, \\ \{i,j\} \notin E}} \frac{1}{n} < 1 \Rightarrow \|C\|_\infty < 1,$$

and therefore $\rho(C) < 1$, where C consists of i and $j \in V$, $j \neq i$, $\{i, j\} \notin E$. We conclude that constraint (9d) is also satisfied by our choice of x and $p \in \mathbb{R}_+$. Note that the objective function (9a) evaluates to ζ for the constructed solution (x, p) , which implies that the optimal value of (9) must be smaller or equal to ζ .

In the second step, we use a feasible solution to (9) with objective value ζ to construct a valid coloring of the graph G with at most ζ colors. Assume that we have a feasible solution (x, p) for problem (9) with objective value ζ . Since we consider a minimization problem, without loss of generality we can assume that $\sum_{t=1}^D x_i(t) = 1$. Hence, we obtain a function if we set $f(i) := t$ if and only if $x_i(t) = 1$. Furthermore, since the objective value (9a) of (x, p) is ζ , the range of f is limited to $\{1, \dots, \zeta\}$. We now show that f constitutes a valid coloring of graph G , that is, $f(i) \neq f(j)$ for all $\{i, j\} \in E$. Assume to the contrary that there is $\{\hat{i}, \hat{j}\} \in E$ with $f(\hat{i}) = f(\hat{j}) = \hat{t}$. In this case, (x, p) must satisfy the constraints

$$p_i(\hat{t}) \geq 2 \sum_{\{\hat{i}, j\} \in E} p_j(\hat{t}) + \frac{1}{n} \sum_{\substack{j \in V, j \neq \hat{i}, \\ \{\hat{i}, j\} \notin E}} p_j(\hat{t}) \geq 2p_i(\hat{t})$$

and

$$p_j(\hat{t}) \geq 2 \sum_{\{\hat{j}, i\} \in E} p_i(\hat{t}) + \frac{1}{n} \sum_{\substack{i \in V, i \neq \hat{j}, \\ \{\hat{j}, i\} \notin E}} p_i(\hat{t}) \geq 2p_j(\hat{t}),$$

which contradicts the assumption that (x, p) is feasible for problem (9). We conclude that the constructed function f indeed constitutes a valid ζ -coloring of the graph G . As a result, problem (9) is NP-hard. ■

VII. SOLUTION APPROACHES

In this section, we propose solution approaches for the ‘‘Minimum number of time slots’’ problem (Model 1). In particular, we aim at developing efficient bounding techniques as well as exact solution methodologies which provide an optimal solution quickly. To this end, we describe an upper bounding technique in Section VII-A and two lower bounds in Section VII-B. These in turn will be incorporated in two exact solution methodologies, namely a Branch-and-Bound (B&B) (Section VII-C) and a Cutting Plane approach (Section VII-D).

Note that despite the fact that we are able to find the exact solutions to the problem with the aid of bounding techniques, the methods still possess an exponential time complexity in the worst case. However, they are able to find the solution for small networks considerably fast and they also constitute a good benchmark for the evaluation of forthcoming approximation methods that aim to solve the wireless scheduling problem.

A. Upper bound UB

When computational time and memory is limited, finding the optimal solutions to problems can be difficult, particularly when dealing with large-sized instances. We describe a simple yet effective heuristic, based on a priority scheduling policy. The derived solution value, referred to as *UB*, serves as a cutoff value in the B&B approach described in the next subsection. The basic idea of the policy is to keep adding new transmissions at the current time slot according to the priorities, until no more transmissions can be scheduled without violating the SINR constraints. In such a case, the next time slot is considered, and the process is repeated for all the remaining unscheduled transmission pairs. Note that for each node considered for a time slot, the spectral radius of the matrix that constitutes the network is calculated, which takes time $O(n^3)$.

It is evident that any priorities may be given to the transmission pairs. In our computational results, the priority value of transmitter $i \in \mathcal{T}$ is found using

$$prio_i = \frac{\nu_i \gamma_i}{g_{ii}}, \quad (10)$$

which is basically the power of transmitter i when no other transmitter is active simultaneously. The complete heuristic algorithm is described in Algorithm 1 below. Note that set A contains all transmission pairs in decreasing order of their priorities.

