Dynamic Power Allocation Under Arbitrary Varying Channels – An Online Approach

Niv Buchbinder
Microsoft Research, New England
Email: nivbuchb@microsoft.com

Liane Lewin-Eytan
Department of Electrical Engineering
Technion, Israel Institute of Technology
Email: liane@tx.technion.ac.il

Ishai Menache
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Email: ishai@mit.edu

Joseph (Seffi) Naor
Computer Science Department
Technion, Israel Institute of Technology
Email: naor@cs.technion.ac.il

Ariel Orda
Department of Electrical Engineering
Technion, Israel Institute of Technology
Email: ariel@ee.technion.ac.il

***Abstract*—A major problem in wireless networks is coping with limited resources, such as bandwidth and energy. These issues become a major algorithmic challenge in view of the *dynamic* nature of the wireless domain. We consider in this paper the single-transmitter power assignment problem under time-varying channels, with the objective of maximizing the data throughput. It is assumed that the transmitter has a limited power budget, to be sequentially divided during the lifetime of the battery. We deviate from the classic work in this area, which leads to explicit "water-filling" solutions, by considering a realistic scenario where the channel state quality changes *arbitrarily* from one transmission to the other. The problem is accordingly tackled within the framework of *competitive analysis*, which allows for worst case performance guarantees in setups with arbitrarily varying channel conditions.**

We address both a "discrete" case, where the transmitter can transmit only at a fixed power level, and a "continuous" case, where the transmitter can choose any power level out of a bounded interval. For both cases, we propose online power-allocation algorithms with proven worst-case performance bounds. In addition, we establish lower bounds on the worst-case performance of *any* online algorithm, and show that our proposed algorithms are optimal.

I. Introduction

A. Background and Motivation

Wireless technologies are broadly used nowadays for both data and voice communications. The transmission protocols of wireless devices need to cope with limited resources, such as bandwidth and energy. Additional difficulties relate to the *dynamic* nature of wireless networks. For example, the mobility of terminals and the frequent change in their population introduces new challenges for routing and resource allocation protocols. Another central dynamic feature of wireless communications, which is the focus of this work, is the possibly frequent time variation in the channel quality between sender and receiver, an effect known as *channel fading* [5].

Much research has been devoted to study optimal power allocation in face of varying channel conditions, assuming that a (typically mobile) transmitter, wishing to maximize its throughout, has an average power constraint to sustain over time. If the channel state is known prior to transmission, the transmitter may obtain the optimal mapping from channel states to power levels via the solution to a convex optimization problem [8]. It turns out that the single-user closed-form solution to this problem is a "water-filling" algorithm. Intuitively, this algorithm makes sure that higher power levels are kept for better channel states.

The "water-filling" solution relies on an a-priori knowledge of the channel-state distribution. However, such information may not be available, and therefore requires adaptive schemes to estimate it. Even worse, the probability rule governing the underlying channel state process might change over time, due to non-stationary network elements that affect the quality of transmissions (e.g., mobility, line of sight, etc.). The goal of this study is to investigate how well can a transmitter do under *arbitrarily varying* channel conditions.

An additional distinctive assumption of our model is that the transmitter has a limited battery that can be recharged only occasionally. Hence, instead of considering a long-term power average constraint, the transmitter has to be aware of its actual remaining energy. Consequently, the underlying optimization task becomes a *dynamic* power control problem (rather than a static mapping from channel states to power levels). Due to the arbitrarily changing channel conditions, we study the problem within the framework of *online computation* [3], with the objective of devising online power-allocation algorithms with proven worst-case performance bounds. A second objective is to establish lower bounds on the worst-case performance of *any* online algorithm that operates under arbitrarily varying channel conditions, hence providing a benchmark for the quality of our proposed solutions.

The technological relevance of our work lies, for example, in sensor networks, where the battery of the mobile is limited and can be charged only occasionally (e.g., by solar energy). Sensors that are required to send informative data, may do so in a relatively slow pace, with the objective of maximizing their overall throughput. Due to the low rate of transmission, the assumption of arbitrary channel conditions is commensurate with the unknown changes (e.g., environmental) that take place between subsequent transmissions.

B. Related Literature

The information theory community has considered the case where the transmitter and the receiver operate with incomplete knowledge of the probability law governing the channel over which transmission takes place. This situation is usually modeled as having an adversarial jammer, whose goal is to diminish system capacity. Various models for such channels and their corresponding capacities have been quite broadly analyzed (see [10] for a survey). The problem that we consider here is fundamentally different: while we assume that the transmitter observes the current channel state, and encodes accordingly, it cannot predict future channel states, and therefore should carefully choose its current power allocations, considering its limited power budget.

Recently, there has been growing interest in *jamming games* (e.g., [1]), in which a malicious adversary, equipped with its own power budget, aims at deteriorating system performance by allocating its own power (which affects the throughput of other users) in a harmful way. Our work differs from the jamming game model by considering arbitrary *gain* (instead of power), which is not subject to a "budget" constraint. In addition, we focus on *competitive* (worst-case) analysis, rather than on the notion of an equilibrium between "equal" players. For such settings, where piece-by-piece decisions need to be made on an arbitrary input pattern, the methodology of *competitive analysis* [3] provides a framework for the systematic design of algorithmic solutions as well as for the establishment of worst-case performance bounds. These bounds are specified in terms of the *competitive ratio* of the online algorithm, which is the worst-case ratio (considering any possible input pattern) between its performance and that of an optimal off-line algorithm, which can observe the entire input sequence. Online methods have gained prominence in solving algorithmic problems in a variety of networking domains, ranging from network switches to e-bay and sponsored search auctions.

C. Contribution and Paper Organization

To the best of our knowledge, this is the first study that proposes to attack the problem of power allocation under dynamic channel quality through the methodology of online (competitive) analysis. Within this framework, we address two scenarios, which correspond to different technological capabilities of the transmitter. In the first,"discrete" scenario, the transmitter can transmit only at a fixed power level, hence its sole decision at each stage is whether to transmit or not. In the second, "continuous" case, the transmitter can choose the power level out of a continuous interval. For each of the two scenarios, we propose an algorithmic solution (an online algorithm), for which we establish a worst-case performance bound. In addition, we establish lower bounds on the performance of any online algorithm, hence benchmarking our solutions. More specifically, our contributions can be summarized as follows:

- Discrete case:
  - We provide a simple ("thresholds") online algorithm, for which we establish a worst-case performance bound.
  - We establish a bound on the performance of any online algorithm, and show that our proposed algorithm is within a small "gap" away from that bound.
  - We show that the above results, in both directions, are maintained also if some limits are imposed on the arbitrariness of the input pattern.
- Continuous case:
  - We provide a simple ("bins") online algorithm, for which we establish a worst-case performance bound.
  - We establish a bound on the performance of any online algorithm, and show that our proposed algorithm is optimal, in the sense that it *matches* that bound.
  - We also consider the case where the channel conditions can vary only within a bounded range, and obtain for this case a simple ("guessing") online algorithm, whose performance depends (quadratically) on the size of the bounded range.

