Fast Algorithms for Resource Allocation in Wireless Cellular Networks

Ritesh Madan, Stephen P. Boyd, *Fellow, IEEE*, and Sanjay Lall, *Senior Member, IEEE*

***Abstract*—We consider a scheduled orthogonal frequency division multiplexed (OFDM) wireless cellular network where the channels from the base-station to the n mobile users undergo flat fading. Spectral resources are to be divided among the users in order to maximize total user utility. We show that this problem can be cast as a nonlinear convex optimization problem, and describe an $O(n)$ algorithm to solve it. Computational experiments show that the algorithm typically converges in around 25 iterations, where each iteration has a cost that is $O(n)$, with a modest constant. When the algorithm starts from an initial resource allocation that is close to optimal, convergence typically takes even fewer iterations. Thus, the algorithm can efficiently track the optimal resource allocation as the channel conditions change due to fading. We also show how our techniques can be extended to solve resource allocation problems that arise in wideband networks with frequency selective fading and when the utility of a user is also a function of the resource allocations in the past.**

***Index Terms*—Fast computation, resource allocation, scheduling, wireless cellular networks.**

I. INTRODUCTION

RESOURCE allocation in wireless networks is fundamentally different than that in wireline networks due to the time-varying nature of the wireless channel [1]. There has been much prior work on scheduling policies in wireless networks to allocate resources among different flows based on the channels they see and the flow state [1], [2]. The flow state can consist of the average rate seen by the flow in the past [3], [4], the delay of the head-of-line packet [5], or the length of the queue [6]. Much prior work in this area can be divided into two categories:

Manuscript received September 16, 2009; revised September 28, 2009; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S.Borst. First published November 24, 2009; current version published June 16, 2010. This work was funded in part by the MARCO Focus Center for Circuit and System Solutions (C2S2, www.c2s2.org) under Contract 2003-CT-888, the AFOSR under Grant AF F49620-01-1-0365, the NSF under Grant ECS-0423905, the NSF under Grant 0529426, the DARPA/MIT under Grant 5710001848, the AFOSR under Grant FA9550-06-1-0514, the DARPA/Lockheed under Contract N66001-06-C-2021, the AFOSR/Vanderbilt under Grant FA9550-06-1-0312, the Stanford URI Architecture for Secure and Robust Distributed Infrastructures (AFOSR DoD award 49620-01-1-0365), and the Sequoia Capital Stanford Graduate Fellowship.

R. Madan was with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA. He is now with Qualcomm-Flarion Technologies, Bridgewater, NJ 08870 USA (e-mail: rkmadan@stanfordalumni.org).

S. P. Boyd is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: boyd@stanford.edu).

S. Lall is with the Department of Electrical Engineering and the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305 USA (e-mail: lall@stanford.edu).

Digital Object Identifier 10.1109/TNET.2009.2034850

1) *Scheduling for elastic (non real-time) flows:* The end-user experience for a elastic flow is modeled by a concave increasing utility function of the rate experienced by the flow [7]. The proportional fair algorithm (see, for example, [8]) where all the resources are allocated to the flow with the maximum ratio of instantaneous spectral efficiency (which depends on the channel gain) to the average rate has been analyzed in [3], [9], [10]; roughly speaking this algorithm maximizes the sum of log utilities of average rates over an asymptotically large time horizon. A more general scheduling rule where potentially multiple users can be scheduled simultaneously has been considered in [11], [12]. Most of the above work assumes that the queues have infinite backlogs, i.e., packets are always available in the buffers of all the queues; extensions to finite queues are provided in, for example, [3]. Joint design of scheduling and congestion control with modeling of queue dynamics has been considered in, for example, [4], [13]–[15]; in this case, packets are always assumed to be available at the congestion controller.
2) *Scheduling for Real-Time Flows:* Real-time flows are typically modeled by a predetermined but unknown arrival process and a delay deadline for each packet. For such flows, we can roughly define the *stability region* as follows: The stability region for a set of queues is defined as the set of arrival rates at the queues for which there exists a scheduling policy such that the length of any queue does not grow without bound over time (see, for example, [16]). A *stabilizing policy* is one which ensures that the queue lengths do not grow without bound. Stabilizing policies for a vector of arrival rates within the stability region for different wireless network models have been characterized in, for example, [5], [6], [16]–[19]. The scheduling policy in [5] minimizes the percentage of packets lost because of deadline expiry, while the delay performance of the *exponential rule* (introduced in [6]) was empirically studied in [20]. Work on providing throughput guarantees for such flows includes [21] and [22], and references therein.

We note that policies to schedule a mixture of elastic (non real-time) and real-time flows have been considered in [20]. Distributed algorithms for interference management to maximize the sum utilities of user signal-to-noise ratios (SNR) in cellular networks have been studied in [23], [24]. Also, related cross-layer optimization problems for resource allocation in wireless networks with different objectives have been analyzed in, for example, [25]–[27]. Resource allocation algorithms which focus on maximizing sum rate (without fairness or with minimum rate guarantees) for OFDM systems include [28]–[32]. The above summary is only a representative sample of the work in the general area of resource allocation in wireless networks. For a more complete description of prior work, we refer the reader to [2], [6], and the references therein.

In this paper, we focus on elastic flows with infinite backlogs; an extension to model constraints of finite backlogs due to congestion control (which can be modeled as an upper bound on the bandwidth allocated to a user) is straightforward. We study the problem of resource allocation in wideband OFDM wireless cellular networks like Ultra Mobile Broadband (UMB) [33] and Long Term Evolution path for 3GPP [34]. In particular, we study the assignment of power and spectral resources to maximize the sum-utility of the achieved data rates. The user utility can be a function of instantaneous rate or average rate over time. For both these cases, the solution in general can result in the distribution of resources to *multiple flows* at the same time. We show that the problem is a convex optimization problem. Hence, it can be solved in $O(n^3)$ time for n users using a general-purpose barrier method (see, for example, [35]). However, the time-varying nature of wireless channels necessitates recomputation of an optimal resource allocation in an online manner. This requires the design of faster computational algorithms to track the optimal resource allocation. We exploit the underlying structure of the problem to derive a specialized barrier method that has a complexity of $O(n)$. We also illustrate the generality of our computational techniques through extensions to frequency selective fading, where we exploit frequency diversity.

We note that our work focusses on computational algorithms and is complementary to that in [3], [9]–[11]. The focus of those papers is on the asymptotic analysis when the user utility is a function of the rate averaged over a very long time.