Algorithm 1 Upper bounding technique

```

initialise
   $S = \emptyset$ 
   $A = \{i_1, \dots, i_{|\mathcal{T}|} \mid i_k \in \mathcal{T}, k = 1, \dots, |\mathcal{T}|, p_{i_{k_1}} \geq p_{i_{k_2}} \text{ if } k_1 \leq k_2\}$ 
   $UB \leftarrow 0$ 

while  $S \neq A$  do
   $set = \emptyset$ 
  for  $j \in A$  do
    if  $j \cup set$  satisfies SINR constraints then
       $set \leftarrow set \cup \{j\}$ 
       $A \leftarrow A \setminus \{j\}$ 
       $S \leftarrow S \cup \{j\}$ 
    end if
  end for
   $UB \leftarrow UB + 1$ 
end while

return  $UB$ 

```

B. Lower bounds LB1 and LB2

Lower bounds are often used in optimization problems to benchmark heuristic solution values whenever the optimal solutions are unknown. In this paper, we use lower bounds to achieve quick convergence to the optimal solution as well as evaluating the heuristic approach described in Section VII-A.

a) *Lower bound LB1*: The first lower bounding technique is based on the idea that for each transmitter $j_0 \in \mathcal{T}$, we can define the set NA_{j_0} of transmitters $j_1 \in \mathcal{T}$ for which j_0 and j_1 cannot be simultaneously scheduled without violating the SINR constraints. This means that if j_0 is scheduled at time $t_0 = 1$, then any $j_1 \in NA_{j_0}$ must be scheduled at time $t_1 \geq 2$, say $t_1 = 2$, without loss of generality. The set of all remaining pairs, $set = \mathcal{T} \setminus NA_{j_0}$, is implicitly assumed to be executed at t_0 . Using the same logic, if $NA_{j_1} \cap NA_{j_0} \neq \emptyset$, then $j_2 \in NA_{j_1} \cap NA_{j_0}$ must also be assigned a different time slot, $t_2 \neq t_0, t_1$, say $t_2 = 3$. Now the set of remaining pairs, $set = \{\mathcal{T} \cap NA_{j_0}\} \setminus NA_{j_1}$, is implicitly assumed to be executed at t_1 . If we repeatedly use the same principle, we will eventually reach iteration K for which $\bigcap_{k=0}^K NA_{j_k} = \emptyset$

and $set = \{\mathcal{T} \bigcap_{k=0}^{K-1} NA_{j_k}\} \setminus NA_{j_K}$ is allowed to be executed at time K ; the lower bound $LB1$ so far is equal to K . In polynomial time, we can check whether any of the members in set cannot be pairwise simultaneously executed (using sets NA_j for all $j \in set$). If we can find such ‘‘infeasible’’ pairs, then the lower bound $LB1$ is increased by 1 and is therefore equal to $K + 1$. To explain this, we note that the members of the infeasible pair cannot be both present in the final slot. Furthermore, neither of the members can be scheduled at any previous time slots (by construction of the lower bound). Hence, one of the members of the pair must be scheduled in a new time slot.

The full description of the lower bounding technique is given in Algorithm 2.

Algorithm 2 Lower bounding technique

```

initialise
   $\forall j \in \mathcal{T}$ , let  $NA_j = \{i \in \mathcal{T}, i \neq j \mid (i, j) \text{ cannot transmit simultaneously without violating SINR constraints}\}$ .
   $A = \{1, \dots, |\mathcal{T}|\}$ 
   $j \leftarrow A(0) = 1$ ,
   $set = \emptyset, LB1 \leftarrow 1$ 

while  $A \neq \emptyset$  do
   $set = A \setminus NA_j$ 
   $A \leftarrow A \cap NA_j$ 
  if  $A \neq \emptyset$  then
     $j \leftarrow A(0)$ 
     $LB1 \leftarrow LB1 + 1$ 
  end if
end while
for  $i, j \in set, i < j$  do
  if  $j \in NA_i$  then
     $LB1 \leftarrow LB1 + 1$ 
    Exit loop.
  end if
end for

return  $LB1$ 

```

b) *Lower bound LB2*: The second lower bounding technique is based on solving a relaxation of Model 1 which ignores the complicating constraints (9d). The derived optimal solution value of the relaxed model is set to be equal to the second lower bound, referred to as $LB2$. The relaxed model is strengthened using constraints:

$$x_j(t) + x_i(t) \leq 1, \quad \forall j \in \mathcal{T}, i \in NA_j, t \in [0, D] \quad (11)$$

C. Branch & Bound Approach

Branch-and-Bound (B&B) is one of the most widely applied and studied implicit enumeration algorithms for finding optimal solutions of various optimization problems involving discrete decision variables. It consists of a systematic construction of candidate solutions and calculations of upper and lower bounds on the objective function to speed up the search process.

The branching process can be seen as building up a tree, where the *relaxed linear* problem under investigation is placed in the root and new problems are stored in child nodes. Each child node is a copy of its parent node plus some additional constraints. The B&B algorithm defines which additional constraints are added (*branching strategy*) and which child node is to be solved first (*node selection strategy*).

Branching: To demonstrate the branching process, consider the following simple Integer Program as being the root node in the B&B tree. Note that the example has been taken from [33].