We complement our work with a simulation study, where we validate our suggested online algorithm for the continuous case, and examine the effect of certain parameters on its performance. We further improve the algorithm for which we gave a complete theoretical analysis, and add several heuristic enhancements. Following our experiments, we observe that our online algorithm performs significantly better than the theoretical bound, resulting in a ratio of approximately 2.5 between the performance of the optimal off-line algorithm and the performance of our online algorithm.

The paper is organized as follows. The channel and transmitter are modeled in Section II. Section III addresses the discrete case, whereas the continuous case is treated in Section IV. Section V presents the simulation study and discusses its results. Finally, conclusions appear in Section VI. Due to space limits, several proofs as well as some technical details are omitted from this version, and can be found in [7].

II. The Model

A. The Channel Model

We consider a transmitter who transmits to a single receiver (base station) over a bandwidth of W hertz. The channel between the user and the receiver is modeled as a frequently-flat fading channel with additive white Gaussian noise. Specifically, at each time t, the received signal $y(t)$ is given by

$$y(t) = \sqrt{\tilde{h}(t)}x(t) + z(t), \qquad (1)$$

where $x(t)$ and $\tilde{h}(t) \geq 0$ are the transmitted signal and channel gain (state), and $z(t)$ is an additive white Gaussian noise with power spectral density $N_0/2$. The sequence of channel gains is modeled as a block-fading process [5], so that for $i = 1, 2, \ldots$

$$\tilde{h}(t) = \tilde{h}_i, \quad \text{for all } t \in [iL, (i+1)L),$$

where L is the length of each time slot.

A distinctive feature of our model is that the process $\{\tilde{h}_i\}$ evolves *arbitrarily*, i.e., without an underlying probability rule. At the beginning of each time slot i, the transmitter obtains some information h_i regarding the current channel gain $\tilde{h}_i$. This information is passed through a finite lossless feedback link with capacity of C bits per second[1]. The information $h_i \in \{q_0 = 0, q_1, \ldots, q_M\} \subset \mathcal{R}_+$ is a quantized version of the actual gain $\tilde{h}_i$, so that if $\tilde{h}_i \in [q_m, q_{m+1})$ (where $q_{M+1} \equiv \infty$), then $h_i = q_m$. Throughout the paper, we use the notation $h_{\min} = q_1$ for the smallest (nonzero) quantized gain, and $h_{\max} = q_M$ for the maximum one. To simplify the exposition, we shall henceforth refer to h_i as the channel gain at time-slot i.

The transmitter observes h_i and can adapt its transmission decision (which may include power adaptation) accordingly. We assume that there is no retransmission mechanism, so that each transmission arrives to the base station with a very high probability. The basic measure that determines the instantaneous throughput is the received Signal to Noise Ratio (SNR), given by $\mathrm{SNR}_i = \frac{h_i p_i}{N_0 W}$, where p_i is the transmission power of the transmitter at time i. To simplify notation, we normalize $N_0 W$ to one, so that $\mathrm{SNR}_i = h_i p_i$.

Let $U(\mathrm{SNR}_i)$ be the instantaneous rate which the user can reliably transmit at time slot i. In the bulk of the paper, we shall consider the function

$$U(\mathrm{SNR}_i) = \log(1 + \mathrm{SNR}_i), \tag{2}$$

which models the case where the transmitter can adjust its coding scheme to obtain rates approaching the Shannon capacity of (1) at each time slot.

B. User Model

We assume that a transmitter has an initial power budget of P that can be divided between different time slots. We further assume that a transmitter can recharge its battery (to the initial power P); however, due to practical limitations, a period of T time slots elapses between consecutive battery charges.

The transmitter wishes to maximize its total throughput subject to the constraints described above. We often refer to the total throughput achieved as *profit*. The general optimization problem is thus the following

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{T} U(\mathrm{SNR}_i) \\ s.t \quad & \\ & \sum_{i=1}^{T} p_i \le P, \end{aligned} \tag{3}$$

where $\mathrm{SNR}_i = h_i p_i$.

[1]E.g., the feedback link may employ a different frequency band in a Frequency Division Duplex (FDD) system, which does not suffer from fading.

In Section III we consider the discrete power allocation case, in which the transmitter has a fixed power level P^* for each transmission. This case corresponds to imposing an additional constraint $p_i \in \{0, P^*\}, \forall i = 1, \ldots, T$, to the above optimization problem. In Section IV we address the continuous case in which any power level between zero and P can be used, which corresponds exactly to (3).

We emphasize that since the channel gain sequence $\{h_i\}$ is not known a-priori, and neither is its distribution, (3) cannot be solved off-line. Accordingly, we pose (3) as an online optimization problem, where at each time slot i, a new channel gain h_i is revealed to the transmitter.

III. The Discrete Case

We consider here the discrete version of the problem, where at each time slot the transmitter can decide whether to transmit at a fixed power P^*, or not to transmit at all. The total number of transmissions that can be performed is thus equal to $P/P* = k$. We refer to this problem as the *k-transmissions problem*. The k-transmissions problem can thus be formulated as an integer linear program:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{T} x_i \cdot \log(1 + h_i P^*) \\ s.t. \quad & \\ & \sum_{i=1}^{T} x_i \le k && (4) \\ \forall \text{ time slot } i: \quad & x_i \in \{0, 1\}. && (5) \end{aligned}$$

We refer to the above problem as the *integral k-transmissions* case. The *fractional k-transmissions* problem is obtained by relaxing constraint (5) to $0 \le x_i \le 1$ (for each slot i). In practice, the fractional problem captures situations where a single transmission can be split to fractions (adding up to 1) over several slots. Thus, x_i denotes the fraction of the time slot in which the transmission is performed.

