Energy Efficient Scheduling with Power Control for Wireless Networks

Bastian Katz
Institute for Theoretical Informatics
Karlsruhe Institute of Technology
Karlsruhe, Germany
Email: bastian.katz@kit.edu

Markus Völker
Institute for Theoretical Informatics
Karlsruhe Institute of Technology
Karlsruhe, Germany
Email: markus.voelker@kit.edu

Dorothea Wagner
Institute for Theoretical Informatics
Karlsruhe Institute of Technology
Karlsruhe, Germany
Email: dorothea.wagner@kit.edu

***Abstract*—Scheduling of transmissions is one of the most fundamental problems in the context of wireless networks. In this article, we consider the problem of computing power efficient schedules with high throughput. We answer the open question concerning the complexity of scheduling with power control in the SINR_G model of interference. Based on a novel scheme for dynamic computation of optimum transmission powers in feasible schedules, we introduce a new and efficient heuristic for finding good schedules along the tradeoff between throughput and energy efficiency in the physical SINR model. Since our algorithms do not rely on simplistic assumptions about path loss, they are suited for realistic scenarios with attenuation and shadowing effects. We compare our approach to a broad selection of state-of-the-art approaches in indoor and outdoor scenarios. In all situations, our approach outperforms the existing approaches with respect to schedule length and power consumption, i. e., yields pareto-superior schedules including schedules that significantly improve the throughput.**

I. Introduction

The scheduling problem in wireless networks has received a lot of attention in recent years. Given a set of transmission requests, one has to find a schedule that satisfies all interference constraints. If wireless nodes can individually adjust transmission powers on a per-transmission base, a schedule must also include power levels for each transmission, adding a powerful degree of freedom. The quality of a schedule then substantially determines throughput *and* power efficiency of the wireless communication. Obviously, distribution of transmissions to a small number of time slots is desirable to decrease the time to complete the request, i. e., to improve the throughput. On the other hand, less interference between concurrent transmissions allows for lower transmission powers and can hence reduce energy consumption. Especially in wireless sensor networks, where energy is a limited and valuable resource, the computation of good time-division multiple-access (TDMA) schedules can help to extend the lifetime of the whole network. Since schedules admissible without power control are also solutions if power control is possible, power control can be exploited to reduce the energy consumption of such schedules in a subsequent and largely orthogonal step. However, solving both problems simultaneously helps to compute schedules with higher throughput (i. e. shorter span) and less energy consumption: power control admits schedules that would otherwise violate interference constraints, since reducing transmission powers can also reduce interference between concurrent transmissions.

Considering the importance of the scheduling problem, it is not surprising that it has been studied intensely. Originally, most algorithmic approaches to scheduling problems were graph-based. This made it possible to reduce the problem to variants of independent set, matching, or coloring problems. Some examples are [1]–[9]. Graph-based methods were especially popular in theoretical computer science, as they allow a thorough theoretical analysis. Unfortunately, graph-based models are too simplistic as they do not model distance-dependance of link quality and accumulation of interference [10]–[12]. In contrast, researchers in information, communication, or network theory usually apply the physical signal-to-interference-plus-noise-ratio (SINR), which reflects the physical reality more precisely. In this model, a transmission is successful if and only if the ratio of the useful signal to interference plus background noise exceeds some threshold. Signal strength and interference depend on path loss between sender and receiver. This path loss is determined by the distance between sender and receiver, as well as by environmental conditions such as obstacles. One simplification of this model is the geometric SINR model (SINR_G model), where one assumes that path loss is fully determined by the sender-receiver distance. Here, path loss is modeled by a power law and different environments by different path loss exponents. Scheduling in the general SINR model is NP-hard [13]. The NP-hardness of scheduling with fixed powers in the more restricted geometric SINR_G model has been shown in [13]. Power control has originally been studied in the context of channelized cellular systems [14], [15] and code-division multiple-access (CDMA) systems [16]. The joint problem of scheduling and power control was first addressed by ElBatt and Ephremides [17], [18], followed by others including [19]–[22].

The complexity of scheduling with power control in the SINR_G model is considered an important open problem [13], [23]. In this work, we answer this question by proving NP-hardness.

More recently, several approximation algorithms have been proposed for scheduling with and without power control [13], [24]–[26] in this simplified model of interference.

Today, there exists a wide range of algorithms for scheduling in the SINR model, mostly aiming solely at maximizing the throughput, i. e., at minimizing the span of computed schedules. An algorithm also taking energy efficiency into account is, e. g., given by Lu and Krishnamachari [27]. The authors also consider the problem to minimize the energy that is needed to schedule a given set of transmissions within a predefined number of time slots. We discuss some relevant algorithms more detailed in Section VI. The algorithms we contribute also allow both, maximizing the throughput and minimizing the energy consumption for a given throughput. They are based on a novel dynamic scheme for maintaining optimum powers for a changing feasible set of transmissions. The proposed data structure supports more sophisticated scheduling decisions without a penalty on the worst-case running time. To evaluate our algorithms, we compare them experimentally to several other state-of-the-art algorithms. In our simulations, we consider three scenarios: randomly distributed sender-receiver pairs, a given network topology, and a scenario that models effects that occur within buildings. In all three scenarios, our algorithms outperform the other approaches in two ways: First, they yield higher throughput. Second, our algorithms can realize any other algorithm's throughput with significantly less energy. Moreover, the algorithms can be used to find a good compromise between throughput and energy consumption.

The rest of the paper is organized as follows. The used models and notations are described in Section II. Subsequently, in Section III, we prove the NP-hardness of scheduling with power control in the SINR_G model. Our approach to the problem of power control is described in Section IV. Based on the new power control algorithm, we describe our scheduling heuristics in Section V. The simulation results are shown in Section VI. Section VII concludes this paper.

II. Models and Notations

We assume a network of wireless nodes. In this network, we are given a set $\{l_1, l_2, \dots, l_n\}$ of n links. Every *link* $l_i = (s_i, r_i)$ is defined by its *sender* s_i and its *receiver* r_i. We say that a link l_i is *active* if s_i transmits data to r_i. The *transmission power* that s_i uses is denoted by P_i. The *path loss* between sender s_i and receiver r_j is denoted by γ_{ij}. It defines how much the signal strength decreases on the way from sender s_i to receiver r_j and can depend on the distance between sender and receiver, as well as on environmental conditions and obstacles. When a sender s_j sends with power P_j then receiver r_i receives the signal with power $P_j\gamma_{ji}$. In the case of $i = j$ the signal a *useful signal*, otherwise we call it *interference*. Additionally to the interference due to concurrent transmissions, every receiver r_i experiences some background noise η_i. A *transmission* $t_i = (s_i, r_i, P_i)$ is defined by sender-receiver-pair (s_i, r_i) and transmission power P_i that s_i uses.

In this paper, we use the physically motivated *SINR model* (signal-to-interference-plus-noise-ratio model). In the SINR model, a transmission t_i is assumed to be *feasible* if the ratio of the useful signal to accumulated interferences plus background noise at receiver r_i exceeds some *minimum SINR* β_i. This SINR condition is expressed by the following equation:

$$\frac{P_i\gamma_{ii}}{\sum_{j\neq i} P_j\gamma_{ji} + \eta_i} \geq \beta_i \qquad (1)$$

A set of transmissions $\{t_1, t_2, \dots, t_k\}$ is said to be feasible if SINR condition (1) of every receiver r_i, $1 \leq i \leq k$, is fulfilled when all senders s_i, $1 \leq i \leq k$, transmit concurrently with their associated transmission power P_i.