A. Organization

The rest of the paper is organized as follows. We first consider the utility for each flow to be a function of the instantaneous rate. We describe the mathematical model and problem formulation, and prove the existence of a unique positive solution in Section II. We exploit the structure of the underlying optimization problem to obtain an $O(n)$ algorithm and illustrate its typical behavior through computational results in Section III. In Sections IV and V, we consider frequency selective fading and the case where the utility of a user is a function of its average rate, respectively. In Section VI, we compare our algorithm with other standard computational approaches.

II. Problem Formulation

A. System Model

We model an OFDM wireless cellular network where spectrum and power need to be divided between communication flows (users) on n links in a cell. We formulate an optimization problem which is applicable to the downlink; as we show later, extensions to the uplink can be similarly obtained. We assume an M-Quadrature Amplitude Modulation (MQAM) scheme for transmission and a total system bandwidth, B. Then, the maximum rate (in nats/sec) at which a user, i, can transmit is given by

$$R_i = B_i \log\left(1 + \frac{KP_iG_i}{N_0B_i}\right)$$

where P_i is the transmit power, G_i is the channel gain over the link to user i, B_i is the bandwidth allocated to user i, N_0 is the noise power spectral density, and $K = -1.5/\log(5\mathrm{BER})$, where BER is the desired (constant) bit error rate [36].

We denote the effective flow rate in nats/s/Hz for user i by $r_i = R_i/B \geq 0$, and the fraction of bandwidth allocated to it by $b_i \geq 0$. We denote the associated vectors of rates and bandwidth-fractions as $r \in \mathbb{R}^n$ and $b \in \mathbb{R}^n$, respectively. The power consumption to support flow rate $r_i > 0$ can be modeled as

$$p_i(r_i, b_i) = a_i b_i\left(e^{r_i/b_i} - 1\right), \qquad a_i = N_0B/(G_iK).$$

When $r_i = 0$, the power required is 0. The power consumption of user i as a function of r_i and b_i has the form $a_i f(r_i, b_i)$, where the function $f: S \to \mathbb{R}$ is defined as follows:

$$f(x,y) = \begin{cases} y(e^{x/y} - 1), & \text{if } y > 0, \\ 0, & \text{otherwise.} \end{cases}$$

The set $S \subset \mathbb{R}^2$ is given by

$$S = \{0\} \cup \{\,(x,y) \in \mathbb{R}^2 \mid x \geq 0,\ y > 0\,\}.$$

We assume that each cell has a (weighted) total power constraint of the form

$$P(r,b) = \sum_{i=1}^{n} w_i a_i f_i(r_i, b_i) \leq P_{\max}$$

where $P(r,b)$ is the (weighted) total power, $P_{\max} > 0$ is the given maximum (weighted) total power, and $w_i > 0$ are the weights. This constraint can be used to model a sum-power constraint, with $w_i = 1$, for the downlink in a cell. For the uplink, it can also be used to model the requirement that the total interference at a neighboring interfering base-station should be kept below some threshold.[1] The weights then represent the power gains to the neighboring base-station.[2] We will normalize the power constraint by defining the normalized power $p: S^n \to \mathbb{R}$ by $p(r,b) = \sum_{i=1}^{n} c_i f(r_i, b_i)$ where $c_i = w_i a_i / P_{\max}$. The power constraint is then $p(r,b) \leq 1$.

We first observe that p_i is a convex function of r_i and b_i. The function $g(x,y) = ye^{x/y}$, defined for $y > 0$, is the perspective of the exponential function, and so is convex in x and y (see, e.g., [35, Sec. 3.2.6]). The function p_i is obtained from g by an affine composition, and the addition of a linear term, and so is convex. The total power $p(r,b)$ is therefore also a convex function of r

[1]In the uplink, some mobiles may be power limited and so, it is necessary to model the individual power constraint for each link. Since we mainly focus on the downlink for the rest of the paper, we do not include this in our analysis for notational simplicity – our techniques can be generalized in a straightforward manner to allow for such constraints as well.

[2]In general we can have a total interference budget constraint at more than one base-station – our analysis extends to this case as well. Also, a total interference budget constraint is a reasonable way to keep interference low at neighboring base-stations when the frequency tones in neighboring cells hop randomly and independently of each other [8]. Setting the interference budgets is out of the scope of our paper. For the uplink, N_0 now represents the noise plus average interference power spectral density.

and b, and so, the total power constraint is a convex constraint for $r, b > 0$.

B. User Utility Functions

The utility for user i is a function of its instantaneous rate, given by $U_i(r_i)$, so the total utility is

$$U(r) = \sum_{i=1}^{n} U_i(r_i).$$

We assume that the utility functions $U_i\colon (0, \infty) \to \mathbb{R}$ are *thrice* continuously differentiable with

$$U_i'(x) > 0, \quad U_i''(x) < 0$$

for all $x > 0$ and

$$\lim_{x \to 0^+} U_i'(x) = \infty.$$

Thus, U_i (and therefore also U) is strictly increasing and strictly concave, and the marginal utility increases without bound as the rate converges to zero. Examples of common utility functions satisfying these conditions include $\log x$ and x^a, for $0 < a < 1$.

Note that the above utility function does not take into account past allocations to users. We consider this extension in Section V. We show that we can use our computational techniques to efficiently compute a scheduling policy that is a generalization of the scheduling policy in [3].

C. Maximum Utility Resource Allocation

Our goal is to choose r and b to maximize the total utility, subject to the power constraint and the bandwidth-fraction constraint:

$$\begin{array}{ll} \text{maximize} & U(r), \\ \text{subject to} & \mathbf{1}^T b = 1 \\ & r > 0, \quad b > 0, \\ & p(r, b) \leq 1 \end{array} \tag{1}$$

where $\mathbf{1}$ denotes the vector with all entries one. The optimization variables are r_i and b_i; the problem data are c_i and the functions U_i. The vector inequalities are componentwise; $r \geq 0$ means $r_i \geq 0$, $i = 1, \ldots, n$. For convenience we will define the feasible set D by

$$D = \left\{ (r, b) \in \mathbb{R}^{2n} \mid \mathbf{1}^T b = 1,\ p(r, b) \leq 1,\ r > 0,\ b > 0 \right\}.$$

We now have the equivalent problem

$$\begin{array}{ll} \text{maximize} & U(r), \\ \text{subject to} & (r, b) \in D. \end{array} \tag{2}$$

In the following section, we will show that there is a unique optimal allocation (r, b) which is achieved at a point with $r > 0$ and $b > 0$. Hence, relaxing these strict inequalities to nonstrict inequalities, and appropriately interpreting p and U, does not change the optimal solution.