As specified in Section II-B, we assume that the channel gains obtain values between zero and $h_{\max}$. Since a gain of zero will result in no transmission, we may ignore the zero-gain instances, and assume that the channel gain obtains values in the range $[h_{\min}, h_{\max}]$. For ease of notation, we denote the maximum gain from an integral transmission by $M = \log(1 + h_{\max} \cdot P^*)$ and the minimum gain by $m = \log(1 + h_{\min} \cdot P^*)$.

Thus, our online problem can be described as follows. At each time slot i a new gain value $v_i \in [m, M]$ is given, and the online algorithm has to decide on the value of x_i, so as to maximize the total gain, subject to the above constraints. The value of x_i cannot be changed in the future. Our goal is to find an online algorithm with the best possible competitive ratio. We note that the fractional k-transmissions case can be solved using the online primal-dual approach of [6], leading to an $O(\log \frac{M}{m})$-competitive algorithm.

A. The Integral Case: A Randomized Algorithm

We now describe a simple randomized algorithm achieving a competitive ratio of $O(\log \frac{M}{m})$ for the integral k-

transmissions problem. We first run the online algorithm for the fractional k-transmissions problem using the techniques of [6], and then apply randomized rounding *online* to the fractional solution. The output of the fractional online solution is a partitioning of the time frame into k time intervals, where in each time interval the fractional values add up to 1. Note that this partitioning is generated online by the fractional algorithm. For each interval j, we denote the fractions assigned to the time slots of the interval by $x_1^j, x_2^j, \ldots, x_{n_j}^j$, and the respective gain values of these time slots by $v_1^j, v_2^j, \ldots, v_{n_j}^j$. We derive an integral solution with expected profit equal to the fractional solution. To this end, in the beginning of each interval j, we choose uniformly in random a value θ^j from $[0,1]$. Define index α_j as the index satisfying $\sum_{i=1}^{\alpha_j-1} x_i^j < \theta^j$ while $\sum_{i=1}^{\alpha_j} x_i^j \geq \theta^j$. Clearly, for each j, the probability that index α_j is chosen is equal to $x_{\alpha_j}^j$. Thus, $\mathbb{E}[v_{\alpha_j}^j] = \sum_{i=1}^{n_j} x_i^j \cdot v_i^j$, i.e., the expected profit of our solution is equal to the profit of the fractional solution. Note that we need to choose the values θ^j independently for each interval. As the profit of the optimal (off-line) fractional solution is at least the profit of the optimal (off-line) integral solution, and since the competitive ratio achieved by the fractional online algorithm is $O(\log \frac{M}{m})$, our randomized online algorithm for the integral case achieves a competitive ratio of $O(\log \frac{M}{m})$ as well.

B. The Integral Case: A Deterministic Solution

We turn to analyze the deterministic integral k-transmissions case. The online integral k-transmissions problem can be directly reduced to an instance of the online knapsack problem, where the input consists of a knapsack of capacity k, and a stream of items of weight 1 and values v_i. An online algorithm with competitive ratio equal to $\ln \frac{M}{m} + 1$ is known for this problem [12]. However, the analysis of this algorithm is valid only for the case where $k >> 1$. We complete the picture and present an algorithm for the case where k is small.

1) The k-Thresholds Online Algorithm: We first compute k threshold values denoted by $w_1, \ldots, w_k$. The *k-thresholds online algorithm* for the integral k-transmissions problem proceeds as follows. For each transmission j (where j goes from 1 to k): if there are more than $k-j$ additional time slots until the end of the time frame, we transmit transmission j only if the gain value of the current time slot is not less than threshold w_j; otherwise, transmission j is done at the current time slot, independently of the gain value.

Lemma 1: The optimal competitive ratio of the k-thresholds algorithm is achieved for an increasing sequence of thresholds, that is, a sequence $w_1 < w_2 < \ldots w_k$.

Finding an optimal choice of thresholds $w_1, w_2, \ldots, w_k$ requires the consideration of several different inputs given to the algorithm by an adversary. Due to space limits, we do not describe here the different cases. A near-optimal choice of a sequence of thresholds turns out to be a geometric series, where $w_1 = (km)^{\frac{k}{k+1}} M^{\frac{1}{k+1}}$, and the ratio between successive thresholds is $\rho = \left(\frac{M}{km}\right)^{\frac{1}{k+1}}$. The competitive ratio of the k-thresholds algorithm, given the latter sequence of thresholds, is summarized by the following theorem.

Theorem 1: The integral k-transmissions online problem can be solved by a deterministic k-thresholds algorithm with competitive ratio $O(k^{\frac{k}{k+1}} \cdot \left(\frac{M}{m}\right)^{\frac{1}{k+1}})$.

Note that for the case where $k = 1$, our problem reduces to the online portfolio selection problem [4], where a certain amount of money has to be changed from one currency to another, and the exchange rates arrive online. For this problem, we get a threshold of $\sqrt{(mM)}$ and a ratio of $\sqrt{\left(\frac{M}{m}\right)}$ which match the optimal solution for this problem.

2) Lower Bound: We present a lower bound of $\Omega(\left(\frac{M}{m}\right)^{\frac{1}{k+1}})$ for the integral k-transmissions problem. The lower bound consists of a sequence of gain values which depend on the decisions of the online algorithm solving this problem. The gain value of the first time slot is some value w'_1, to be specified later on. Now, if the algorithm performs a transmission, then the gain value of the next slot is changed to be w'_2 (to be specified later), otherwise, it stays w'_1. Similarly, in case the algorithm already performed j transmissions, the gain value will be w'_{j+1} (to be specified later as well) and will stay as is until the algorithm performs the next transmission. If the algorithm does not make an additional transmission and reaches the $k-j$ last time slots, the gain values are changed to be the minimum value m until the end of the sequence. Otherwise, as soon as the algorithm performs k transmissions, the gain values are changed to be the maximum value M until the end of the sequence.

Optimizing the choice of $w'_1, w'_2, \ldots$ is similar to the analysis of the k-thresholds algorithm in Section III-B1. We get a geometric series of thresholds, where $w'_1 = (m)^{\frac{k}{k+1}} M^{\frac{1}{k+1}}$, and the ratio between successive thresholds is $\rho' = \left(\frac{M}{m}\right)^{\frac{1}{k+1}}$. The lower bound is summarized in the next theorem.

Theorem 2: The competitive ratio of any algorithm solving the integral k-transmissions online problem is at least $\Omega(\left(\frac{M}{m}\right)^{\frac{1}{k+1}})$.

There is a gap of at most a factor of k between our lower and upper bounds. However, for small values of k we get nearly matching upper and lower bounds.