Only in Section III we deal with the more restricted geometric SINR model (SINR_G model). In the SINR_G model, a close relationship between distance and path loss is assumed. Let $d(s_i, r_j)$ denote the distance between nodes s_i and r_j. The path loss between s_i and r_j is then given by $\gamma_{ij} = d(s_i, r_j)^{-\alpha}$, with path loss exponent α. The path loss exponent α depends on the environment and defines how fast the signal decays with distance. Usually, α is assumed to be about 2 in free space and between 3 and 5 in buildings.

In the *scheduling problem*, one is given a set $\mathcal{L}$ of links and transmission powers P_i to be used by the senders. Usually, only the case of *uniform* transmission powers $P_i \equiv P$ is considered. The problem is to find a partition of $\mathcal{L}$ into transmission sets such that all sets are feasible. The single transmission sets are called *slots*, and we refer to the number of slots in a schedule as the schedule's *span*. The goal is to find a schedule with minimum span in order to maximize the communication throughput.

A somehow dual problem to the scheduling problem is the *power control problem*: given a set of links $\mathcal{L}$ and a partition into transmission sets, find transmission powers $P_i \in (0, P_{\max}]$ such that each transmission set is feasible. In order to extend the lifetime of the network, the optimization goal is to find *minimum* such transmission powers. As we will see in Section IV, the notion of minimum transmission powers is well-defined and independent of the metric applied.

A generalization of both problems, the scheduling and the power control problem, is the problem of *scheduling with power control*, in which one has to compute a partition of a set of links and proper transmission powers. Again, the computation of a feasible schedule with minimum span is an important problem, but one can also be interested in minimizing the energy consumption for a given span. Note that pure energy minimization trivially leads to solutions where time slot contains exactly one single link. This paper addresses both relevant problems, pure throughput maximization and the bicriterial problem. Particularly, the algorithms proposed in Section V can be used to find a good compromise between power consumption and throughput.

III. Complexity of Scheduling with Power Control

In the following, we sketch an NP-hardness proof for minimizing a schedule's span with power control in the SINR_G model. The proof is an extension of the NP-hardness proof for scheduling without power control given in [13]. Due to space limitations, we restrict ourselves to the main

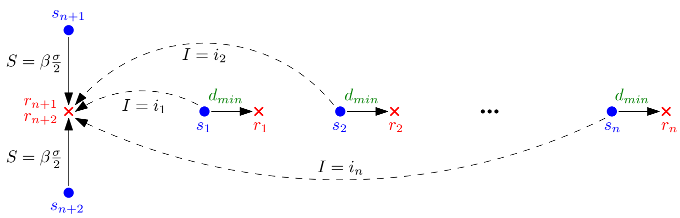


Fig. 1. Polynomial-time reduction of PARTITION to SCHEDPC

ideas. The technical details of the proof can be found in [28]. Our proof requires that we know minimum and maximum transmission powers $P_{\min} > 0$ and $P_{\max} < \infty$. This is no real restriction, as the hardware usually dictates an upper bound on the transmission power and due to the background noise there exists a power beneath which no transmission can be successful. As $P_{\min}$ and $P_{\max}$ are arbitrary, all practically relevant cases are covered. In particular, this proof contains the situation where every sender has a finite set of available transmission powers from which it can choose. For the sake of simplicity, we ignore the influence of background noise in this section. This makes the equations clearer but has no significant effect on the results.

In order to show the NP-hardness of scheduling with power control (SCHEDPC), we will give a polynomial time reduction of the well-known PARTITION problem to SCHEDPC. The PARTITION problem has been shown to be NP-complete in [29]. It is defined as follows: Given a set $\mathcal{I} = \{i_1, \ldots, i_n\}$ of integers, find $\mathcal{I}_1, \mathcal{I}_2 \subset \mathcal{I}$ such that $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$ and

$$\sum_{i_j \in \mathcal{I}_1} i_j = \sum_{i_j \in \mathcal{I}_2} i_j = \frac{1}{2} \sum_{i_j \in \mathcal{I}} i_j \ .$$

Let $\mathcal{I} = \{i_1, \ldots, i_n\}$ be an instance of PARTITION. Without loss of generality, we assume that all elements are distinct and positive and we define $\sigma := \sum_{j=1}^{n} i_j$. In order to solve the PARTITION problem for $\mathcal{I}$, we construct an instance $L_{\mathcal{I}} = \{l_1, \ldots, l_{n+2}\}$ of SCHEDPC with $n+2$ links such that there exists a schedule of length 2 if and only if the PARTITION instance $\mathcal{I}$ has a solution. The construction is depicted in Figure 1: For every integer $i_j \in \mathcal{I}$ we introduce a link $l_j = (s_j, r_j)$. Every sender s_j is placed at position $pos(s_j) = ((P_{\min}/i_j)^{1/\alpha}, 0)$. The position is chosen such that the interference caused at the origin $(0,0)$ of the coordinate system equals i_j when s_j sends with power $P_{\min}$. Next, we place the corresponding receivers such that every transmission l_j can be executed successfully, even if s_j sends with power $P_{\min}$ and every other sender sends with power $P_{\max}$. For this, every sender-receiver-pair has to be placed sufficiently close together. One can show that distance

$$d_{\min} = \frac{P_{\min}^{1/\alpha} \cdot \left(\frac{1}{(i_{\max}-1)^{1/\alpha}} - \frac{1}{i_{\max}^{1/\alpha}} \right)}{\left(1 + (\frac{P_{\max}}{P_{\min}} n\beta)^{\frac{1}{\alpha}} \right)},$$

where $i_{\max}$ is the maximum value in $\mathcal{I}$, is sufficient. Thus, we place every receiver $r_i, 1 \le i \le n$, at position $pos(r_i) = pos(s_i) + (d_{\min}, 0)$.

Finally, we have to place l_{n+1} and l_{n+2}. We positioned the senders $s_1, \ldots, s_n$ such that the interference which they cause at the origin is proportional to $i_1, \ldots, i_n$. In order to take advantage of this property, we place r_{n+1} and r_{n+2} at the origin, $pos(r_{n+1}) = pos(r_{n+2}) = (0,0)$. Last, we place their senders s_{n+1} and s_{n+2} perpendicular to the x-axis at distance $(2P_{\max}/\beta\sigma)^{1/\alpha}$, i.e., $pos(s_{n+1}) = \left(0, \left(\frac{2P_{\max}}{\beta \cdot \sigma}\right)^{1/\alpha}\right)$ and $pos(s_{n+2}) = \left(0, -\left(\frac{2P_{\max}}{\beta \cdot \sigma}\right)^{1/\alpha}\right)$.

As the receivers r_{n+1} and r_{n+2} share the same position, transmissions t_{n+1} and t_{n+2} cannot be active concurrently. Thus, one needs at least two slots in order to schedule all transmissions. The senders s_j, $1 \le j \le n$, have to send at least with power $P_{\min}$ while they are active. Thus, together they produce at least the interference $\sigma = \sum_{j=1}^{n} i_j$ at the position of r_{n+1} and r_{n+2}. The senders s_{n+1} and s_{n+2}, at the same time, cannot transmit with more power than $P_{\max}$. Thus, they cannot tolerate more interference than $\frac{\sigma}{2}$. It follows that a schedule with only two slots is possible if and only if s_{n+1} and s_{n+2} send with $P_{\max}$, all other senders send with $P_{\min}$, and the senders s_j, $1 \le j \le n$, can be partitioned into two sets $\mathcal{I}_1$ and $\mathcal{I}_2$ such that $\sum_{i_j \in \mathcal{I}_1} i_j = \sum_{i_j \in \mathcal{I}_2} i_j = \frac{\sigma}{2}$. This, in turn, implies a solution to the original PARTITION problem.