The resource allocation problem (2) is a convex optimization problem, with $2n$ variables and $2n + 2$ constraints. Roughly speaking, this means that its global solution can be efficiently computed, for example by a general interior-point method. These methods typically converge in a few tens of iterations; each iteration in a general-purpose implementation requires $O(n^3)$ arithmetic operations (see, e.g., [35, Ch. 11] or [37]). The algorithm we describe in the next section solves the resource allocation problem much faster by exploiting its special structure. The resulting interior point method converges in about 25 to 30 iterations, where each iteration requires $O(n)$ operations with a modest constant.

D. Existence and Uniqueness of a Positive Solution

In this section, we show that the resource allocation problem (1) has a unique solution $(r^\star, b^\star)$, with $r^\star > 0$ and $b^\star > 0$. We will do this by constructing a sequence of points converging to the maximum, which must therefore lie in the closure of the feasible set. We first show the following. (The proofs of the next three lemmas have been moved to the Appendix .)

Lemma 1: The closure of D satisfies $\bar{D} \subset S^n$.

The interpretation of this result is that allocating zero bandwidth-fraction and positive rate to a user requires infinite power. Hence for every point (r, b) in the feasible set, we must have $b_i > 0$ whenever $r_i > 0$, and in fact this holds for the closure of the feasible set also.

The next result shows that a point (r, b) with $(r_i, b_i) = (0, 0)$ for some i cannot be optimal. The idea here is that since U_i has infinite slope at 0, slightly increasing r_i and b_i will give an increase in utility U_i which outweighs the decrease in the other rates necessary to maintain the power constraint.

Lemma 2: Suppose (r^k, b^k) is a sequence in S^n with limit

$$\lim_{k \to \infty} (r^k, b^k) = (r, b)$$

and $(r, b) \in S^n$, with $\mathbf{1}^T b = 1$ and $p(r, b) \leq 1$. Suppose also that for all $i = 1, \ldots, n$ either $r_i > 0$ or $(r_i, b_i) = (0, 0)$. If there is some i such that $(r_i, b_i) = (0, 0)$ then there exists $(x, y) \in D$ such that

$$\lim_{k \to \infty} U(r^k) < U(x).$$

The final lemma needed shows that a point (r, b) with $r_i = 0$ for some i must also have $b_i = 0$. If this were not the case, we could decrease b_i to zero, spreading this bandwidth-fraction among the other users, who can use the extra bandwidth-fraction to increase their rates without increasing their powers, thus giving a feasible point with larger total utility. Then using Lemma 2, we can rule out the possibility that a maximizing sequence converges to $(r, b) = 0$.

Lemma 3: Suppose (r^k, b^k) is a sequence in S^n with limit

$$\lim_{k \to \infty} (r^k, b^k) = (r, b)$$

and $(r, b) \in S^n$, with $\mathbf{1}^T b = 1$ and $p(r, b) \leq 1$. If there is some i such that $r_i = 0, b_i > 0$, then there exists $(x, y) \in D$ such that

$$\lim_{k \to \infty} U(r^k) < U(x).$$

We now have the following theorem showing the existence and uniqueness of the solution.

Theorem 1: There exists a unique $(r^\star, b^\star) \in D$ with $r^\star, b^\star > 0$ such that

$$U(r^\star, b^\star) = \sup\left\{ U(r) \mid (r, b) \in D \right\}.$$

Proof: First notice that problem (2) is feasible. That is, the set D is nonempty, since for small enough $\epsilon > 0$ the choice $b = (1/n)\mathbf{1}$, $r = \epsilon\mathbf{1}$ satisfies $(r, b) \in D$. Let

$$U^\star = \sup\{\, U(r) \mid (r, b) \in D \,\}.$$

Then $U^\star$ is finite, since D is bounded and U is concave. We must show that this optimal value is actually achieved. Suppose (r^k, b^k) is a maximizing sequence in D, so that $U(r^k, b^k) \to U^\star$. By extracting a subsequence, we can assume that (r^k, b^k) converges to a point $(\bar{r}, \bar{b}) \in \bar{D}$. Lemma 1 implies this point lies in S^n and since it is optimal on $\bar{D}$ Lemma 3 implies that $\bar{r} > 0$ and $\bar{b} > 0$. Hence the optimal value is achieved in D. Uniqueness now follows from strict concavity of U. ■

III. Fast Online Resource Allocation Algorithm

In this section, we describe the barrier method to compute an optimal resource allocation. Such a method, in general, has complexity $O(n^3)$. However, we exploit the structure of the problem to reduce the complexity to $O(n)$.

A. Barrier Method

We use the barrier method to solve the optimization problem in (2) [35]. The central point $(r^\star(t), b^\star(t))$ for a given value of the barrier parameter t is given by the solution of the following problem:

$$\begin{aligned} \text{minimize} \quad & -tU(r) - \sum_{i=1}^{n} (\log r_i + \log b_i) \\ & -\log(1 - p(r, b)), \\ \text{subject to} \quad & \mathbf{1}^T b = 1. \end{aligned} \tag{3}$$

As t increases, $(r^\star(t), b^\star(t))$ becomes a more accurate approximation to the solution to the problem in (2). Note that the objective function above is convex, and the above problem is a convex optimization problem. Moreover, the solution to the above problem is unique. This follows, in particular, from the positive-definiteness of the Hessian of the objective function, as argued in Section III-C.

We collect the variables into one vector $x \in \mathbb{R}^{2n}$, $x = (r_1, b_1, \ldots, r_n, b_n)$. Note that we have interleaved the rate and bandwidth-fraction variables here, so that the variables associated with a given user are adjacent. Also, we denote the barrier function as

$$\phi(x) = -\sum_{i=1}^{n} (\log r_i + \log b_i) - \log(1 - p(r, b))$$

and

$$\psi_t(x) = -tU(r) - \phi(x).$$

The barrier method is then as follows.

Given strictly feasible starting point x, $t := t^{(0)}$, $\mu > 1$, tolerance ϵ.
Repeat
1) *Centering Step.* Minimize $\psi_t(x)$ subject to $\mathbf{1}^T b - 1 = 0$, starting at x.
2) *Update.* $x := x^\star(t)$.
3) *Stopping Criterion.* **quit** if $(2n + 1)/t < \epsilon$.
4) *Increase t.* $t := \mu t$.