3) Progressive Change of Gain Values: Finally, we consider the case where successive gain values (arriving online) in the range $[m, M]$ cannot be arbitrary, but are rather limited to be as far as Δ from each other. That is, given that $v_i = x$, then $v_{i+1} \in [x - \Delta, x + \Delta]$. We show that the results presented in Sections III-B1 and III-B2 for the case of arbitrary gain values remain valid here as well. Our online algorithm stays as described in Section III-B1 (the worst-case sequences can be easily adapted to be progressive).

The lower bound in the case where the gain values can only change progressively can be directly derived from the lower bound described in Section III-B2, leading to a similar result. Due to space limits, we do not further elaborate on this case.

IV. The Continuous Case

We consider the general case of the continuous online power allocation problem defined in Section (II-B), where a transmitter maximizes its total throughput given an initial power of P, and can use any power level at each time slot (subject to the remaining power constraint). The optimization problem of the transmitter for the continuous case is:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{T} \log(1 + h_i p_i) \\ s.t \quad & \\ & \sum_{i=1}^{T} p_i \leq P. \end{aligned}$$

A. Lower Bound

As explained in Section II-B, we assume that channel gains are within the range $[h_{\min}, h_{\max}]$. For ease of notation, we denote $h = h_{\min}$ and $H = h_{\max}$. We present a lower bound of $\Omega(\log \frac{H}{h})$ for the general continuous case. We set $P = \frac{H}{h}$; this will guarantee that the throughput achieved by the algorithm over time slots with channel gains equal to h is negligible, as is explained later.

The lower bound consists of a sequence of channel gain values that change every ℓ slots. We refer to such as a set of consecutive ℓ slots as a *sub-sequence*. The number of slots ℓ in each such sub-sequence is large enough, so that even if we divide the total power P evenly between all slots in a sub-sequence with gain value x, the throughput achieved can be nearly approximated using the following equation:

$$\sum_{t=1}^{\ell} \log\left(1 + \frac{P \cdot x}{\ell}\right) = \ell \cdot \log\left(1 + \frac{P \cdot x}{\ell}\right) \cong P \cdot x.$$

The idea is as follows. Suppose the goal of the adversary is to reach a competitive ratio of r. Then, as soon as the ratio achieved is at least r, the adversary "stops" giving sub-sequences, and turns to give a minimum gain of h till the end. Otherwise, as long as the competitive ratio achieved along the sub-sequences is less than r, the adversary gives consecutive sub-sequences with gain values that are doubled each time, until the budget of the algorithm is exhausted. At this point, the adversary gives a maximum gain value of H till the end. The adversary will then reach its desired ratio by investing its power over the slots with gain H.

We set the channel gain values of the time slots of the first sub-sequence to be $\frac{H}{\sqrt{P}}$. This value is chosen so that the algorithm cannot achieve a valuable throughput with respect to the adversary by transmitting over slots with minimum gain h, as shown by the following lemma.

Lemma 2: Suppose that after $x \geq 1$ sub-sequences, the competitive ratio achieved by the algorithm is $r' \geq r = \Theta(\log \frac{H}{h})$, and the adversary then gives a minimum gain of h till the end. Then, the competitive ratio achieved at the end of the sequence cannot be lower than r.

We denote by $r_1, r_2, \ldots, r_j$ the competitive ratios achieved after the first, second, and jth sub-sequence, respectively. Note that r_i $(1 \leq i \leq j)$ is the ratio between the optimal solution and the total throughput achieved by the algorithm after sub-sequences $1, \ldots, i$. Clearly, for each i, $r_i < r$, as otherwise the lower bound is reached, and the adversary stops the sub-sequences. We first prove the following lemma.

Lemma 3: The total budget invested by the algorithm over these j sub-sequences is at least the total budget invested by the algorithm in case the ratio after each subsequence is equal to r (more precisely to $r - \epsilon$ for ϵ arbitrarily small).

Proof: We compute the total budget spent by the algorithm. We denote by y_i the budget fraction spent over sub-sequence i, and by x_i the gain value given in slots of sub-sequence i. The maximum profit made by the algorithm over the first sub-sequence is maximized if the budget spent, Py_1, is evenly split between the ℓ slots of the sub-sequence. The profit of the algorithm is then equal to $\ell \cdot \log(1 + \frac{Py_1}{\ell} \cdot x_1) = Py_1 x_1$. Similarly, the adversary makes a maximum profit by equally dividing all its power over these ℓ slots. The profit of the adversary is then equal to $\ell \cdot \log(1 + \frac{P}{\ell} \cdot x_1) = Px_1$. As the ratio after the first sub-sequence is r_1, we get that $Py_1 = P/r_1$.

We repeat this calculation for the second sub-sequence. Again, the throughput of the adversary is Px_2. The maximum throughput achieved by the algorithm over sub-sequences 1 and 2 is $Py_1x_1 + Py_2x_2$. As the competitive ratio achieved after sub-sequence 2 is r_2, we get that $Py_1x_1 + Py_2x_2 = \frac{Px_2}{r_2}$. Substituting y_1 by $1/r_1$, and as $x_2 = 2x_1$ (the gain has doubled), we get that $Py_2 = \frac{P}{r_2} - \frac{P}{2r_1}$.

More generally, the maximum throughput of the algorithm over sub-sequences $1, \ldots, i$ is $\sum_{z=1}^{i} Py_z x_z$, whereas the maximum profit of the adversary is $P \cdot x_i$. Thus, $\sum_{z=1}^{i} Py_z x_z = \frac{Px_i}{r_i}$. We can write each term y_z as a function of r_z and r_{z-1} (as specified above for y_2). Doing so, and as $x_z = 2x_{z-1}$, we get that the budget spent by the algorithm over sub-sequence i is $Py_i = \frac{P}{r_i} - \frac{P}{2r_{i-1}}$. Thus, the total budget spent by the algorithm over sub-sequences $1, \ldots, j$ is

$$\sum_{z=1}^{j} Py_z = \frac{1}{2} \sum_{z=1}^{j-1} \frac{P}{r_z} + \frac{P}{r_j}. \tag{6}$$

Note that the adversary always makes a maximum profit by equally dividing all its power over the slots of the last sub-sequence. Now, in case for each i, $r_i = r$ (more precisely, $r - \epsilon$), the algorithm spends a budget of P/r over the first sub-sequence, and a budget of $P/2r$ over each other sub-sequence. In that case, the total budget spent by the algorithm over sub-sequences $1, \ldots, j$ is

$$(j-1) \cdot \frac{P}{2r} + \frac{P}{r}. \tag{7}$$

For each i, $r_i < r$, (6) dominates (7), proving the lemma.