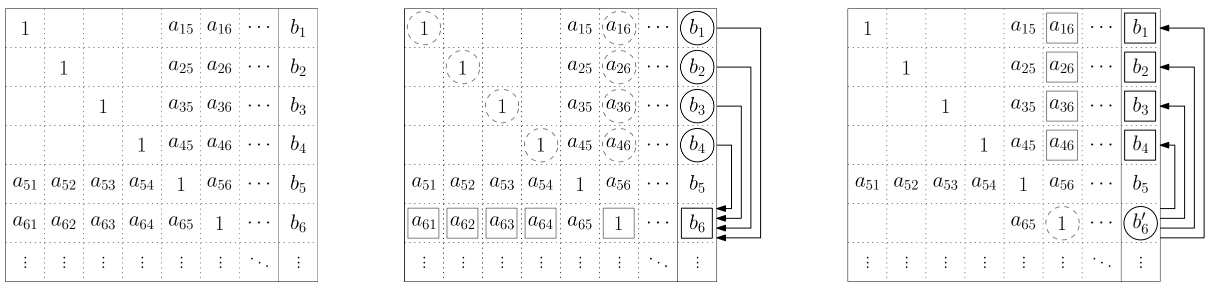
IV. COMPUTATION OF OPTIMUM TRANSMISSION POWERS

Solving the power control problem, i. e., finding minimum transmission powers for a given schedule, is a common subproblem of many algorithms for scheduling problems with power control. Since minimizing transmission powers for a given schedule trivially reduces to minimizing transmission powers for each slot separately, we restate the power control problem in a simplified form: Given a set $\mathcal{L} = \{l_1, \ldots, l_n\}$ of links, find the minimum transmission powers $P_1, \ldots, P_n$, $P_i \in (0, P_{\max}]$ (possibly with $P_{\max} = \infty$), such that the resulting transmission set is feasible, or decide that no such powers exist.

As this problem is very fundamental, several approaches have been proposed in the literature. For example, one can express the problem as a linear program and use an LP solver to compute optimum powers, e. g., as

$$\begin{aligned} \text{min!} \quad & \sum_{1 \le i \le n} P_i \\ \text{s. t.} \quad & P_i \ge \sum_{j \ne i} P_j \frac{\beta_i \gamma_{ji}}{\gamma_{ii}} + \eta_i \frac{\beta_i}{\gamma_{ii}} \quad \text{for } 1 \le i \le n \\ & P_i \ge 0 \quad \text{and} \quad P_i \le P_{\max} \quad \text{for } 1 \le i \le n \ . \end{aligned} \tag{2}$$

In this form, we also observe that minimizing the sum of powers yields the same result as minimizing any function $f(P_1, \ldots, P_n)$ that is (not necessarily strongly) monotonically increasing in the P_i (such functions include the maximum, the minimum, or any norm). Explicitly, if the problem is feasible, for any such function f, the optimum solution $p^\star := (P_1^\star, \ldots, P_n^\star)^T$ is given by setting $P_i^\star$ to the minimum value



(a) (b) (c)

Fig. 2. Matrix $[A|b]$ of the system of linear equations. (a) Partial solution for $S = \{l_1, l_2, l_3, l_4\}$. (b) Computation of optimum power for link l_6 in $S \cup \{l_6\}$. (c) Computation of updated powers for already active links.

of P_i in any feasible solution $p = (P_1, \ldots, P_n)^T$. This is well-defined since the value of any P_i in any feasible solution is bounded from below and the set of feasible solutions is closed. Assume $P^\star$ is not feasible (if it is, it is obviously optimal), i. e., assume that Equation (2) is violated for some i. By construction, there is a feasible solution $p^i := (P_1^i, \ldots, P_n^i)^T$ with $P_i^\star := P_i^i$. For all $j \neq i$, we have $P_j^\star \leq P_j^i$ by construction and hence Equation (2) holds for i, in contradiction with the assumption.

An approach used by most heuristics for the combined problem of scheduling and power control is to start with small transmission powers and then iteratively increase the powers until the transmission set becomes feasible or until one of the senders exceeds its maximum transmission power. This can be done in such a way that the resulting transmission powers converge to the optimum [17]. It has also been shown that under certain conditions only a constant number c of iterations is necessary to either get near-optimal transmission powers or to determine that no proper transmission powers exist. Every iteration takes $O(n^2)$ time, so a near-optimal power assignment can be computed in $O(cn^2)$ time.

If one has to compute such a power assignment only once for a given set of links, one can hardly do better. However, in the combined problem of scheduling and power control, such power assignments have to be computed over and over until the final set of links and powers is found. The usual approach is to start with an empty set and to add links one by one in some order that aims to find a good schedule. This gives an $O(cn^3)$ computation time for filling a time slot with n transmissions.

In this work, we go a different way by stating the problem as a system of linear equations and then solving the system stepwise in an advantageous way. This will allow us to use more powerful scheduling heuristics without a penalty on the worst-case running time. As we will see in the experimental section, this intertwining of scheduling and power control allows the computation of substantially better schedules.

We start with the observation that for the optimal solution $p^\star$, Equation (2) is tight, since otherwise, some $P_i^\star$ could be reduced without losing feasibility. That is, finding powers minimizing any monotonically increasing function f subject to $0 \leq P_i \leq P_{\max}$ and

$$P_i = \sum_{j \neq i} P_j \frac{\beta_i \gamma_{ji}}{\gamma_{ii}} + \eta_i \frac{\beta_i}{\gamma_{ii}} \quad, 1 \leq i \leq n \tag{3}$$

is equivalent to the problem stated above.

Writing Equation (3) as

$$A \cdot p = b \tag{4}$$

with

$$A = \begin{pmatrix} 1 & -\frac{\beta_1 \gamma_{21}}{\gamma_{11}} & \ldots & -\frac{\beta_1 \gamma_{n1}}{\gamma_{11}} \\ -\frac{\beta_2 \gamma_{12}}{\gamma_{22}} & 1 & \ldots & -\frac{\beta_2 \gamma_{n2}}{\gamma_{22}} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\beta_n \gamma_{1n}}{\gamma_{nn}} & -\frac{\beta_n \gamma_{2n}}{\gamma_{nn}} & \ldots & 1 \end{pmatrix},$$

$$p = (P_1, \ldots, P_n)^T, \quad b = (\beta_1 \eta_1/\gamma_1 1, \ldots, \beta_n \eta_n/\gamma_n n)^T \quad ,$$

we obtain the following result, which justifies to drop the optimization criterion:

Proposition 1: If there is any feasible solution to the power control problem, Equation (4) has a unique solution, which is the optimum $p^\star$.