B. Newton Method

We now describe the Newton method to compute the central point $x(t)$, i.e., solve the problem in (3) for a given value of t. The Newton step Δx at x, and the associated dual variable are given by following equations:

$$\begin{bmatrix} \nabla^2 \psi_t(x) & d \\ d^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \nu \end{bmatrix} = \begin{bmatrix} t\nabla U(r) - \nabla \phi_t(x) \\ 0 \end{bmatrix} \tag{4}$$

where $d = [0\ 1\ \cdots\ 0\ 1]^T$. For the Newton method, we use a backtracking line search to ensure an adequate decrease in ϕ (see, *e.g.*, [35, Ch.11] or [38]). The method is then as follows.

Given starting point x such that $\mathbf{1}^T b = 1$, tolerance ϵ, $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$.
Repeat
1) Compute Δx and $\lambda^2 := -\nabla \psi_t(x) \Delta x$.
2) *Stopping Criterion.* **quit** if $\lambda^2/2 \le \epsilon$
3) Backtracking line search on $\psi_t(x)$. $s := 1$.
while $\psi_t(x + s\Delta x) > \psi_t(x) - \alpha s \lambda^2$, $s := \beta s$.
4) *Update.* $x := x + s\Delta x$.

C. Fast Computation of Newton Step

We now describe how we can exploit the structure of the problem to compute the Newton step in $O(n)$ time rather than using matrix inversion in (4) which has a cost of $O(n^3)$. The gradient of the barrier function is given by

$$\begin{aligned} \frac{\partial \phi(x)}{\partial r_i} &= -\frac{1}{r_i} + \frac{c_i e^{r_i/b_i}}{1 - p(r, b)} \\ \frac{\partial \phi(x)}{\partial b_i} &= -\frac{1}{b_i} + \frac{c_i e^{r_i/b_i}(1 - 1/b_i) - c_i}{1 - p(r, b)}. \end{aligned}$$

The Hessian of the barrier function is given by

$$\nabla^2 \phi(x) = \begin{bmatrix} 1/r_1^2 & & & & \\ & 1/b_1^2 & & & \\ & & \ddots & & \\ & & & 1/r_n^2 & \\ & & & & 1/b_n^2 \end{bmatrix} + \frac{1}{(1 - p(r, b))^2} \nabla p(r, b) \nabla p(r, b)^T + \frac{1}{1 - p(r, b)} \nabla^2 p(r, b).$$

Hence, it follows that

$$\begin{aligned} \nabla^2 \psi_t(x) &= -t\nabla^2 U(r) + \nabla^2 \phi(x) \\ &= \frac{1}{(1 - p(r, b))^2} \nabla p(r, b) \nabla p(r, b)^T \\ &\quad + \begin{bmatrix} H_1 & & & \\ & H_2 & & \\ & & \ddots & \\ & & & H_n \end{bmatrix} \end{aligned}$$

where the blocks not shown are all zero, and

$$H_i = \begin{bmatrix} -tU_i''(r_i) + 1/r_i^2 & 0 \\ 0 & 1/b_i^2 \end{bmatrix} + \frac{1}{1-p(r,b)} \begin{bmatrix} e^{r_i/b_i}c_i/b_i & -e^{r_i/b_i}c_i r_i/b_i^2 \\ -e^{r_i/b_i}c_i r_i/b_i^2 & e^{r_i/b_i}c_i r_i^2/b_i^3 \end{bmatrix}.$$

The gradient, $\nabla p(r,b)$, of $p(r,b)$ is given by

$$\frac{\partial p(r,b)}{\partial r_i} = c_i e^{r_i/b_i}$$
$$\frac{\partial p(r,b)}{\partial b_i} = c_i e^{r_i/b_i}(1 - 1/b_i) - c_i.$$

Let us denote

$$g = \frac{1}{(1-p(r,b))}\nabla p(r,b)$$
$$h = t\nabla U(r) - \nabla \phi_t(x).$$

Then we have

$$\nabla^2 \psi_t(x) = \begin{bmatrix} H_1 & & & \\ & H_2 & & \\ & & \ddots & \\ & & & H_n \end{bmatrix} + gg^T.$$

It is easy to show that $H_i > 0$. Since $gg^T \geq 0$, it follows that $\nabla^2\psi_t(x) > 0$. Since d is a nonzero vector, it follows that the KKT matrix on the left in (4) is invertible. Also, the KKT matrix on the left in (4) is the sum of a *block-arrow* matrix and a *rank-one* matrix. We exploit this structure to compute the Newton step in $O(n)$ time. Let us denote $H = \operatorname{diag}(H_1, \ldots, H_n)$. In particular, we have (see, for example, [35, Appendix C])

$$\begin{bmatrix} \Delta x \\ \nu \end{bmatrix} = u - \frac{[g^T\ 0]u}{1 + [g^T\ 0]v} v$$

where

$$\begin{bmatrix} H & d \\ d^T & 0 \end{bmatrix} u = \begin{bmatrix} h \\ 0 \end{bmatrix} \tag{5}$$

and

$$\begin{bmatrix} H & d \\ d^T & 0 \end{bmatrix} v = \begin{bmatrix} g \\ 0 \end{bmatrix}.$$

We now obtain analytical formulas for u and v, which can be computed in $O(n)$ time. We consider the computation of u in detail; the computation for v is identical. It follows from (5) that

$$\begin{bmatrix} u_{2i-1} \\ u_{2i} \end{bmatrix} = H_i^{-1} \begin{bmatrix} h_{2i-1} \\ h_{2i} - u_{2n+1} \end{bmatrix}.$$

Substituting these back in (5), it follows that

$$u_{2n+1} = \frac{1}{\sum_{i=1}^n H_{i_{2,2}}^{-1}} \sum_{i=1}^n (H_{i_{2,1}}^{-1} h_{2i-1} + H_{i_{2,2}}^{-1} h_{2i}).$$

To compute u, we first obtain u_{2n+1}, and then obtain the other u_i's. Both these operations cost $O(n)$.