Model 2 Simple Integer Program

$$\underset{x}{\text{minimize}} c^T x \quad (12a)$$

subject to

$$Ax = b \quad (12b)$$

$$x_i \in \{a_i, a_i + 1, \dots, b_i\} \forall i \quad (12c)$$

where $\{a_i, a_i + 1, \dots, b_i\}$ is the set of all integers from a_i to b_i . The B&B procedure starts by replacing the integrality constraints (12c) with $a_i \leq x_i \leq b_i$ for all i . The resulting model is the so-called *relaxed linear* program, whose solution is x_{LP} . If x_{LP} happens to satisfy integrality conditions, the algorithm is terminated because the optimal solution has been found, since integrality is satisfied without being enforced. Otherwise, a lower bound $z_{LB} = c^T x_{LP}$ on the true optimal objective is obtained, since the solution to a relaxation of a minimization problem yields a lower bound. The B&B algorithm continues with the branching procedure, i.e. by picking one (non-integer) variable x_j , called the branching variable, and one integer $d_j \in [a_i, b_i]$. We then replace our original problem (Model 2) by two similar problems (child nodes):

Model 3 Child node 1

$$\underset{x}{\text{minimize}} c^T x \quad (13a)$$

subject to

$$Ax = b \quad (13b)$$

$$x_i \in [a_i, b_i] \forall i \neq j \quad (13c)$$

$$x_j \in [a_i, d_j] \quad (13d)$$

Model 4 Child node 2

$$\underset{x}{\text{minimize}} c^T x \quad (14a)$$

subject to

$$Ax = b \quad (14b)$$

$$x_i \in [a_i, b_i] \forall i \neq j \quad (14c)$$

$$x_j \in [d_j + 1, b_j] \quad (14d)$$

Node selection: Once the next node to solve is selected, either of the following three cases may occur:

- 1) The solution leads to an infeasibility, in which case the node is fathomed (further exploration of that part of the tree is prohibited).
- 2) The solution leads to an integral solution x_{int} , in which case we fathom the node and update the best objective function value found so far, $z(best) = \min[z(best), c^T x_{int}]$. If $z(best) = z_{LB}$, the optimal solution has been found and the B&B is terminated.
- 3) The solution leads to a non-integral solution, x_{nint} . In the latter case, if the obtained objective value $c^T x_{nint}$ is greater than $z(best)$, then the node is dropped as it

is impossible to obtain the optimal solution from that part of the solution space. If $c^T x_{nint} \leq z(best)$, then we continue branching, as we did in Model 2.

Because the solution space is finite, the algorithm will eventually terminate, either without finding an integral solution (infeasible problem), or by giving the optimal integral solution. In our case, we avoid the possibility of infeasibility; there always exist a solution which is the assignment of a separate time slot to each link since we can use up to D time slots (equal to the number of links in the network).

Much research in integer programming concerns how to choose the best branching and node selection strategy. However, the focus in this paper is not the methodology itself, but its application in the context of wireless networks, so we refer the reader to [34] for more details on such strategies. For the purposes of this paper, we implement two B&B approaches, referred to as BB_{slots}^1 , which at the initialization stage sets $z_{LB} = LB1$, and as BB_{slots}^2 , which at the initialization stage sets $z_{LB} = LB2$, ($LB1$ and $LB2$ have been described in Section VII-B). Both B&B methodologies use UB (described in Section VII-A) as the initial cutoff value $z(best)$.

D. Cutting Plane Approach

In this section, we propose a cutting plane approach, referred to as CP_{slots} , to construct the ‘‘Minimum number of time slots’’ problem (Model 1). The basic idea is to partition the problem into a Master Problem and a Subproblem. The iterative procedure solves the Master Problem first and uses the obtained optimal decision values to solve the Subproblem. Depending on whether the Subproblem is feasible or infeasible, cuts may be derived. These cuts are then added to the Master Problem and the procedure is repeated. The algorithm terminates when the solution obtained at any iteration satisfies the optimality/termination criteria.

Problem partitioning. Model 1 may be partitioned into a Master Problem (Model 5) and a Subproblem (Model 6), as shown below:

Model 5 Master problem

$$\underset{x}{\text{minimize}} \tau = \max_{i \in \mathcal{T}} \sum_{t=1}^D tx_i(t) \quad (15a)$$

subject to

$$\sum_{t=0}^D x_i(t) \geq 1 \quad \forall i \in \mathcal{T} \quad (15b)$$

$$x_i(t) \in \{0, 1\} \quad \forall i \in \mathcal{T}, t \in [0, D] \quad (15c)$$