Proof: Assume that the power control problem has a feasible solution. Then $p^\star$ is an optimum solution to the power control problem and also a solution to Equation (4). Assume another solution p' to Equation (4). Then A is not regular and $A \cdot \bar{p} = 0$ for $\bar{p} = p^\star - p'$. Then $A \cdot (p^\star - \lambda \bar{p}) = b$ for all $\lambda \in \mathbb{R}$, and taking any non-zero entry $\bar{P}_i$ and a sufficiently small $\epsilon \in \mathbb{R}^+$, we get a feasible solution $p^s := p^\star - (\epsilon/\bar{P}_i)\bar{p}$ with $P_i^s = P_i^\star - \epsilon$ and $P_j^s \geq 0$ for all $1 \leq i \leq n$, which contradicts with the optimality of $p^\star$. ■

Hence, from now on, for a given set of links, we are only interested in finding any solution $p = (P_1, \ldots, P_n)^T$ to Equation (4). If all P_i are nonnegative and smaller than $P_{\max}$, this solution is the optimum (and unique). If there is no solution or any solution with some $P_i < 0$ or $P_i > P_{\max}$—which is always the case if there is more than one solution—, we can conclude that the problem is infeasible.

In the remainder of this section, we will show how this system of equations can be used to efficiently determine transmission powers.

Dynamic Computation of Optimum Transmission Powers

We now assume that we are given a set T of links that we want to schedule by adding transmissions from T to initially empty, always disjoint sets of active and feasible transmissions. For such a set S, we transform the matrix $[A|b]$ along these additions to maintain a matrix $[A^S|b^S]$ with a special structure, but without altering the solution of the underlying equation system.

We start with $S = \emptyset$ and $A^\emptyset := A$ and $b^\emptyset := b$. During the execution of our algorithm, we maintain the invariant that after each step it holds that

$$a_{ii} = 1 \quad \forall l_i \in S \tag{5}$$
$$a_{ij} = 0 \quad \forall (l_i, l_j) \in S \times S, i \neq j \ . \tag{6}$$

This obviously holds for $S = \emptyset$. Figure 2a shows this for a small example where $S = \{l_1, l_2, l_3, l_4\}$. Let us assume that we want to add some link $l_i \in T$ to the set S, yielding $S' = S \cup \{l_i\}$. To preserve the special structure of our data structure, row i of matrix $[A^S|b^S]$ (the SINR constraint of link l_i) becomes important. For all $l_j \in S$, we have to subtract row j multiplied by a_{ij} from row i to get $a_{ij} = 0$ (Step A). Next, we divide row i by a_{ii} to get $a_{ii} = 1$ (Step B). Finally, for every l_j in S, we subtract row i multiplied by a_{ji} from row j, resulting in $a_{ji} = 0$ (Step C). This update can be done in $O(kn)$ time, where $k = |S|$ and $n = |T \cup S|$. As stated above, all described matrix operations do not alter the solution of the underlying equation system.

Let us now see why we chose this special structure for matrix $[A^S|b^S]$. First of all, if S is feasible, $P_i = b_i^S$ for all $l_i \in S$ and $P_j = 0$ for all $l_j \notin S$ immediately gives an optimum power assignment for link set S. More importantly, one can efficiently predict the effect that the addition of some arbitrary link $l_i \in T$ to S would have (cf. Figures 2b and 2c): Looking at $[A^{S'}|b^{S'}]$ for active transmissions $S' = S \cup \{l_i\}$, the optimum power P'_i of link l_i in S' is

$$P'_i = b_i^{S'} = \frac{b_i^S - \sum_{j \in S} a_{ij}^S b_j^S}{1 - \sum_{j \in S} a_{ij}^S a_{ji}^S} \ . \tag{7}$$

Thus, given $[A^S|b^S]$, we can compute P'_i in $O(k)$ time. In similar fashion, with b'_i known, for every link $l_j \in S$ we can compute the new optimum power P'_j in constant time as

$$P'_j = b_j^{S'} = b_j^S - a_{ji}^S b_i^{S'} \quad \forall j \in S \ . \tag{8}$$

This means, given matrix $[A^S|b^S]$ for some feasible link set S, we can compute optimum powers for link set $S \cup \{l_i\}$ in $O(k)$ time. This is a significant improvement over the naive approach that starts all over with the initial equation system.

So far we did not deal with the case that a set of links cannot be scheduled concurrently when trying to add a link l_i. Obviously, this can happen. With the considerations above, closer inspection of the equation system reveals that we can identify an event that indicates and proves infeasibility:

Proposition 2: Let S be a feasible subset of T with corresponding matrix $[A^S|b^S]$. The power control problem has a feasible solution for $S \cup l_i$ for some $l_i \in T \setminus S$ if and only if the matrix operations in Step A yield an entry $a_{ii}^S > 0$ and the operations in Step B and C yield only entries $b_j^S \leq P_{\max}$ for $l_j \in S \cup \{l_i\}$.

Proof: Making an induction over the cardinality of S, we can assume that during the computation of $[A^S|b^S]$ the case of a $a_{ii}^S \leq 0$ never occurred in step A (this trivially holds for $S = \emptyset$). Hence, during all preceding matrix operations it is an invariant that (a) entries $a_{ii} > 0$, (b) entries $b_i > 0$, (c) entries $a_{ij} \leq 0$ for $i \neq j$. This can be seen by induction over the matrix operations: the invariants hold for the initial matrix $[A^\emptyset|b^\emptyset]$, and if they hold, Step A effectively adds positive multiples of other rows to the changed row. When adding a multiple of row j to row i, the only possible violation of the invariants is exactly the case that a_{ii} becomes non-positive, which, by the outer induction, did not happen so far.

Then, in Step B, we multiply row i by a positive factor (not violating any invariant), and in Step C, we effectively add a positive multiple of row i to the other rows, which by construction cannot violate any invariant either.

If we perform Step A for a new link $l_i \in T \setminus S$, and we get $a_{ii} > 0$, then again all invariants hold during all update operations and the relevant blocks of the matrix $[A^{S\cup\{l_i\}}|b^{S\cup\{l_i\}}]$ encode a feasible (and hence optimal) solution for the links in $S \cup \{l_i\}$, given by the $P_i = b_i^{S\cup\{l_i\}}$, which is feasible if and only if all these powers additionally are below $P_{\max}$. If, on the other hand, in Step A a matrix entry $a_{ii} \leq 0$ occurs, then there cannot be a solution to the corresponding subproblem with all $P_i \geq 0$, since we still have $b_i > 0$, but all $a_{ij} \leq 0$. ■

In summary, given matrix $[A^S|b^S]$ of equation system (4) for a set S of active links ($|S| = k$), this approach allows to compute optimum powers for an arbitrary link set $S \cup \{l\}$ in $O(k)$ time and to compute the updated matrix $[A^{S\cup\{l\}}|b^{S\cup\{l\}}]$ in $O(kn)$ time. In the next section, we propose two heuristics that use this data structure to compute very good schedules efficiently.

V. Scheduling with Power Control

Two obvious optimization criteria for good schedules are the length of the schedule and the power that is needed to process all transmissions. At the first glance, those two objectives seem to contradict each other. In shorter schedules, the number of transmissions per slot is higher. This results in higher interference which in turn means that more power is needed. And surely, the most power efficient schedule would be the one where every transmission has its own slot. However, there is also some synergy. In order to compute short schedules, the interference between concurrent transmissions has to be kept small. And smaller interference also means less power consumption. Therefore, power minimization seems to be a good greedy strategy for both objectives. Based on this observation, we propose two greedy heuristics for the scheduling problem. The first one fills the slots one by one, iteratively adding the most power efficient transmission. The second one fills the slots in parallel and makes it possible to find a good compromise between schedule length and power consumption.