Note that in case $1/r_i \leq 1/2r_{i-1}$, the algorithm doesn't transmit over sub-sequence i. In that case, terms of the form $(\frac{P}{r_i} - \frac{P}{2r_{i-1}})$ might be subtracted from (6). However, in that case, $2r_{i-1} \leq r_i < r$. Thus, $P/r_{i-1} > 2P/r$, and the lemma remains valid. ■

¿From the above lemma it follows that the case where the algorithm spends its whole budget after a maximum number of sub-sequences happens when the ratio achieved at the end of each sub-sequence is $(r - \epsilon)$. We compute the maximum number of such sub-sequences, that is, the maximum number of sub-sequences needed in order to make the algorithm spend all of its budget.

As explained before, we set the gain of the first sub-sequence slots to be $\frac{H}{\sqrt{P}}$. The gain values of the last possible sub-sequence are set to be $\frac{H}{\log P}$. The value of $\frac{H}{\log P}$ is chosen to guarantee that by dividing its power equally over slots with maximum gain H, the adversary can guarantee a competitive ratio of $\Omega(\log P)$, as shown by the following lemma.

Lemma 4: Consider the case where the last sub-sequence is reached, and thus, the algorithm finishes its budget. Then, the competitive ratio achieved at the end of the whole sequence is $\Omega(\log P)$.

Proof: The adversary achieves a maximum throughput of $P \cdot H$ by dividing its power equally over slots with maximum gain H. The profit of the algorithm is upper bounded by $\frac{P \cdot H}{\log P}$, which is the throughput achieved in case the budget P is split equally over the slots of the last sub-sequence. (Note that this is an upper bound on the throughput, as the algorithm invests part of its power in earlier sub-sequences, with lower gain values). Thus, the competitive ratio obtained at the end of the whole sequence is $\Omega(\log P)$. ■

We denote the maximum number of sub-sequences by $\beta+1$. As in each sub-sequence the gain is doubled, $\frac{H}{\sqrt{P}} \cdot 2^{\beta} = \frac{H}{\log P}$ holds, and thus

$$\beta = \frac{\log P}{2} - \log\log P. \tag{8}$$

Furthermore, the budget of the algorithm must be totally spent after at most $\beta + 1$ sub-sequences. Thus,

$$\begin{aligned} \frac{P}{r} + \beta \cdot \frac{P}{2r} &= P \\ \beta &= 2(r-1). \end{aligned} \tag{9}$$

It now follows from (8) and (9) that $r = \Theta(\log P)$. (Note that $\log\log P$ is a lower order term.) As $P = \frac{H}{h}$, we get the following theorem.

Theorem 3: The competitive ratio of any algorithm for the continuous online power allocation problem is $\Omega(\log(\frac{H}{h}))$.

B. The Online Power Allocation Algorithm

In this section we design an online algorithm for the continuous power allocation problem. We follow a two step approach. First, in Section IV-B1 we analyze a special case in which the gain value range is bounded, and the ratio between the maximum gain value, H, and the minimum gain value, h, is at most λ. We present an online algorithm with competitive ratio $O(\lambda^2)$ for this case. Furthermore, we consider an extension of the bounded range case, where the budget given to the optimal off-line algorithm is different from the budget given to the online algorithm. Denoting the ratio between the two budgets by $\rho \geq 1$, we present an algorithm with competitive ratio $O(\rho \cdot \lambda^2)$ for this case. Based on this special case we design in Section IV-B2 an algorithm for the general case.

1) Bounded Range of Gain Values: In this section we study the case where the ratio between the maximum and minimum possible gain values is at most λ. Given a sequence of gain values in the range $[h, \lambda h]$, we describe an online algorithm with competitive ratio $O(\lambda^2)$.

We denote by $\mathcal{OPT}^{\lambda}$ the optimal off-line algorithm, and by $\mathcal{ALG}^{\lambda}$ our online algorithm, for sequences with bounded range of gain values $[h, \lambda h]$. We denote by $OPT^{\lambda}(P)$ and $ALG^{\lambda}(P)$ the profits of the optimal off-line algorithm and our online algorithm for such a sequence (that is, their total respective throughput), given a budget of P, and refer to them simply as OPT^{λ} and ALG^{λ} for ease of notation.

Let T be the number of time slots, and P be the total budget. Given the concavity of the objective function, and that the gain value in each time slot is bounded by λh, we derive the following upper bound on the value of the optimal value:

Observation 1:

$$OPT^{\lambda} \leq T \cdot \log\left(1 + \frac{P}{T} \cdot \lambda h\right).$$

The algorithm works as follows. It guesses the length T of the sequence, starting from 1, and doubling it each time the current length of the online sequence turns out to be longer than the guess. For a sequence length T, the algorithm invests in each time slot a power equal to $(\frac{P}{h \cdot T \cdot c})^{1/2}$, where c is a constant to be determined later on. That is, when at time T the algorithm realizes that its current guess is wrong, it updates its guess to be $2T$, and works with this value until time $2T$. At time $2T$ the algorithm will again update its guess to $4T$, etc. The algorithm continues to invest some power until the sequence length becomes longer than $P \cdot \lambda h$. After that point, the algorithm no longer invests any power, and thus does not make any additional profit.

Lemma 5: For a sequence of gain values in $[h, \lambda h]$:

1) The competitive ratio achieved by the algorithm is $4c \cdot \lambda$.
2) The power spent over all the time slots of the sequence is at most P.

Proof: We start with the proof of (1). For a sequence of length $T \leq P\lambda h$, power of $(\frac{P}{h \cdot T \cdot c})^{1/2}$ is spent over each of the $T/2$ last slots (as the length T is guessed after slot number $T/2$). In each of these slots, the gain value is at least h, and thus the profit made by the online algorithm is at least

$$\begin{aligned} ALG^{\lambda} &\geq \frac{T}{2} \log\left(1 + h \cdot \left(\frac{P}{h \cdot T \cdot c}\right)^{1/2}\right) \\ &\geq \frac{T}{4} \log\left(1 + \frac{P \cdot h}{T \cdot c}\right) && (10) \\ &\geq \frac{T}{4c \cdot \lambda} \log\left(1 + \frac{P}{T} \cdot \lambda h\right) && (11) \\ &\geq \frac{OPT^{\lambda}}{4c \cdot \lambda}. && (12) \end{aligned}$$

Inequality (10) follows since $1 + x^{1/2} \geq (1+x)^{1/2}$, and Inequality (11) follows since $1 + x/\alpha \geq (1+x)^{1/\alpha}$. Inequality (12) follows from observation (1).