D. Convergence Analysis

We now prove the convergence of the Newton method for this problem for a given t. The convergence of the barrier method then follows. Consider the minimization of $\psi_t(x)$. Define the set of iterates for the Newton method by $L = L(x^{(0)})$, where the initial point $(x^{(0)})$ is chosen to be strictly feasible. For the initial value of t, such a point is easy to find by allocating equal bandwidth fractions, and powers to users such that the total power is less than 1, i.e., $p(r^{(0)}, b^{(0)}) < 1$; for other iterations of the barrier method, the solution for the previous value of t is guaranteed to be strictly feasible. The Newton method is a descent method, i.e., $\psi_t(x^{(k)}) \leq \psi_t(x^{(0)})$, for any iteration k.

We first consider the following two lemmas, the proofs of which have been moved to the Appendix .

Lemma 4: For all iterations k of the Newton method, $x^{(k)}$ is strictly feasible.

Now, it can be shown that the iterates belong to a closed and bounded set.

Lemma 5: The set $L \subset \bar{L}$, where for any $(r,b) \in \bar{L}$, r_i, b_is are bounded above and bounded away from zero.

Since the KKT matrix on the left in (4) is invertible, and is a continuous function of (r,b), it follows that its inverse is bounded on the closed set $\bar{L}$. Also, $\nabla^2\psi_t$ is a continuously differentiable function of (r,b) and hence, $\nabla^2\psi_t$ is Lipschitz continuous on $\bar{L}$, and $\|\nabla^2\psi_t\|$ is bounded above on $\bar{L}$. The convergence of the Newton method then follows (see, for example, [35, Ch. 10]).

A formal complexity analysis (i.e., a bound on the number of Newton steps required to attain an accurate solution) can be carried out, but this seems irrelevant to us, given the extremely fast convergence of the algorithm in practice. A typical number of steps required is 25, and often less.

E. Warm Start

The Newton method can be initialized with $b = (1/n)\mathbf{1}$, and $r = \epsilon\mathbf{1}$, where $\epsilon > 0$ such that (r,b) is strictly feasible, i.e., $p(r,b) < 1$. It can also be initialized with an approximate solution, such as the solution of a resource allocation problem that is 'close'. Consider, for example, the situation where we have computed the optimal resource allocation, and then the problem changes, but not drastically; for example, the utility functions change, or the channel parameters a_i change, or the maximum available power $P_{\max}$ changes. Running the barrier method starting from the previously computed optimal point and a larger value of t typically cuts the number of iterations required to 10 to 15. This can be repeated, in order to efficiently track the optimal resource allocation as the physical parameters or requirements change.

F. Numerical Results

In this section, we show the typical behavior of the algorithm described in this paper. We consider a system of $n = 200$ users in a cell. The utility function for user i is taken to be $U_i(r_i) = k_i \log r_i$, where k_i are generated as independent uniform random variables on $[1, 10]$. We take $w_i = 1$, i.e., we model the sum-power constraint for the downlink.

We first study the convergence of our algorithm for randomly generated c_i's. In particular, we consider each c_i to be randomly

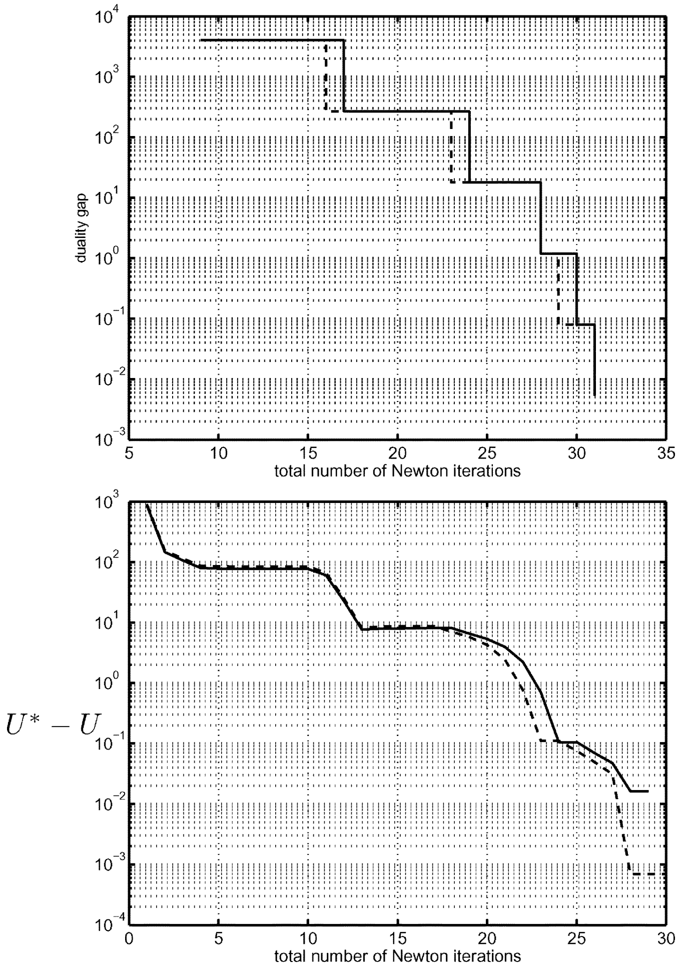

Fig. 1. Typical convergence of the barrier method. *Top.* Norm of residual versus iteration for two different instances. *Bottom.* Convergence of $U^\star - U$ versus iteration.