A. Filling Slots Sequentially

Many scheduling heuristics fill slots greedily one-by-one, i. e., a new slot is populated with links by adding links greedily. Usually, the order in which the links are processed is precomputed and depends on node degree, sender-receiver distance, or similar criteria. In contrast, we propose to select links depending on the effect that the decision has on the resulting transmission powers. For example, we pick the link that minimizes the maximum power that some transmission in the time slot requires. This way, links are picked which either can tolerate a lot of interference or fit well to the links that are already in the slot. Another good choice is to pick the link that minimizes the combined power used by all links in the slot, thus extending the overall lifetime of the whole network. We refer to these selection strategies as Greedy-Least-Maximum-Power (GLMP) and Greedy-Least-Additional-Power (GLAP), respectively.

Both approaches require to compute the new optimum transmission powers of the resulting link set $S \cup \{l\}$ for every link $l \in T$. Using the iterative power control algorithm that is used by most existing heuristics, this takes $O(ck^2)$ time for each of the $O(n)$ possible links, where $k = |S|$ is the number of links that are already assigned to the repective slot. At this point, our power control algorithm from Section IV comes into play. It allows us to compute updated transmission powers for $S \cup \{l\}$ in $O(k)$ time. This makes it possible to find the best link in $O(kn)$ time. As soon as we have determined the optimum link, we can add it in another $O(kn)$ time to the set of active links.

In summary, the sequential approach works as follows: Start with an empty set S of active links. While the set T of links is non-empty, find the link $l \in T$ that fits best to the set of active links. Depending on the objective, this can either be the link that minimizes the maximum transmission power (GLMP) or the one that minimizes the combined transmission power (GLAP). If such a link exists, add it to S and continue with the set $T' = T \setminus \{l\}$. If no more link fits to the set of active links, continue with a new time slot. This process is repeated until T is empty. The overall running time is in $O(k_{\max} n^2)$, where $k_{\max}$ is the maximum number of concurrently active transmissions. Note that the worst case occurs when $k_{\max} \in O(n)$. In this case, the running time is $O(n^3)$. This is quite good as the size of the input, given by the gain matrix, is already in $\Theta(n^2)$. And even if one had an optimum assignment of links to slots given as input, the computation of optimum powers using equation system (4) and common approaches such as Gaussian elimination would already require $O(n^3)$ time. Thus, the selection of good links does not affect the worst-case run time.

B. Filling Slots Simultaneously

As stated in the introduction, the schedule length is not the only optimization criterion. In this section, we deal with the problem of finding a good compromise between schedule length and power consumption. Given a set of links and a number of slots, we want to distribute the links to the available slots in a power-efficient manner.

For this purpose, we process the links in order of increasing sender-receiver-gain so that we deal with the most sensitive links first. For every link, we greedily determine the best slot, again measuring goodness by power consumption, and either choose the slot that minimizes the maximum occurring transmission power, or the slot that minimizes the additional power consumption. We refer to the former approach as Balanced-Least-Maximum-Power (BLMP) and to the latter as Balanced-Least-Additional-Power (BLAP). As before, this means that we have to precompute for every open link the powers that its addition would cause. Again, we use the method from Section IV to compute the new powers for a slot with k transmissions in $O(k)$ time. Fortunately, it is not necessary to maintain the complete matrix $[A|b]$ for every single slot. Instead, it is sufficient to maintain only the rows that correspond to SINR constraints of links that are assigned to the slot. Thus, for a slot with k active links we only need $O(kn)$ matrix cells, giving the algorithm $O(n^2)$ space complexity. This is optimal as the gain matrix in the input already needs $\Theta(n^2)$ space.

In comparison to the sequential approach, the parallel approach aims at significantly more balanced schedules, since all slots are expected to have a similar number of active links.

With a fixed number of slots, it may happen that a link does not fit in any of the available slots. In this case, we assume that a new slot is added to the set of available slots.

When the desired number of slots is not clear from the application, one can for example start with a single slot, which still yields more balanced schedules, since to links with high gain, which are scheduled latest, all opened slots are available.

However, if there is a need to create more than the initial number of slots, balancing is imperfect, since links scheduled early do not have all slots of the final schedule at choice. Hence, when interested in a balanced schedule with low span, it is best to find the minimum number of slots that is sufficient for this approach, e. g., by binary search. A reasonable result with only a single restart can also be produced taking the schedule returned for a single initial slot and take its span to make a good guess for a restart. For example, 80% of the returned schedules span as initial number of slots should already be a good choice.

VI. Simulations

A. Scenarios and Model Parameters

For our experiments, we used three different scenarios. The first one assumes that the sender-receiver pairs are randomly distributed in the Euclidean plane. There is no connected network structure. The signal strength is computed according to the SINR_G model with path-loss exponent $\alpha = 3$. This scenario resembles a sensor network with high number of nodes where only some of the nodes want to send concurrently. This kind of scenario was for example used in [25]. An example is shown in Figure 3a.

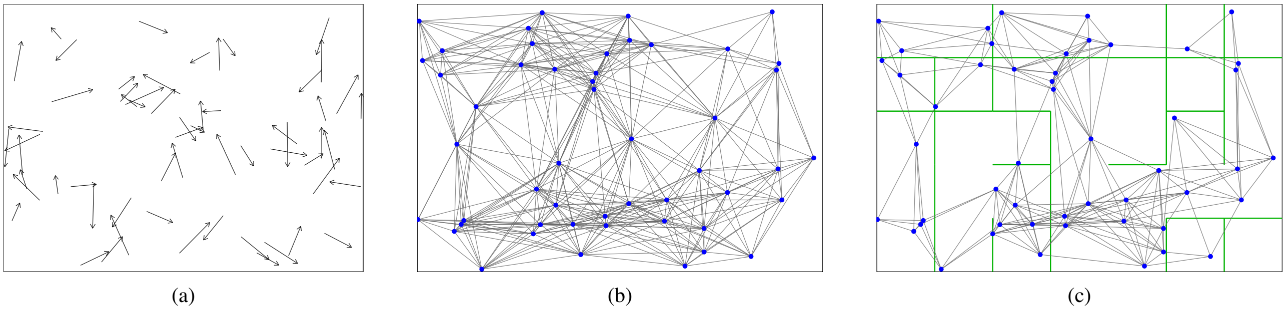


Fig. 3. Simulation scenarios. (a) Random links in free space. (b) Network in free space. (c) Network in building with wall attenuation and random effects.

The second scenario is based on a network topology. The nodes are placed randomly in the Euclidean plane. Every pair of nodes that can communicate according to the SINR_G model ($\alpha = 3$) is connected by two links, one in each direction. This scenario represents a network in which all nodes try to communicate frequently. Similar models were used in [17], [18]. We assume that all links have the same traffic demands. This prevents that single links dominate the length of the computed schedules. Of course, all algorithms can just as well be used in situations with arbitrary link demands.

Our last scenario aims at recreating effects that occur in buildings with walls and obstacles. We assume that the signal strength falls of with path-loss exponent $\alpha = 2.5$ as long as there is no wall. Additionally, every wall that crosses the line of sight between a sender-receiver-pair results in normally distributed attenuation. Finally, the random effects that are caused by reflections and self-interference are represented by adjusting the signal strength of every link with a zero-mean, normally distributed random attenuation with $\sigma = 2\mathrm{dB}$. An example is shown in Figure 3c. One can see that this leads to some kind of cluster formation within rooms, a feature that does not show up in the other scenarios.