If $T > P\lambda h$ then we bound the profit made by the algorithm by setting $T = P\lambda h$ in Inequality (11). We get that the profit made by the algorithm until that time is at least:

$$\frac{T}{4c\cdot\lambda}\log\left(1+\frac{P}{T}\cdot\lambda h\right) = \frac{P\lambda h}{4c\cdot\lambda}\log\left(1+\frac{P\lambda h}{P\lambda h}\right) = \frac{Ph}{4c}\log(2).$$

Since $OPT^{\lambda} \le P\lambda h$ we are done.

Proof of (2): We prove that the power spent over all the time slots of the sequence does not exceed P. As the algorithm stops investing power after the sequence reaches the length of $P\lambda h$, we sum the power spent over length guesses $\gamma_j = 1, 2, 4, \ldots, P\lambda h$. For each guess γ_j, the power spent is at most $\gamma_j \cdot (\frac{P}{h\cdot\gamma_j\cdot c})^{1/2}$. (As for each length guess $\gamma_j \ge 2$, a power of $(\frac{P}{h\cdot\gamma_j\cdot c})^{1/2}$ is spent only in each of the slots coming after slot number $\gamma_j/2$, and a lower power is spent in earlier slots). Thus, the power spent over all length guesses is at most

$$\begin{aligned}\sum_{j=1}^{P\lambda h}\gamma_j\cdot\left(\frac{P}{h\cdot\gamma_j\cdot c}\right)^{1/2} &= \left(\frac{P}{h\cdot c}\right)^{1/2}\sum_{j=1}^{P\lambda h}\sqrt{\gamma_j}\\ &= \left(\frac{P}{h\cdot c}\right)^{1/2}\cdot\frac{\sqrt{2}((P\lambda h)^{1/2}-1)}{\sqrt{2}-1} \qquad (13)\\ &\le \frac{\sqrt{2\lambda}\cdot P}{\sqrt{c}(\sqrt{2}-1)}.\end{aligned}$$

Equality (13) is obtained by summing up over the terms of the geometric series. Now, in order to spend at most the constrained power budget P, we require that $\frac{\sqrt{2\lambda}\cdot P}{\sqrt{c}(\sqrt{2}-1)} \le P$, and we thus set $c = \frac{2\lambda}{(\sqrt{2}-1)^2}$. ■

Combining the value of c with the proof of the first part of Lemma 5, we get the following theorem.

Theorem 4: $\mathcal{ALG}^{\lambda}$ is $O(\lambda^2)$-competitive for the online power allocation problem with bounded gain values in $[h, \lambda h]$.

We now consider the case where the power budget given to the optimal off-line algorithm is different from the power budget given to the online algorithm. Given a sequence of gain values in the range $[h, \lambda h]$, we denote by $OPT^{\lambda}(x)$ and $ALG^{\lambda}(x)$ the profit made by each of the algorithms in case the given budget is x. Assume that the budget given to $\mathcal{OPT}^{\lambda}$ is P, whereas the budget given to $\mathcal{ALG}^{\lambda}$ is Q, where $\frac{P}{Q} = \rho \ge 1$. We turn to analyze the competitive ratio for this case (that is, the ratio $\frac{OPT^{\lambda}(P)}{ALG^{\lambda}(Q)}$).

We denote by $p_i^{\mathcal{OPT}^{\lambda}(x)}$ the power invested in time slot i by $\mathcal{OPT}^{\lambda}$, given a total budget of x.

$$\begin{aligned}OPT^{\lambda}(P) &= \sum_{i=1}^{T}\log\left(1+h_i\cdot p_i^{\mathcal{OPT}^{\lambda}(P)}\right)\\ &\le \sum_{i=1}^{T}\rho\cdot\log\left(1+h_i\cdot\frac{p_i^{\mathcal{OPT}^{\lambda}(P)}}{\rho}\right) \qquad (14)\\ &\le \sum_{i=1}^{T}\rho\cdot\log\left(1+h_i\cdot p_i^{\mathcal{OPT}^{\lambda}(P/\rho)}\right) \qquad (15)\\ &= \rho\cdot OPT^{\lambda}(P/\rho) = \rho\cdot OPT^{\lambda}(Q).\end{aligned}$$

Inequality (14) follows since $\log(1+\frac{x}{\alpha}) \ge \frac{1}{\alpha}\log(1+x)$. Inequality (15) follows since the profit of $\mathcal{OPT}^{\lambda}$ given a budget of $\frac{P}{\rho}$ is at least the profit made by any other algorithm for the same sequence of slots, given the same budget. Specifically, it is at least the profit made by the algorithm that invests in each slot i a value of $p_i^{\mathcal{OPT}^{\lambda}(P)}/\rho$.

Thus, together with the competitive ratio of $4c\lambda$ achieved by $\mathcal{ALG}^{\lambda}$ (first part of Lemma 5), we get

$$OPT^{\lambda}(P) \le \rho\cdot OPT^{\lambda}(Q) \le \rho\cdot 4c\lambda\cdot ALG^{\lambda}(Q). \qquad (16)$$

Setting the value of c, we get the following corollary.

Corollary 1: Given a sequence of gain values in the range $[h, \lambda h]$, the ratio between the profits $OPT^{\lambda}(P)$ and $ALG^{\lambda}(Q)$, where $\frac{P}{Q} = \rho \ge 1$, is $O(\rho\cdot\lambda^2)$.

2) General Case: Based on the previous section we design in this section an algorithm with competitive ratio $O(\log\frac{H}{h})$ for the general online power allocation problem. Thus, our upper bound for the general case nearly matches the lower bound described in Section IV-A.

We denote by $\mathcal{OPT}$ the optimal off-line algorithm, and by $\mathcal{ALG}$ our online algorithm, for general gain sequences. The idea is the following. We partition the range of gain values $[h, H]$ into $\log\frac{H}{h}$ levels, where the jth level contains gain values in the range $[2^{j-1}h, 2^j h)$, and $j \in \{1, 2, 3, \ldots, \lceil\log\frac{H}{h}\rceil\}$. We give each level a budget of $P/(\log\frac{H}{h})$. Given an online sequence of gain values, note that the time slots with values belonging to the same level need not be consecutive slots. We refer to these slots as belonging to the same *bin*. Note that for each bin b_j, $\lambda = 2$. Our general online algorithm $\mathcal{ALG}$ works as follows. It simply runs our online algorithm for bounded range of gain values $\mathcal{ALG}^{\lambda=2}$ on each such bin independently, with budget equal to $P/(\log\frac{H}{h})$.