Model 6 Subproblem

$$\text{minimize } \sum_p \sum_{i \in \mathcal{T}} \sum_{t=0}^D p_i(t) \quad (16a)$$

subject to

$$x_i^*(t) = 0 \Rightarrow p_i(t) = 0 \quad \forall i \in \mathcal{T}, t = 1, \dots, D, \quad (16b)$$

$$x_i^*(t) = 1 \Rightarrow p_i(t) g_{ii} \geq \gamma_i \left(\sum_{j \in \mathcal{T}, j \neq i} g_{ji} p_j(t) + \nu_i \right) \quad (16c)$$

$$\begin{aligned} \forall i \in \mathcal{T}, t = 1, \dots, D, \\ p_i(t) \geq 0 \quad \forall i \in \mathcal{T}, t \in [0, D] \end{aligned} \quad (16d)$$

Let $x_i^*(t, k)$ denote the optimal values of the processing time variables obtained by solving Model 5 at iteration k . For brevity, we will denote these as $x^*(k)$. Using $x^*(k)$, Model 6 decomposes into distinct problems for each time t derived from $x^*(k)$. Each of the decomposed problems corresponds to a time t and reduces to the problem of finding the minimum total transmission power for the schedule defined by all transmissions i such that $x_i^*(t, k) = 1$. In fact, the objective function of the Subproblem may be changed as we are only testing the feasibility of the solution derived from the Master problem.

Given $x^*(k)$, the Subproblem may be infeasible; the schedule derived from the Master problem may assign the same slot to transmissions which cannot be simultaneously processed without violating SINR constraints.

Cut generation. If there is a time t such that the simultaneous transmission of all $i \in \mathcal{T}$ for which $x_i^*(t, k) = 1$ is infeasible with respect to the Subproblem (Model 6), the following feasibility cuts may be applied to the Master problem (Model 5).

Let $EX_t(k) = \{i \in \mathcal{T} | x_i^*(t, k) = 1\}$. The following cut forbids the simultaneous processing of all the transmission pairs in $EX_t(k)$:

$$\sum_{i \in EX_t(k)} x_i(t') < |EX_t(k)|, \quad \forall t' \in [0, D] \quad (17)$$

These cuts should be applied for $t \in [0, D]$ for which the corresponding scheduled transmissions are violating the SINR constraints. The cuts are valid (global) feasibility cuts for the original problem and, as such, can be imposed at each iteration and of the algorithm.

Additional to the cuts applied at each iteration of the algorithm, we can apply cuts (11) before the very first iteration. These cuts help in achieving a quicker convergence to the optimal solution.

Termination criteria. Let $MP^*(k)$ denote the value of the optimal solution to Model 5 obtained at iteration k . τ_{min} denotes the value of the optimal solution to Model 1. The cutting plane algorithm proceeds iteratively until Model 6 becomes feasible. This termination criterion guarantees the optimality of the solution; we provide a proof for this below.

Proof.

1) At each iteration k , the Master Problem (Model 5) is a relaxation of the original problem (Model 1). Hence, its

optimal solution value, $MP^*(k)$, is a lower bound to τ_{min} , i.e. $MP^*(k) \leq \tau_{min}$.

- 2) At each iteration k , and before the algorithm terminates, a new feasibility cut (distinct from cuts found at all previous iterations) is found by solving the Subproblem.
- 3) Since there are only a finite number of cuts available (corresponding to each possible set of transmission pairs), the algorithm is guaranteed to converge. For the final iteration K , the Subproblem is feasible. Therefore, the values of the processing time variables for that iteration represent a feasible solution to the original problem. Hence, $MP^*(K) \geq \tau_{min}$. Since $MP^*(K) \leq \tau_{min}$ (from the first point), we have $MP^*(K) = \tau_{min}$, and thus convergence to the optimal solution has been achieved.

The complete cutting plane algorithm is summarized below.

Algorithm 3 Cutting plane algorithm for the ‘‘Minimum number of time slots’’ problem

initialise

$k = 1$

Compute LB {Lower bound, obtained from either of the lower bounding techniques, described in Section VII-B}

Compute UB {Upper bound, obtained from the heuristic, described in Section VII-A}

$feas = false$ {feasibility indicator, obtained from solving the Subproblem (Model 6)}

Add cuts of the form (11) to Model 5.

if $LB < UB$ **then**

repeat

Solve Model 5 to get optimal values $x^*(k)$ and $MP^*(k)$. Set $LB = \max[MP^*(k), LB]$.

Solve Model 6 to check feasibility of SINR constraints.

if Model 6 is feasible **then**

$feas = true$

else

Add cuts of the form (17) to Model 5.

end if

$k = k + 1$

until $feas = true$ or $LB = UB$

end if

$K = k - 1$

return $LB, x^*(K)$.

The algorithm begins with the computation of a lower bound (LB) and upper bound (UB) according to either of the techniques described in Section VII-B and the technique described in VII-A, respectively. The initial cuts of the form (11) are also determined and added to the Master Problem (Model 5) at this stage.

It is evident that if LB has the same value as UB , the cutting plane algorithm is redundant; the heuristic solution would be optimal for the transmission scheduling problem.