In all scenarios, we assume an omnipresent background noise η and minimum SINR $\beta = 10\mathrm{dB}$. The other model parameters are normalized such that the maximum transmission radius $d_{\max}$ with power $P_{\max} = 1$ equals distance 100 if there are no walls, and if we assume $\alpha = 2$ for the path loss exponent. For $\alpha = 3$, this gives $d_{\max} = 21.54$.

B. Input Generation

In the first scenario, we distributed between 100 and 2500 links randomly in an area with dimensions 400×400. First, the senders were placed randomly. Subsequently, the receivers were placed within a radius of $0.9d_{\max}$ around their senders. In the second scenario, we placed between 25 and 250 nodes randomly in an 200×200 area. All pairs of nodes with distance less than $0.9d_{\max}$ were connected. In this scenario, every node can act both as sender and receiver and be part of many links. However, in every time slot every node can only take part in one transmission.

In the indoor scenario, between 20 and 200 nodes were randomly distributed in an 200×200 area. Additionally, wall segments were placed on a regular 8×8 grid. On each of the 112 inner grid segments, a wall was put with probability 60%. The attenuation of each wall segment was determined by a normally distributed random value with $\mu = 5\mathrm{dB}$ and $\sigma = 2\mathrm{dB}$. Finally, links were added for all nodes that were able to communicate with each other according to the general SINR model.

C. Examined Algorithms

In order to evaluate our algorithms, we implemented several existing algorithms for the scheduling problem. For the sake of fairness, we chose algorithms that are especially designed and optimized for the physical SINR model. This section gives a short overview on the selected algorithms. In case an algorithm was not named by the authors, we gave a name based on author names and year of publication. Some of the examined algorithms do not include power control. In this cases, we computed optimum powers for the generated schedules.

ElBatt04 [18] was the first algorithm for scheduling with power control in the SINR model. The algorithm alternates between scheduling and power control and defers transmissions with minimum SINR until an admissible set of powers can be found. In contrast, *LiEph05* [30] starts with empty slots and adds links according to some scheduling metric that takes into account queue sizes and number of blocked links. Every time a slot is filled, a new slot is opened. A similar approach is used by *GreedyPhysical* [31]. Instead of the scheduling metric, an interference number is used to sort the links in the beginning. The links are processed according to their order and added to the first possible time slot. *DiGreedy* [27] is another approach to find a good compromise between schedule length and power consumption. The underlying principle is similar to the one in *ElBatt04*. Links with high interference are deferred until a feasible link set is found. Additionally, every feasible set is rated based on schedule length and power consumption. In order to optimize the relation of throughput and power consumption, the remaining links are removed one by one from the set of active links. In the end, from all feasible link sets the one with the best rating is chosen. Using a parameter, one can decide whether throughput or power efficiency is more important. *ApproxLogN* [25] was the first algorithm which guaranteed a $O(\log n)$ approximation for the problem of minimizing the number of time slots needed to schedule a given set of requests. The authors showed that their

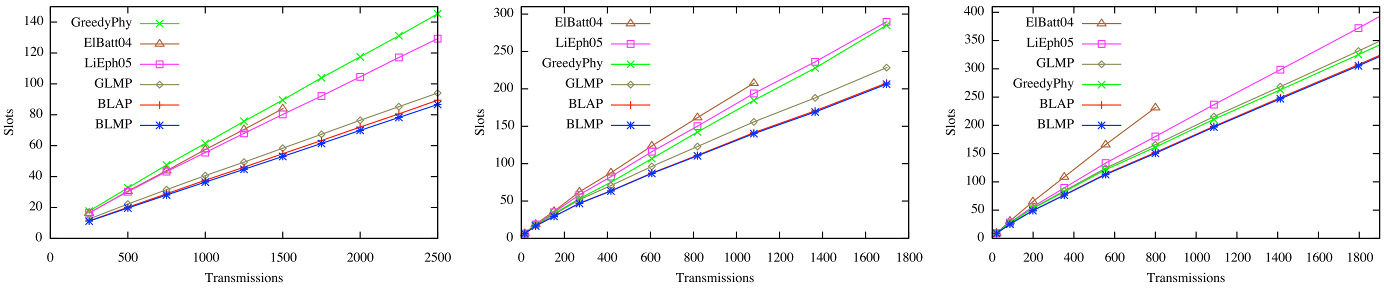


Fig. 4. Average schedule length. (a) Random links in free space. (b) Network in free space. (c) Network in building.

algorithm is superior to the algorithm *ApproxDiversity* which was proposed in [13]. However, the algorithm needs to know the sender-receiver distances and it relies on the special properties of the SINR_G model, thus it cannot be used in realistic scenarios with obstacles and random effects. *Hall09* [32] is a simplified version of *ApproxLogN*. The authors showed that the algorithm computes a constant factor approximation for the scheduling problem in the SINR_G model. Just as *ApproxLogN*, this algorithm relies on the properties of the SINR_G model.

D. *Throughput*

We start with an examination of the schedule lengths. The schedule length determines the communication throughput and all considered algorithms are intended for computing short schedules. Figure 4 shows how the schedule length increases with increasing link density. It is not surprising that the relation between schedule length and link density is almost linear. Apparently, not all algorithms perform equally well in all scenarios. Moreover, in the indoor scenario, the differences between the algorithms seem to diminish. However, in all scenarios the heuristics *BLMP* and *BLAP* compute the best schedules, with *BLMP* being marginally better. Table I compares average schedule lengths for certain inputs. For the random links scenario, the input consisted of 1500 random links. For the network and building scenarios, the input consisted of 200 and 120 random nodes, resulting in 1081 links and 797 links on average, respectively.

For the heuristics *BLMP* and *BLAP* two values are given. For the first value, the algorithm was initially started with only one open slot. Then, 80% of the length of the computed schedule was used as a better start value for a second iteration of the algorithm. One can see that this resulted in slightly better schedules. However, the difference is rather small. Obviously, the approximation algorithms *ApproxLogN* and *Hal09* give significantly worse throughput than the other algorithms. The reason is that they are mainly of theoretical interest. The focus was not on computing the best schedule possible, but on showing that one can give approximation guarantees. It would be easy to use those algorithms to compute an initial schedule and then to use the ideas proposed in this paper to improve the schedule, thus getting a competitive algorithm with approximation guarantees. However, it is unlikely that this would result in significantly better schedules.