Given a sequence of gain values belonging to level j, we denote by $OPT_j(x)$ and $ALG_j(x)$ the profits of $\mathcal{OPT}^{\lambda=2}$ and $\mathcal{ALG}^{\lambda=2}$, respectively, for this sequence, given a budget of x. Given an arbitrary gain sequence, we denote by $OPT(P)$ and by $ALG(P)$ the profits of $\mathcal{OPT}$ and $\mathcal{ALG}$, respectively, for this sequence, given a budget of P. Now, it holds that

$$\begin{aligned}OPT(P) &\le \sum_{j=1}^{\lceil\log\frac{H}{h}\rceil} OPT_j(P)\\ &\le \sum_{j=1}^{\lceil\log\frac{H}{h}\rceil} 4c\lambda\log\frac{H}{h}\cdot ALG_j\left(\frac{P}{\log\frac{H}{h}}\right) \qquad (17)\\ &= 8c\log\frac{H}{h}\cdot ALG(P). \qquad (18)\end{aligned}$$

where $c = \frac{4}{(\sqrt{2}-1)^2}$.

Inequality (17) follows directly from inequality (16) by setting $\rho = \log\frac{H}{h}$. Equality (18) is obtained by setting $\lambda = 2$, as the ratio between the maximum and minimum gain value in each bin is at most 2. We thus get the following theorem.

Theorem 5: For a gain sequence in the range $[h, H]$, the general online power allocation problem can be solved by $\mathcal{ALG}$ with competitive ratio $O(\log\frac{H}{h})$.

V. Simulations Study

The objective of this section is to validate our suggested online algorithm for the continuous case, and to examine the effect of certain parameters on its performance. We first describe some heuristic enhancements added to the online algorithm, and explicitly specify the algorithm used in the performed experiments. Then, we turn to describe the experiments and their respective results. We use Rayleigh and Rice distributions [5] as benchmarks.

A. Heuristics Enhancements

We add several enhancements to the online algorithm for the continuous general case described in Section IV-B2. The first natural improvement is to continue investing power in the sequence of each bin, until the budget of that bin is finished (as opposed to the theoretical analysis, where the maximum considered length of a sequence is $P \cdot 2h$). In addition, we shift budget from lower to higher bins according to the following rule. Consider the case where we get a gain value belonging to the jth level, and the budget of the jth level bin is finished. Then, we try to "collect" the power that we wish to invest in the current slot from lower bins (going from bin 1 to $j-1$). The collected power (at most the needed power) is reduced from the lower bins and invested in the current slot of bin j. The algorithm used for our numerical experiments is specified below (Algorithm 1).

Algorithm 1 Online Algorithm for the General Continuous Case

Initialization: $\forall$ bins $j = 1, \ldots, \lceil \log \frac{H}{h} \rceil$ initial power $P_j = P/(log \frac{H}{h})$, current length guess $G_j = 1$ and current length $T_j = 0$.
set $P' = P/(log \frac{H}{h})$, $c = \frac{4}{(\sqrt{2}-1)^2}$.

Given a new gain value h_i:

bin level of current gain value is $j = \lfloor \log \frac{h_i}{h} \rfloor + 1$;
power to be invested in current slot is $p_i = \left(\frac{P'}{h \cdot G_j \cdot c} \right)^{1/2}$;
if ($P_j \geq p_i$) **then** ▷ there is enough power in bin j
  remaining power of bin j is $P_j = P_j - p_i$;
  current sequence length of bin j is $T_j = T_j + 1$;
else ▷ there is not enough power in bin j

  set collected power $cp_i = P_j$; ▷ take the remaining power of bin j
  for ($k = 1; k = k + 1; k < j$) **do**
   ▷ collect power from lower bins
   if ($P_k \geq (p_i - cp_i)$) **then**
    ▷ there is enough power in bin k
    remaining power of bin k is $P_k = P_k - (p_i - cp_i)$;
    update collected power $cp_i = p_i$;
    goto (23);
   else
    update collected power $cp_i = cp_i + P_k$;
    remaining power of bin k is $P_k = 0$.
   end if
  end for
  if (collected power $cp_i > 0$) **then**
   invest at current slot i a power of cp_i;
   current sequence length of bin j is $T_j = T_j + 1$;
  end if
end if
if (length T_j is equal to length guess G_j) **then**
  double the length guess $G_j = 2G_j$.
end if

B. Experimental Results

Our first goal is to examine the effect of the power budget on performance. To that end, we test our algorithm on the two following distributions: (i) Rayleigh fading, for which the gain is distributed according to an exponential distribution. In the experiments we set the average gain to 2. (ii) Rice fading with $v = 1.2$ and $\sigma = 0.534$, where v is the non-centrality parameter, and σ is the scale (this distribution yields an average gain of 2 as well). For both cases, we set $h_{\min}$ to 0.1, and the value $h_{\max}$ to 9.2. The probability of obtaining a higher value than the selected $h_{\max}$ is less than one percent under both distributions. Hence, $h_{\max}$ is an effective upper bound for dividing the power budget to bins (accordingly, gains higher than $h_{\max}$ use the power budget from the last bin).

The results for Rayleigh fading and Rice fading are depicted in Figures 1 and 2, respectively. For both cases, observe that the performance ratio improves with the power budget. The intuitive explanation for this phenomenon is that when the budget is limited, every "mistake" of the online algorithm is costly, as it leads to a significant reduction in the bin's budget. It is also worth noticing that the performance ratio with large budgets is almost three times better than the worst case bound of $\log(h_{\max}/h_{\min}) = 6.52$.

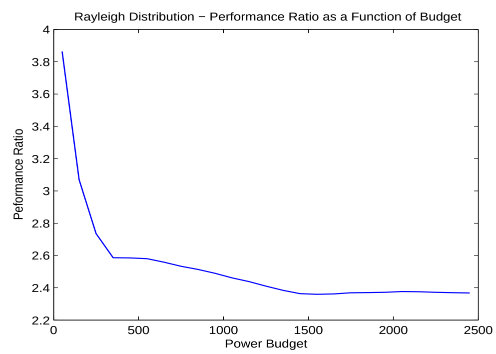

Fig. 1. Rayleigh fading. The ratio between the optimal (offline) power allocation and the online allocation, as a function of the power budget. Results are averaged over 10 runs.