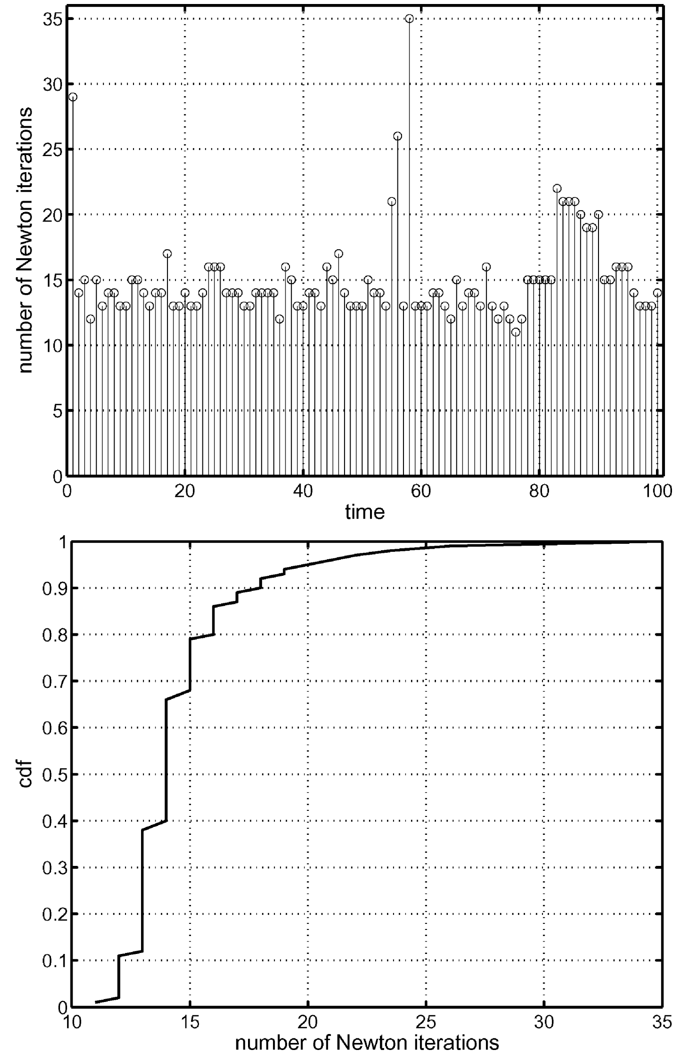

Fig. 2. Number of Newton iterations needed for reconvergence with Rayleigh fading channels. *(Top)* Number of Newton iterations for reconvergence during the first 100 time-steps. *(Bottom)* CDF of number of Newton iterations for reconvergence over 500 time-steps.

distributed over $[0.1, 5]$, i.e., the received signal to noise ratio (SNR) at the mobile can vary over the large range of -9.6 to 20 dB. Fig. 1 (top) shows the convergence of the norm of the residual, versus cumulative Newton iteration, for two different instances of the problem. The bottom plot shows the convergence of the utility to its optimal value; note that all intermediate iterates are feasible. This plot shows that the resource allocation obtained is close to optimal, from a practical point of view, within 20 or so Newton iterations. Highly accurate solutions can be obtained in about 30 iterations or so. Both plots are quite typical; similar results are obtained as n and other problem parameters are varied.

To illustrate warm-start methods, we simulated a wireless network with time-varying fading channels. The resulting scheduling policy obtained by solving (1) has the following properties. Users with a higher average channel gain get more resources on average. Users get allocated more resources when their instantaneous channel gain is relatively high than when their instantaneous channel gain is low. In our simulation, each user's channel undergoes mutually independent Rayleigh fading with a Doppler frequency of 5 Hz and mean SNR of 0 dB. Thus, the channel completely decorrelates after 200 time-steps or so. We recomputed the optimal resource allocation at every time step of 1 ms. Also, the variation in channel gains over time is very high; the channel can easily swing over a range of 30 dB.

Fig. 2 shows the number of Newton steps required to reconverge to a very accurate optimal resource allocation, starting from the previously computed one. The first computation (from a generic initial resource allocation) requires 29 cumulative Newton steps. For the rest of the time-steps we used a larger value of $t^{(0)}$ such that only two centering steps were required for a guaranteed duality gap of less than 10^{-3}. About 80% of the time, the number of Newton iterations required for reconvergence is less than 15. A larger number of Newton iterations is occasionally required at times when the rate of change of the channel is high; for example during deep fades.

IV. Frequency Selective Fading

In this section, we describe an extension to the case where there are m frequency bands such that over a given frequency band, each user's channel undergoes flat fading. For example, it is sufficient to choose the bandwidth of each band to be less than the minimum coherence bandwidth of the users [39]. Denote by G_i^j, the channel gain on the jth frequency band for user i. Similarly, denote the rate and bandwidth for user i on the jth frequency band by r_i^j and b_i^j, respectively. Then the total rate allocated to user i is

$$r_i = \sum_{j=1}^{m} r_i^j.$$

Also, the total (weighted) power consumption is given by

$$p(r,b) = \sum_{i=1}^{n}\sum_{j=1}^{m} c_i^j f(r_i^j, b_i^j)$$

where $c_i^j = w_i N_0 B/(G_i^j K)$.

We again would like to compute a resource allocation to maximize the total utility, i.e., solve the following optimization problem:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} U_i\left(\sum_{j=1}^{m} r_i^j\right) \\ \text{subject to} & \mathbf{1}^T b^j = 1,\ j = 1,\ldots,m, \\ & \sum_{j=1}^{m} r_i^j > 0,\ i = 1,\ldots,n, \\ & (r_i^j, b_i^j) \in S,\ i=1,\ldots,n,\ j=1,\ldots,m, \\ & p(r,b) \le 1 \end{array} \tag{6}$$

where r^j and b^j are in $\mathbb{R}^n$ and denote the vectors of the rates and the bandwidth-fractions given to the n users in frequency band j, respectively.

The analysis to show the existence of a solution and convergence of the barrier method is similar to that before. We now illustrate an efficient method to compute the Newton step during each Newton iteration. Again, we interleave all the variables into one vector $x \in \mathbb{R}^{2nm}$, $x = (r_1^1, b_1^1, \ldots, r_1^m, b_1^m, \ldots, r_n^1, b_n^1, \ldots, r_n^m, b_n^m)$.

The barrier function is given by

$$\phi(x) = -\sum_{i=1}^{n}\sum_{j=1}^{m}(\log r_i^j + \log b_i^j) - \log(1 - p(r,b)).$$

Also, denote

$$\psi_t(x) = -tU(r) - \phi(x)$$

where now $U(r) = \sum_{i=1}^{n} U_i\left(\sum_{j=1}^{m} r_i^j\right)$.