E. *Power Consumption*

In some situations, for example in wireless sensor networks, the available energy is limited and one might be interested in energy efficient schedules. Figure 5 shows the dependence between average power consumption and link density. For every algorithm, the average transmission power in throughput optimal schedules is given as a fraction of P_{max}. Moreover, a lower bound P_{req} on the average transmission power is shown. P_{req} is defined as the average power that is used when all transmission are scheduled one by one, thus avoiding interference. One can see that the curves become flat for large numbers of links. A direct comparison of the algorithms is given for certain input sizes in Table II. This time, the powers are given relative to P_{req}, thus highlighting the overhead that

TABLE I
AVERAGE SCHEDULE LENGTH

Algorithm	Random (∅ 1500 links)	Network (∅ 1081 links)	Building (∅ 797 links)
ElBatt04	84.0	207.6	229.4
LiEph05	80.2	193.9	179.1
GreedyPhy	89.6	184.6	160.2
ApproxLogN	531.4	826.0	-
Hal09	1288.9	1051.9	-
GLMP	58.4	155.8	164.6
BLMP	57.1 / 53.0	141.6 / 140.1	149.4 / 149.4
BLAP	56.4 / 54.8	143.7 / 141.4	150.2 / 150.9

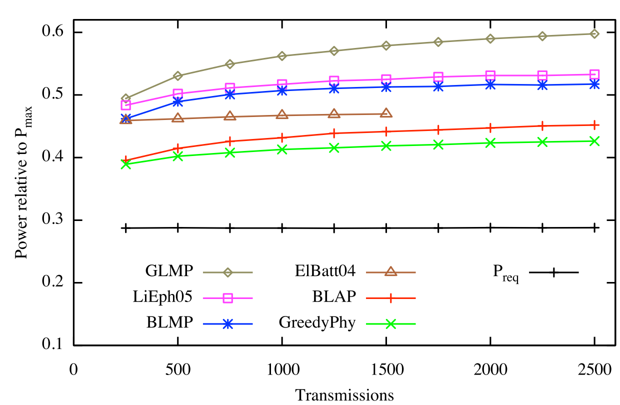
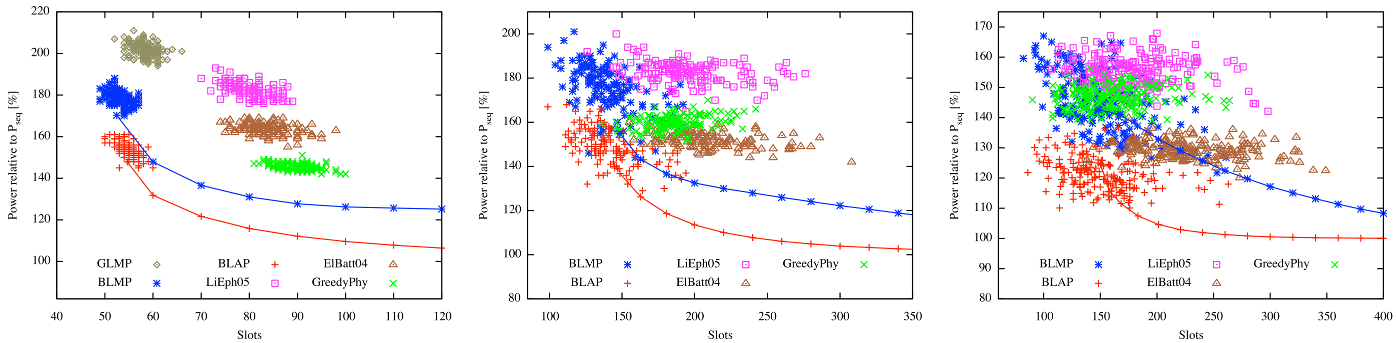


Fig. 5. Power consumption (random links in free space).



(a) (b) (c)

Fig. 6. Throughput vs. power consumption. (a) 1500 random links. (b) Network with 200 random nodes. (c) Network in building with 120 random nodes.

one has to accept for the higher throughput. However, when interpreting this numbers, one has to keep in mind that at this point we compare schedules of different lengths. It is no miracle that schedules with lower throughput need less power. Extreme examples are *ApproxLogN* and *Hal09*, as they only schedule transmissions together that hardly interfere with each other.

F. Throughput vs. Power Consumption

In this section, we try to make a fairer comparison regarding power consumption by also taking into account the throughput of the considered schedules. For this purpose, we randomly selected schedules and plotted them in Figure 6. The further left a point is, the higher is the throughput of the corresponding schedule, and the lower a point is, the less energy needs the schedule. In the random scenario, the algorithms are separated very nicely. Again, it becomes apparent that the algorithms *BLMP* and *BLAP* achieve the highest throughput. At the same time, *BLAP* also is highly competitive with respect to required energy. *GLMP* also computes short schedules. However, the slot-by-slot approach makes it much harder to distribute the transmissions beneficially, thus the energy demand is much higher. We already mentioned that *BLMP* and *BLAP* allow the computation of energy efficient schedules by using more slots than necessary. This compromise is visualized by the curves in Figure 6. For example, one can deduce that with an average power of about $1.2P_{\text{req}}$ one can schedule all 1000 links of the random link scenario within 70 slots. In all scenarios, the curve of *BLAP* is way below all other algorithms. Thus, given the same number of slots, the schedules computed by *BLAP* are significantly more economical. Compared to the first scenario, in the network scenarios the differences between the single algorithms are not so distinct. Especially in the indoor scenario, the random effects lead to much scatter. However, in all scenarios *BLAP* is evidently superior.

G. Network Lifetime

Energy aware schedules increase the lifetime of the network. We will examine this exemplary for the network scenario with 200 randomly distributed nodes. For this purpose, we consider the time that passes until 10%, 20%, 30%, 50%, and 75% of the nodes run out of battery, assuming that all links are active for about the same time and that all nodes have the same initial battery charge. The lifetime improvement that can be achieved by using the *BLAP* heuristic is shown in Table III. For example, when using *BLAP* instead of *ElBatt04*, using the same number of time slots, it takes 29.3% longer until 10% of the nodes run out of battery. The last row of Table III shows the advantage of a throughput optimized schedule using *BLAP* in comparison to scheduling without power control. It is obvious that even in short schedules with high throughput a lot of energy can be saved by using good power control.

VII. Conclusion

In this article, we considered several aspects of scheduling with power control in wireless networks. First, we showed that the problem is NP-hard in the SINR_G model. Subsequently, we described a power control algorithm that is optimized for situations where transmissions are added one by one. Thus, it can be used to improve several existing heuristics. However, the algorithm's main advantage is that it allows to look efficiently one step ahead. Given the solution for a set of active links, it is

TABLE II
AVERAGE POWER CONSUMPTION RELATIVE TO P_{REQ} [%]

Algorithm	Random (∅ 1500 links)	Network (∅ 1081 links)	Building (∅ 797 links)
ElBatt04	163.4	151.8	129.4
LiEph05	182.6	182.8	155.5
GreedyPhy	145.7	160.0	146.7
ApproxLogN	100.0	100.0	-
Hal09	100.0	100.0	-
GLMP	201.4	176.3	141.3
BLMP	190.1 / 178.4	178.9 / 175.3	139.6 / 145.4
BLAP	164.9 / 153.6	156.5 / 149.0	126.4 / 121.7

TABLE III
LIFETIME IMPROVEMENT BY HEURISTIC *BLAP* [%]

compared to	10%	20%	30%	50%	75%
ElBatt04	29.3	32.5	35.1	40.2	48.9
LiEph05	45.0	51.4	57.8	67.7	85.7
GreedyPhy	42.2	38.7	36.6	33.5	29.9
GLMP	26.4	31.2	35.4	43.1	56.0
Fixed Power	157.6	214.6	270.9	406.4	738.2

very cheap to see what would happen if another link is added. Based on this property, we proposed scheduling heuristics that gradually assign links to slots, taking into account all entailed effects. In order to evaluate our heuristics, we compared them to several existing approaches. For this purpose, we considered three different scenarios. In all scenarios, our algorithms outperformed the existing approaches in terms of throughput and power efficiency. Moreover, the algorithms can be used to find a good compromise between throughput and energy efficiency and they are also useful in situations where the number of available slots is predefined.