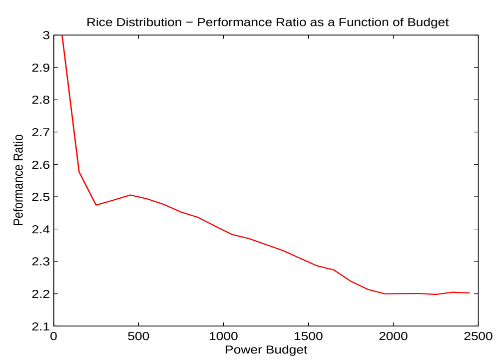

Fig. 2. Rice fading. The ratio between the optimal (offline) power allocation and the online allocation, as a function of the power budget. Results are averaged over 10 runs.

Our second experiment focuses on the effect of $h_{\min}$ on performance. Specifically, we are interested in examining the influence of $h_{\min}$ on the throughput ratio between the optimal (off-line) and online algorithms. Accordingly, we fix the underlying channel distribution (Rayleigh, with parameters as above) and the power budget (1000), and increase $h_{\min}$ by ignoring a larger percentage of the channel gains. It is seen in Figure 3 that the performance bound improves with $h_{\min}$. This result is consistent with the theoretical performance bound, which is inversely proportional to $h_{\min}$.

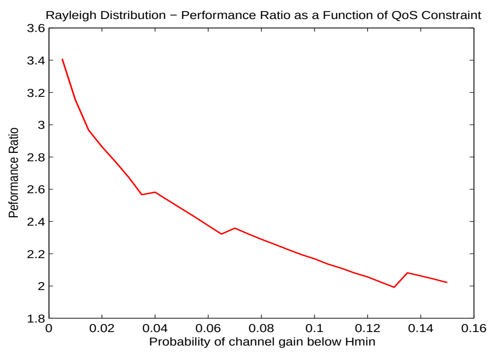

Fig. 3. Rayleigh fading. The ratio between the optimal (offline) power allocation and the online allocation, as a function of $h_{\min}$. Results are averaged over 10 runs. Note that the "glitches" in the graph are a consequence of the change in the number of bins.

Summarizing our results, we have observed that the online algorithm performs significantly better than the theoretical guarantees. In addition, it seems that performance is directly affected by the available power and the $h_{\min}$ constraint. We have chosen to simulate channel processes that are generated according to a stationary probability rule. This choice is made mainly for its simplicity of implementation, and it is expected that a similar characterization would be valid under other distributions, as well as for arbitrary varying channels.

VI. Conclusions

We considered the problem of power allocation under dynamic channel quality, within the framework of online computation. We addressed both a "discrete" case, where the transmitter can transmit only at a fixed power level, and a "continuous" case, where the transmitter can choose the power level out of a continuous interval. For both cases, we proposed online power-allocation algorithms with proven worst-case performance bounds. In addition, we established lower bounds on the online power allocation problem in both cases, hence benchmarking our solutions. To the best of our knowledge, this is the first study that proposes to attack this problem through the methodology of online (competitive) analysis. We complemented our work with a simulation study, where we validated our suggested online algorithm for the continuous case, and observed that our online algorithm performs significantly better than the theoretical bound.

Our framework can be extended in several ways. Our online approach provides an algorithmic solution to the case where the channel gain process is arbitrarily varying. In general, however, there could possibly be a probability rule that governs certain characteristics of the process (e.g., the gain corresponds to an arbitrary element times a random variable). An interesting research direction is therefore to consider the case where some probability rule partially governs the channel statistics. Our suggested algorithms can obviously be enhanced by incorporating (or estimating) the known characteristics. In this context, one may consider additional approaches for dealing with parameter uncertainty, such as robust optimization [2].

At a higher level, a challenging future direction would be to consider the multiuser network case, where each user observes a private (possibly correlated) channel gain and adapts its current power level accordingly. The difficulty in this case arises from the fact that the powers of neighboring users affect the throughput of each user. Unlike the channel state, the power allocation of other users is unknown prior to the individual power adaptation. The algorithms for the multiuser case, as well as the analysis thereof (possibly in the framework of dynamic noncooperative games), may require novel methods and solution concepts.

References

[1] E. Altman, K. Avrachenkov and A. Garnaev, "A jamming game in wireless networks with transmission cost", *Network Control and Optimization, First EuroFGI International Conference, NET-COOP*, pp. 1–12, 2007.

[2] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Math. Oper. Res.*, vol. 23, no. 2, pp. 769-805, 2000.

[3] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*. Cambridge University Press, 1998.

[4] A. Borodin, R. El-Yaniv and V. Gogan. On the competitive theory and practice of portfolio selection *Proc. of the 4th Latin American Symposium on Theoretical Informatics*, pp. 173–196, 2000.

[5] E. Biglieri, J. Proakis, and S. Shamai, "Fading channels: information-theoretic and communications aspects," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2619–2692, 1998.

[6] N. Buchbinder and J. Naor. Online primal-dual algorithms for covering and packing problems. *Proc. of the 13th Annual European Symposium on Algorithms*, pp. 689–701, 2005.

[7] N. Buchbinder, L. Lewin-Eytan, I. Menache, J. Naor and A. Orda, "Dynamic Power Allocation Under Arbitrary Varying Channels – An Online Approach" (full version). Available from http://www.ee.technion.ac.il/Sites/People/ArielOrda/Info/Other/BLMNO08.pdf.

[8] A. Goldsmith and P. P. Varaiya, "Capacity of fading channels with channel side information," *IEEE Transactions on Information Theory*, vol. 43, no. 6, 1997.

[9] B. Hughes and P. Narayan, "Gaussian arbitrarily varying channels," *IEEE Transactions on Information Theory*, vol. 33, no. 2, pp. 267-284, 1987.

[10] A. Lapidoth and P. Narayan, "Reliable communication under channel uncertainty," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2148–2177, 1998.

[11] I. Menache and E. Altman, "Battery-state dependent power control as a dynamic game," *WiOpt*, 2008.

[12] Y. Zhou, D. Chakrabarty and R. Lukose. Budget constrained bidding in keyword auctions and online knapsack problems, *WWW*, pp. 1243-1244, 2008.

Acknowledgements

The authors would like to thank Michael Katz and Amichai Sanderovich for helpful discussions and comments on this work.