If $LB < UB$, the problem scale is reduced by setting the deadline of the problem $D = UB$; the Master Problem (Model 5) is then solved and the lower bound of the algorithm is updated. The Subproblem (Model 6) is then examined and the SINR constraints are tested for feasibility. If infeasibilities occur, cuts of the form (17) are added to Model 5 and the process is repeated. The algorithm terminates when the solution obtained from Model 5 is either feasible with respect to the SINR constraints or the optimal objective value of the Master problem, $MP^*(k)$, at iteration k equals UB (the solution value found by the heuristic at the initialization stage of Algorithm 3). In the latter case, the optimal objective function value for the transmission scheduling problem is equal to UB and the optimal schedule would be the one found using the priority scheduling policy.

For the computational study in Section VIII, the lower bound used in CP_{slots} is $LB2$. Further to CP_{slots} , we also develop another cutting plane, referred to as CP_{cuts} . The latter technique is similar to CP_{slots} , however, the initial bounds are not used. A comparison between the computational performance of CP_{slots} and CP_{cuts} will emphasise the importance of using initial lower bounds.

VIII. PERFORMANCE EVALUATION

Algorithms BB_{slots}^1 , BB_{slots}^2 (Section VII-C), CP_{cuts} and CP_{slots} (Section VII-D) have been coded in Microsoft Visual Studio 2005 C++ using CPLEX v12.1 and run on an Intel Core 2 computer, with 2.5GHz processor and 3.5GB of RAM. For benchmarking purposes, we also code another B&B approach, referred to as BB_{gen} , which is generic as it only implements CPLEX's default settings (no bounds are incorporated).

We demonstrate the solution obtained by solving the scheduling problem with an example. The graph of the network is given in Figure 2 and includes $|\mathcal{T}| = |\mathcal{R}| = 10$ communication pairs. Green nodes denote the transmitters, whereas red ones denote the receivers.

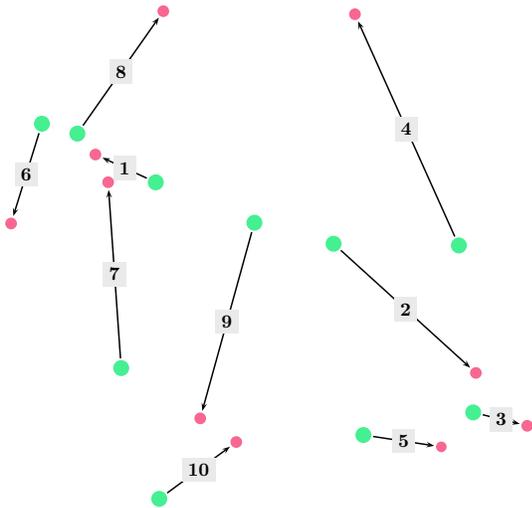


Fig. 2. A wireless ad hoc network with 10 communication pairs.

The optimal schedule found by solving Model 1 using BB_{slots}^1 for the above example is given in Table I. The

numbers in parentheses in Table I are the optimal power levels assigned to each communication pair. The minimum number of time slots required for a successful communication is 4.

Time Slot	Links in process
1	5 (15820.5), 8 (385019), 9 (579226)
2	3 (17739.8), 4 (763932), 7 (355380)
3	2 (206559), 6 (24638.4)
4	1 (2924.78), 10 (166.464)

TABLE I
MINIMUM TRANSMISSION SCHEDULE FOR THE EXAMPLE NETWORK
USING BB_{slots} .

The solution is visualized in Figure 3. The colors on the links show the scheduling time for each communication pair. Hence, links with similar colors are active simultaneously.

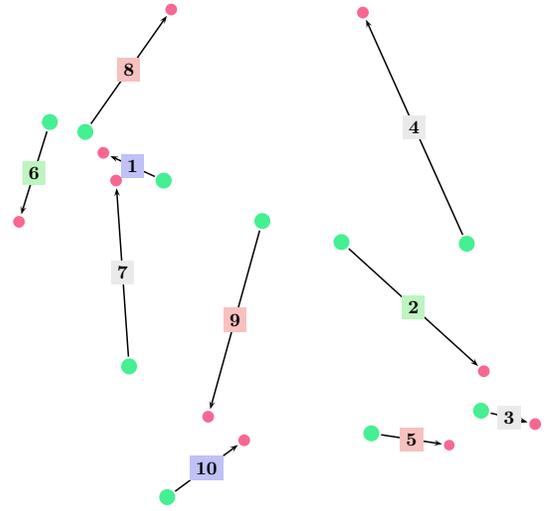


Fig. 3. Illustrative example where the transmission scheduling for the Physical model is solved for this network (using BB_{slots}).