ACKNOWLEDGMENT

This work was partially supported by the German Research Foundation (DFG) within the Research Training Group GRK 1194 "Self-organizing Sensor-Actuator Networks".

REFERENCES

[1] L. C. Pond and V. O. K. Li, "A distributed time-slot assignment protocol for mobile multi-hop broadcast packet radio networks," in *Proc. of the IEEE Military Communications Conference (MILCOM'89)*. IEEE, 1989, pp. 70–74.

[2] R. Ramaswami and K. Parhi, "Distributed scheduling of broadcasts in a radio network," in *Proc. of the 8th IEEE Int'l Conf. on Computer Communications (INFOCOM'89)*, vol. 2. IEEE, 1989, pp. 497–504.

[3] S. Ramanathan and E. L. Lloyd, "Scheduling Algorithms for Multihop Radio Networks," *IEEE/ACM Transactions on Networking*, vol. 1, no. 2, pp. 166–177, 1993.

[4] G. Pei and C. Chien, "Low power TDMA in large wireless sensor networks," in *Proc. of the IEEE Military Communications Conference (MILCOM'01)*. IEEE, 2001, pp. 347–351.

[5] S. Coleri, A. Puri, and P. Varaiya, "Power Efficient System for Sensor Networks," in *Proc. of the 8th IEEE Symp. on Computers and Communications (ISCC'08)*. IEEE, 2003, pp. 837–842.

[6] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," in *Proc. of the 1st Int'l Conf. on Embedded Networked Sensor Systems (SenSys'03)*. ACM, 2003, pp. 181–192.

[7] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan, "End-to-end packet-scheduling in wireless ad-hoc networks," in *Proc. of the 15th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'04)*. SIAM, 2004, pp. 1021–1030.

[8] T. Moscibroda and R. Wattenhofer, "Coloring Unstructured Radio Networks," in *Proc. of the 17th Ann. ACM Symp. on Parallelism in Algorithms and Architectures (SPAA'05)*. ACM, 2005, pp. 39–48.

[9] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in wireless networks," in *Proc. of the 12th Annual Int'l Conf. on Mobile Computing and Networking (MobiCom'06)*. ACM, 2006, pp. 227–238.

[10] J. Grönkvist and A. Hansson, "Comparison between graph-based and interference-based STDMA scheduling," in *Proc. of the 2nd Int'l Symp. on Mobile Ad Hoc Networking & Computing (MobiHoc'01)*. ACM, 2001, pp. 255–258.

[11] R. Maheshwari, S. Jain, and S. R. Das, "A measurement study of interference modeling and scheduling in low-power wireless networks," in *Proc. of the 6th Conf. on Embedded Network Sensor Systems (SenSys'08)*. ACM, 2008, pp. 141–154.

[12] T. Moscibroda, R. Wattenhofer, and Y. Weber, "Protocol Design Beyond Graph-Based Models," in *5th Workshop on Hot Topics in Networks (HotNets'06)*, 2006.

[13] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer, "Complexity in Geometric SINR," in *Proc. of the 8th Int'l Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc'07)*. ACM, 2007, pp. 100–109.

[14] J. Zander, "Distributed cochannel interference control in cellular radio systems," *IEEE Transactions on Vehicular Technology*, vol. 41, no. 3, pp. 305–311, Aug 1992.

[15] G. J. Foschini and Z. Miljanic, "A simple distributed autonomous power control algorithm and its convergence," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 4, pp. 641–646, Nov 1993.

[16] S. Ulukus and R. Yates, "Stochastic power control for cellular radio systems," *IEEE Transactions on Communications*, vol. 46, no. 6, pp. 784–798, Jun 1998.

[17] T. ElBatt and A. Ephremides, "Joint scheduling and power control for wireless ad-hoc networks," in *Proc. of the 21st IEEE Int'l Conf. on Computer Communications (INFOCOM'02)*, vol. 2. IEEE, 2002, pp. 976–984.

[18] ——, "Joint scheduling and power control for wireless ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 3, no. 1, pp. 74–85, Jan. 2004.

[19] K. Wang, C. F. Chiasserini, R. R. Rao, and J. G. Proakis, "A Distributed Joint Scheduling and Power Control Algorithm for Multicasting in Wireless Ad Hoc Networks," in *Proc. of the Int'l Communication Conf. (ICC'03)*. IEEE, 2003, pp. 725–731.

[20] R. L. Cruz and A. V. Santhanam, "Optimal routing, link scheduling and power control in multihop wireless networks," in *Proc. of the 22nd Conf. of Computer Communications (INFOCOM'03)*, vol. 1. IEEE, 2003, pp. 702–711.

[21] R. Bhatia and M. Kodialam, "On power efficient communication over multi-hop wireless networks: joint routing, scheduling and power control," in *Proc. of the 23rd Conf. on Computer Communications (INFOCOM'04)*, vol. 2. IEEE, 2004, pp. 1457–1466.

[22] U. C. Kozat, I. Koutsopoulos, and L. Tassiulas, "A Framework for Cross-layer Design of Energy-efficient Communication with QoS Provisioning in Multi-hop Wireless Networks," in *Proc. of the 22nd Conf. of Computer Communications (INFOCOM'04)*. IEEE, 2004.

[23] T. Locher, P. von Rickenbach, and R. Wattenhofer, "Sensor Networks Continue to Puzzle: Selected Open Problems," in *9th International Conference on Distributed Computing and Networking (ICDCN'08)*, 2008.

[24] T. Moscibroda, Y. A. Oswald, and R. Wattenhofer, "How Optimal are Wireless Scheduling Protocols?" in *Proc. of the 26th Int'l Conf. on Computer Communications (INFOCOM'07)*. IEEE, May 2007, pp. 1433–1441.

[25] 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.

[26] M. M. Halldórsson, "Wireless Scheduling with Power Control," in *Proc. of the 17th European Symp. on Algorithms (ESA'09)*. Springer, 2009, pp. 361–372.

[27] G. Lu and B. Krishnamachari, "Energy efficient joint scheduling and power control for wireless sensor networks," in *2nd Ann. Conf. on Sensor and Ad Hoc Communications and Networks (SECON'05)*. IEEE, 2005, pp. 362–373.

[28] M. Völker, B. Katz, and D. Wagner, "On the Complexity of Scheduling with Power Control in Geometric SINR," Fakultät für Informatik, Universität Karlsruhe, Tech. Rep. 2009-15, 2009, http://digbib.ubka.uni-karlsruhe.de/volltexte/1000012389.

[29] R. M. Karp, "Reducibility Among Combinatorial Problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. Plenum Press, 1972, pp. 85–103.

[30] Y. Li and A. Ephremides, "Joint Scheduling, Power Control, and Routing Algorithm for Ad-Hoc Wireless Networks," in *Proc. of the 38th Ann. Hawaii Int'l Conf. on System Sciences (HICSS'05)*. IEEE, 2005.

[31] G. S. Brar, D. M. Blough, and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," in *Proc. of the 12th Annual Int'l Conf. on Mobile Computing and Networking (MobiCom'06)*. ACM, 2006, pp. 2–13.

[32] M. M. Halldórsson and R. Wattenhofer, "Wireless Communication is in APX," in *Proc. of the 36th Int'l Coll. on Automata, Languages and Programming (ICALP'09)*. Springer, 2009, pp. 525–536.