Then, at each iteration of the barrier method, we solve the following problem using Newton's method:

$$\begin{array}{ll} \text{maximize} & \psi_t(x) \\ \text{subject to} & \mathbf{1}^T b^j = 1,\ j = 1,\ldots,m. \end{array} \tag{7}$$

The Newton step for this problem can be computed through the solution of the linear equation in (4), where now, d is a $2mn \times m$ matrix give by

$$d = \begin{bmatrix} d_{\text{user}} \\ \vdots \\ d_{\text{user}} \end{bmatrix}$$

where d_{user} is a $2m \times m$ matrix whose $(2i, i)$ entry is one for $i = 1, \ldots, n$, and all other entries are zero. Now

$$\begin{aligned} \nabla^2 \psi_t(x) &= -t\nabla^2 U(r) + \nabla^2 \phi(x) \\ &= \frac{1}{(1-p(r,b))^2} \nabla p(r,b) \nabla p(r,b)^T \\ &\quad + \begin{bmatrix} K_1 & & & \\ & K_2 & & \\ & & \ddots & \\ & & & K_n \end{bmatrix} \end{aligned}$$

where the blocks not shown are all zero, and K_is are $2m \times 2m$ matrices given by the following:

$$\begin{aligned} K_i = &-tU_i''\left(\sum_{j=1}^{m} r_i^j\right) \begin{bmatrix} 1 & 0 & 1 & 0 & \ldots & 1 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ & & & & \vdots & & \\ 1 & 0 & 1 & 0 & \ldots & 1 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \end{bmatrix} \\ &+ \begin{bmatrix} H_i^1 & & \\ & \ddots & \\ & & H_i^m \end{bmatrix} \end{aligned}$$

where

$$\begin{aligned} H_i^j = &\begin{bmatrix} 1/(r_i^j)^2 & 0 \\ 0 & 1/(b_i^j)^2 \end{bmatrix} \\ &+ \frac{1}{1-p(r,b)} \begin{bmatrix} e^{r_i^j/b_i^j} c_i^j / b_i^j & -e^{r_i^j/b_i^j} c_i^j r_i^j / (b_i^j)^2 \\ -e^{r_i^j/b_i^j} c_i^j r_i^j / (b_i^j)^2 & e^{r_i^j/b_i^j} c_i^j (r_i^j)^2 / (b_i^j)^3 \end{bmatrix}. \end{aligned}$$

Thus, K_i is the sum of a block diagonal matrix (where the blocks are 2×2) and a rank one matrix. Hence, K_i can be inverted in $O(m)$ time. Now, the Hessian of $\psi_t(x)$ is the sum of a rank one matrix and a block diagonal matrix with blocks given by the K_is, each of which can be inverted in $O(m)$ time. Using the elimination of variables as before, it can be shown that each Newton iteration can be performed in $O(nm)$ time – compare this with a general-purpose method which costs $O(n^3 m^3)$. Thus, the reduction in complexity is huge, especially because in many systems the number of users, n, can be large [33], [34].

V. Scheduling Algorithms With Memory

We now illustrate the application of our computational techniques to design a scheduling heuristic which greedily maximizes the sum utility of user rates at every time-step. The average is computed in an online manner using an exponential filter. This can be used to model the behavior that the end-user experience is a function of the scheduled rates over multiple consecutive time-slots rather than a single scheduling decision. We focus on the downlink.

A. Utility Functions

The utility for user i is a function of its average rate. We consider an exponential averaging filter; in particular the average rate, $y_i(\tau)$, for user i is computed at time τ as follows:

$$y_i(\tau) = \alpha r_i(\tau) + (1-\alpha) y_i(\tau - 1) \tag{8}$$

where $r_i(\tau)$ is the rate allocated to user i at time τ, and $0 < \alpha < 1$. Also, we assume all users are initialized with (possibly very small) nonzero average rates $y_i(0) > 0$. Then the utility of user i at time τ is given by $U_i(y_i(\tau))$, so the total utility is

$$\sum_{i=1}^{n} U_i(y_i(\tau)).$$

The assumptions on U_i are the same as those in previous sections. However, note that now $U_i(\alpha r_i(\tau) + (1-\alpha)y_i(\tau-1))$ is well defined for $r_i(\tau) = 0$ because $y_i(0) > 0$ (and hence, $y_i(\tau) > 0$ for all finite τ).

B. Resource Allocation

The total (weighted) normalized power consumption when each user i is allocated rate $r_i(\tau)$ and bandwidth-fraction $b_i(\tau)$ is

$$p(r(\tau), b(\tau)) = \sum_{i=1}^{n} c_i(\tau) f(r_i(\tau), b_i(\tau))$$

where $c_i(\tau) = w_i N_0 B / (G_i(\tau) K P_{\max})$ and $G_i(\tau)$ is the channel gain for the ith user at time τ.

Our goal is to choose $r(\tau)$ and $b(\tau)$ at each time τ to greedily maximize the total utility subject to the power constraint and the total bandwidth constraint. Thus, at each time τ, we solve the following resource allocation problem:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} U_i(\alpha r_i(\tau) + (1-\alpha) y_i(\tau-1)) \\ \text{subject to} & \mathbf{1}^T b(\tau) = 1 \\ & (r(\tau), b(\tau)) \in S \\ & p(r(\tau), b(\tau)) \le 1. \end{array} \tag{9}$$

The optimization variables are $r_i(\tau)$ and $b_i(\tau)$; the problem data are $c_i(\tau)$, $y_i(\tau-1)$, and the functions U_i. We refer to the resulting scheduling algorithm as a *greedy utility maximization* algorithm. Even though at each time-step, the solution to the above problem is computed with high accuracy, we study the resulting scheduler over a longer time horizon only via a numerical experiment. Hence, when viewed over multiple time-steps, the resulting algorithm is a heuristic.

C. Relation to Asymptotically-Optimal Bandwidth Allocation

Note that when we take α to be small enough and restrict power allocation to be uniform across the entire bandwidth, the problem in (9) can be approximated as

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} U_i'(y_i(\tau-1)) r_i(\tau) \\ \text{subject to} & r_i(\tau) = b_i \log(1 + 1/c_i(\tau)), \ \forall i = 1, \dots, n \\ & \mathbf{1}^T b(\tau) = 1, (r(\tau), b(\tau)) \in S. \end{array} \tag{10}$$

The above problem is thus essentially an optimization problem in the $b_i(\tau)$s where the objective function is a linear combination of the $b_i(\tau)$s with positive coefficients:

$$\sum_{i=1}^{n} b_i(\tau) U_i'(y_i(\tau-1)) \log(1 + 1/c_i(\tau))$$

and the constraint is a sum constraint on the $b_i(\tau)$s. Hence, a solution to the above optimization problem is one where all the bandwidth (and power) is allocated to a user i for which

$$\log\left(1 + \frac{1}{c_i(\tau)}\right) U_i'(y_i(\tau-1)) \ge \log\left(1 + \frac{1}{c_j(\tau)}\right) U_j'(y_j(\tau-1)), \ \forall j = 1, \dots, n. \tag{11}$$

This scheduling scheme has been widely studied in the literature. It has been shown that under appropriate assumptions on the channel gain processes $G_i(\tau)$s and when power is uniformly allocated across the bandwidth, the above bandwidth allocation scheme (roughly) maximizes the total utility of rates averaged over a very long time horizon [3]. Hence, we refer to this scheme as an *asymptotically optimal bandwidth allocation* scheme.