The feasibility of our solution is verified by proving that each of the subnetworks formed is a feasible network and hence all the communication pairs can be active simultaneously. This is done by forming matrix C and checking if the Perron-Frobenius eigenvalue is less than one. The validity of the power levels can be found in a centralized way by

$$\mathbf{p}_s^* \geq (I - C_s)^{-1} \eta \quad (18)$$

where s is the time slot considered; \mathbf{p}_s^* and C_s are the vector of power levels and the matrix formed by the subnetwork for time slot s , respectively. However, once the scheduling problem is solved and the information required is sent to the communication pairs, the Foschini-Miljanic Power Control algorithm can be used to attain the required SINRs for all pairs in the subnetwork in a distributed manner, as detailed in [10].

For our computational study we generated a set of test networks of various sizes which will serve as the benchmark for evaluating our algorithm. More specifically, the benchmark set consists of networks with 10, 20, 30, ..., 100 pairs (we have 10 instances of each) and hence, in total, we have a test set consisting of 100 networks.

A. Performance evaluation of the lower bound

Table II shows the average computational time (in CPU seconds) required for finding UB , $LB1$ and $LB2$, as described in Sections VII-A and VII-B, the percentage of instances for which either $LB1$ or $LB2$ is equal to UB , hence proving optimality, as well as the relative deviation of $LB2$, compared to $LB1$.

No. of pairs	CPU time (sec)			Optimality proven		% $\frac{LB2-LB1}{LB1}$
	UB	$LB1$	$LB2$	$LB1$	$LB2$	
10	0.02	0.00	0.08	40%	70%	25%
20	0.06	0.00	0.08	0%	10%	50%
30	0.10	0.00	0.10	20%	60%	57%
40	0.13	0.00	0.09	0%	20%	55%
50	0.17	0.00	0.10	0%	30%	70%
60	0.22	0.00	0.10	10%	30%	48%
70	0.28	0.00	0.12	0%	10%	57%
80	0.39	0.00	0.15	0%	10%	72%
90	0.40	0.00	0.16	0%	0%	52%
100	0.44	0.00	0.15	0%	0%	60%

TABLE II
NUMERICAL RESULTS FOR THE LOWER BOUND FOR MODEL 1

The results in Table II show that all the described bounding techniques are very fast and provide upper bounds within 0.02 CPU seconds on average for the smallest sized-networks (10 pairs) and up to 0.44 CPU seconds on average for the largest sized-networks (100 pairs). On average, $LB2$ is able to prove optimality for a larger number of test instances, compared to $LB1$. In fact, the average relative deviation of $LB2$, as compared with $LB1$, is at least over 50% for most test problems, with the exception of the 10- and 60-pair instances. It is worth mentioning that although this improvement comes at an increased computational cost, the required computational times are still within reasonable limits.

B. Computational evaluation of the B&B approach

Tables III, IV and V show the average computational time (in CPU seconds) required for finding the optimal solution to Model 1 using BB_{gen} , BB_{slots}^1 and BB_{slots}^2 , respectively, within the set time limit. We chose 10000 CPU seconds as a time limit. Tables III-V also show the number of instances whose solution time exceeded the time limit (#out of time) and the number of instances which could not be solved due to memory limitations (#out of mem).

No. of pairs	BB_{gen}		
	CPU time (sec)	#out of time	#out of mem
10	0.40	0	0
20	442.21	1	0
30	1592.84	1	1
40	1030.10	4	4

TABLE III
NUMERICAL RESULTS FOR FINDING THE OPTIMAL SOLUTION TO MODEL 1 USING BB_{gen}

No. of pairs	BB_{slots}^1		
	CPU time (sec)	#out of time	#out of mem
10	0.15	0	0
20	3.93	0	0
30	246.65	0	0
40	715.13	0	0
50	465.37	2	1
60	3231.10	7	1

TABLE IV
NUMERICAL RESULTS FOR FINDING THE OPTIMAL SOLUTION TO MODEL 1 USING BB_{slots}^1

No. of pairs	BB_{slots}^2		
	CPU time (sec)	#out of time	#out of mem
10	0.23	0	0
20	2.04	0	0
30	152.32	0	0
40	56.40	0	0
50	507.81	1	0
60	2711.01	5	0

TABLE V
NUMERICAL RESULTS FOR FINDING THE OPTIMAL SOLUTION TO MODEL 1 USING BB_{slots}^2

The results in Table III show that for networks including more than 30 pairs, the model becomes prohibitively large for BB_{gen} to solve within reasonable computational time. In fact, the optimal solution to 80% of the instances with 40 pairs could not be found either because the time limit of 10000 CPU seconds was exceeded or no more memory was available. More promising results can be found in Tables IV and V for the customized versions of the B&B algorithm, BB_{slots}^1 and BB_{slots}^2 . According to both tables, the limitations of BB_{slots}^1 and BB_{slots}^2 were only observed for instances with 60 or more pairs. Apart from being capable of solving larger sized instances, both approaches were able to do so with lower computational times. The importance of the quality of the lower bound is especially evident when comparing the results in Tables IV and V. BB_{slots}^2 is capable of solving 90% of the test instances with 50-pairs and 50% of the 60-pair instances, whereas only 70% and 20% were solved with BB_{slots}^1 , respectively. This is expected due to the fact that $LB2$ is a stronger lower bound than $LB1$.

C. Computational evaluation of the cutting plane approach

Table VI shows the average computational time (in CPU seconds) required to solve the network instances within the set time limit using both CP_{cuts} and CP_{slots} .

No. of pairs	CPU time (sec)		#out of time	
	CP_{cuts}	CP_{slots}	CP_{cuts}	CP_{slots}
10	0.69	0.59	0	0
20	44.70	13.96	0	0
30	1840.18	587.23	3	2
40	N/A	548.20	10	4

TABLE VI
NUMERICAL RESULTS FOR FINDING THE OPTIMAL SOLUTION TO MODEL 1 USING CP_{cuts} AND CP_{slots}

It is evident from the results in Table VI that CP_{slots} outperforms CP_{cuts} in all sets of test problems; this is especially evident for the largest-sized instances in the table (40-pairs) where CP_{slots} is capable of solving 60% of the instances while none of them could be solved by CP_{cuts} in the set time limit.

While CP_{slots} outperforms BB_{gen} , its computational performance is not as powerful as the performance of BB_{slots}^2 . However, it is worth mentioning that CP_{slots} provides solutions with a smaller total power, on average, than the one provided by BB_{slots}^2 . This is a natural consequence of the objective value (16a) in Model 16. Table VIII-C shows the average percentage deviation in the sum of the powers found by the solutions provided by BB_{slots} ($power_{bb}$) as compared to the solution found by CP_{slots} ($power_{cp}$).

No. of pairs	$\% \frac{power_{bb} - power_{cp}}{power_{cp}}$
10	53%
20	3%
30	1646%
40	2409%

TABLE VII

NUMERICAL RESULTS OF TOTAL POWER REQUIRED BY THE SOLUTION FOUND BY BB_{slots}^2 AS COMPARED TO THE SOLUTION FOUND BY CP_{slots}

According to Table VIII-C, the solution obtained by BB_{slots}^2 is associated to a total power which is on average significantly larger than the one provided by CP_{slots} . It is worth noting though, that CP_{slots} is able to minimize the total power of the given solution found by solving the Master Problem (Model 5), but not provide the minimum total power over all optimal solutions to Model 1. In fact, it is possible that the solution provided by BB_{slots}^2 (even though not superior in terms of the number of time slots) could provide a smaller total power than the solution found by CP_{slots} .

IX. CONCLUSIONS

We firstly proved that the transmission scheduling problem in wireless networks for the physical model with power control is NP-hard. Since we established that the problem is unlikely to admit a polynomial-time optimal solution, the attention should now be turned towards approximation algorithms. Secondly, we developed algorithms for the abstract physical model using B&B and CP approaches that are further enhanced by techniques that find good upper bounds and lower bounds. Both the analytic and heuristic approaches are useful in deriving the optimal solution value quickly, as well as providing feasible solutions of known quality in case the optimal solution is unknown. The significance of these results is twofold. On one hand, we have proved that the well known wireless scheduling problem, considered as an open problem so far, is NP-hard and therefore no analytical method can find the optimal solution in polynomial time in the worst case. On the other hand, the problem of transmission scheduling, with transmitters being able to adjust their power levels to fully benefit from spatial reuse, has been formulated and solved. The results are of practical importance when a central

controller exists that is able to make the calculations and disseminate the information to the rest of the network (for example in cellular networks where the base station can act as a centralized agent). In addition, the solution constitutes an important benchmark when evaluating approximation algorithms or distributed algorithms for scheduling without knowledge of the whole network.

REFERENCES

- [1] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46:388–404, March 2000.
- [2] B. Krishnamachari. *Networking Wireless Sensors*. Cambridge University Press, New York, USA, 2005.
- [3] T. Moscibroda, R. Wattenhofer, and Y. Weber. Protocol Design Beyond Graph-based Models. In *ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets)*, 2006.
- [4] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer. Complexity in geometric SINR. In *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 100–109, New York, NY, USA, 2007.
- [5] Thomas Locher, Pascal von Rickenbach, and Roger Wattenhofer. Sensor networks continue to puzzle: Selected open problems. In *9th International Conference on Distributed Computing and Networking (ICDCN)*, Kolkata, India, January 2008.
- [6] B. Hajek and G. Sasaki. Link scheduling in polynomial time. *IEEE Transactions on Information Theory*, 34(5):910–917, September 1988.
- [7] V.S.A. Kumar, M.V. Marathe, S. Parthasarathy, and A. Srinivasan. End-to-end packet-scheduling in wireless ad-hoc networks. In *Symposium on Discrete Algorithms (SODA)*, pages 1021 – 1030. ACM-SIAM, 2004.
- [8] T. Moscibroda and R. Wattenhofer. Coloring unstructured radio networks. In *Symposium on Parallel Networks and Architectures (SPAA)*, pages 39 – 48, 2005.
- [9] J. Zander. Distributed co-channel interference control in cellular radio systems. *IEEE Transaction on Vehicular Technology*, 41(3):305–311, August 1992.
- [10] G. Foschini and Z. Miljanic. A Simple Distributed Autonomous Power Control Algorithm and its Convergence. *IEEE Transactions on Vehicular Technology*, 42(4):641–646, November 1993.
- [11] S. Grandhi, J. Zander, and R. Yates. Constrained power control. *Wireless Personal Communications*, 2(3), August 1995.
- [12] R. D. Yates. A framework for uplink power control in cellular radio systems. *IEEE Journal on Selected Areas in Communications*, 13:1341–1347, September 1995.
- [13] F. Gunnarsson and F. Gunnarsson. Power control in cellular radio systems: Analysis, design and estimation, 2000.
- [14] Z. Gajic, D. Skataric, and S. Koskie. Optimal SIR-based Power Updates in Wireless CDMA Communication Systems. In *IEEE Conference on Decision and Control*, volume 5, pages 5146–5151, December 2004.
- [15] E.-S. Jung and N. H. Vaidya. A power control MAC protocol for ad hoc networks. In *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 36–47, New York, NY, USA, 2002. ACM.
- [16] T. ElBatt and A. Ephremides. Joint Scheduling and Power Control for Wireless Ad-hoc Networks. In *Proceedings of IEEE INFOCOM*, 2002.
- [17] J. Tang, G. Xue, C. Chandler, and W. Zhang. Link scheduling with power control for throughput enhancement in multihop wireless networks. In *Proceedings of the Second International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks*, page 1, Washington, DC, USA, 2005. IEEE Computer Society.
- [18] S. Kompella, J. E. Wieselthier, and A. Ephremides. A Cross-layer Approach to Optimal Wireless Link Scheduling with SINR Constraints. In *MILCOM*, 2007.
- [19] Y. Li and A. Ephremides. A joint scheduling, power control, and routing algorithm for ad hoc wireless networks. *Ad Hoc Netw.*, 5(7):959–973, 2007.
- [20] A. Behzad and I. Rubin. Optimum Integrated Link Scheduling and Power Control for Multihop Wireless Networks. *Vehicular Technology*, 56(1):194–205, January 2007.
- [21] A. Muqattash and M. Krunz. A single-channel solution for transmission power control in wireless ad hoc networks. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 210–221, New York, NY, USA, 2004. ACM.

- [22] T. Moscibroda and R. Wattenhofer. The Complexity of Connectivity in Wireless Networks. In *IEEE INFOCOM*, 2006.
- [23] O. Goussevskaia, M. Halldorsson, R. Wattenhofer, and E. Welzl. Capacity of Arbitrary Wireless Networks. In *28th Annual IEEE Conference on Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, April 2009.
- [24] M. M. Halldórsson and R. Wattenhofer. Wireless communication is in APX. In *ICALP '09: Proceedings of the 36th International Colloquium on Automata, Languages and Programming*, pages 525–536, Berlin, Heidelberg, 2009. Springer-Verlag.
- [25] A. Fanghänel, T. Kesselheim, H. Räcke, and B. Vöcking. Oblivious interference scheduling. In *PODC '09: Proceedings of the 28th ACM symposium on Principles of distributed computing*, pages 220–229, New York, NY, USA, 2009. ACM.
- [26] J. Grönkvist. Assignment methods for spatial reuse TDMA. In *MobiHoc '00: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pages 119–124, Piscataway, NJ, USA, 2000. IEEE Press.
- [27] P. Bjorklund, P. Varbrand, and Di Yuan. Resource Optimization of Spatial TDMA in Ad Hoc Radio Networks: A Column Generation Approach. In *IEEE INFOCOM*, pages 818–824, March 2003.
- [28] E. Klerides and T. Charalambous. Transmission scheduling in wireless networks with SINR constraints. In *International Conference on Networking and Services (ICNS)*, April 2009.
- [29] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
- [30] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1994.
- [31] S.U. Pillai, T. Suel, and Seunghun C. The Perron-Frobenius theorem: some of its applications. *Signal Processing Magazine, IEEE*, 22:62–75, March 2005.
- [32] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [33] P. Kall and S. W. Wallace. *Stochastic Programming*. Wiley, Chichester, 1994.
- [34] R. S. Garfinkel and G. L. Nemhauser. *Integer Programming*. Wiley, Chichester, 1972.