The above scheduling scheme is a good one for narrowband systems and when there are few users in the system—it exploits multiuser diversity well and users get scheduled after relatively short intervals of time. However, with the advent of fourth generation wideband systems (e.g., LTE, WiMax, and UMB) we need to consider schemes which will distribute the resources among multiple users simultaneously due to the following reasons.

1) Wideband systems can have a total bandwidth of 20 MHz, and if all the bandwidth is allocated to one user (cell-phone), the user (cell-phone) may not even have enough processing power to decode the huge burst of data. In fact, the UMB spec specifies an upper bound on the amount of data that can be transmitted to a user in a single time-slot [33].
2) Fourth generation systems can have thousands of flows per cell and hybrid ARQ mechanisms. Consider the case where there are 5000 flows and each time-slot is 1 ms. Moreover, assume that it takes 3 hybrid ARQ transmissions to transmit a packet. Then if all the flows experience independent and identically distributed (i.i.d.) channels, on average each flow will get scheduled roughly every 15 s—this is clearly not acceptable for many types of traffic even when the individual packets do not have strict delay requirements. In many applications (e.g., web browsing), a user's utility, i.e., the end-user experience is a function of the average rate it sees over a short time horizon in the past rather than over a very long time horizon. Also, in many practical systems, this will lead to TCP time-outs because the long interscheduling time will be interpreted as congestion, thereby deprecating performance.

We note that the problem formulation in (9) is for a general value of $\alpha \in (0,1)$ and without any restriction on the power profile across the total bandwidth.

D. Existence and Uniqueness of Solution to Problem (9)

For convenience we will redefine the feasible set D by

$$D = \left\{ (r(\tau), b(\tau)) \in \mathbb{R}^{2n} \mid \mathbf{1}^T b(\tau) = 1, \ p(r(\tau), b(\tau)) \le 1, \ (r(\tau), b(\tau)) \in S \right\}.$$

We now have the equivalent problem

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} U_i(\alpha r_i(\tau) + (1-\alpha) y_i(\tau - 1)) \\ \text{subject to} \quad & (r(\tau), b(\tau)) \in D. \end{aligned} \tag{12}$$

Also, we simplify notation and drop the dependence of the variables on τ. And, we denote

$$U(r) = \sum_{i=1}^{n} U_i((1-\alpha) y_i(\tau - 1) + \alpha r_i). \tag{13}$$

We show that the resource allocation problem (12) has a unique solution $(r^\star, b^\star)$. The proof of the following lemma can be found in the Appendix .

Lemma 6: The set D is closed.

We now have the following theorem showing the existence and uniqueness of the solution.

Theorem 2: There exists a unique $(r^\star, b^\star) \in D$ such that

$$U(r^\star) = \sup\{ U(r,b) \mid (r,b) \in D \}.$$

Proof: First notice that problem (12) is feasible. That is, the set D is nonempty, since for small enough $\epsilon > 0$ the choice $b = (1/n)\mathbf{1}$, $r = \epsilon \mathbf{1}$ satisfies $(r,b) \in D$. The boundedness of D is easy to see. Since D is closed, the supremum is achieved. Uniqueness follows from strict concavity of U. ■

E. Fast Barrier Method

The barrier method to solve problem (12) is identical to that in Section III except that the utility function is now given by that in (13). Hence, using our approach we can solve problem (12) in $O(n)$ time.

F. Numerical Results

We considered a time-varying channel model similar to that in Section III-F. In particular, we consider 300 users with i.i.d. Rayleigh fading channels with 25 Hz Doppler and mean gain of 0 dB. A typical sample path for this channel is shown in Fig. 3. We again set $U_i(y_i(\tau)) = k_i \log(y_i(\tau))$, where k_i were generated as independent uniform random variables on $[1, 10]$. Also, we set $1/\alpha = 100$ ms. Thus, if a user, i, does not get scheduled for 100 ms, its average rate, $y_i(\tau)$, decays by about 33%. The problem in (9) was resolved every 1 ms.

In Fig. 3, we plot the utility function as a function of time (after initial transients) for the following three resource allocation schemes.

1) *Greedy utility maximization:* This scheme corresponds to allocating resources according to the solution of (9) which is updated every millisecond.
2) *Asymptotically-Optimal Bandwidth Allocation:* All the resources are allocated to a single user according to the scheduling policy in (11).
3) *Equal Resource:* In this scheme, power and spectrum are equally distributed among all users at all times.

Since we use log utilities for our computations, the difference in utilities is a reasonable metric for comparison (vs. ratios of

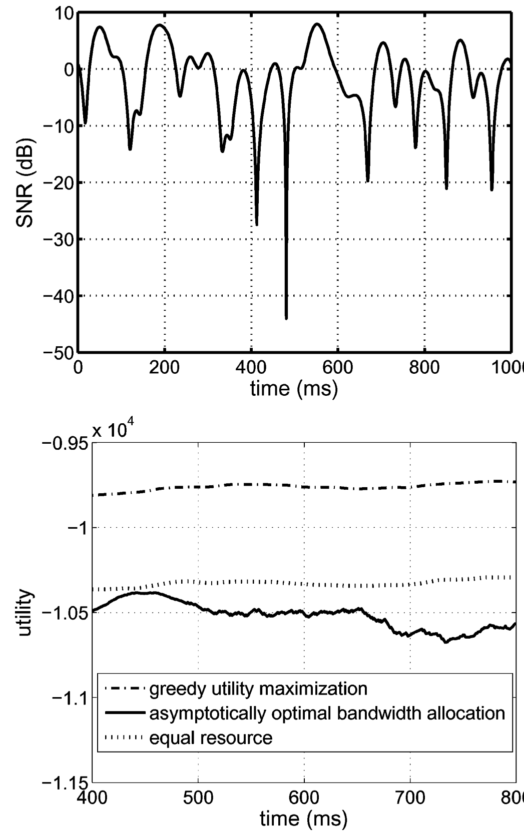

Fig. 3. Scheduling with memory and log utilities. *(Top)* Typical sample path of channel gain. *(Bottom)* Evolution of utility functions with time for three different scheduling policies.

utilities which can change a lot depending on the units of r_is). Also, note that the large negative values for the total utility are because we consider normalized rates $r_i(\tau)$s, and so $r_i(\tau) \leq 1$ always. We see that the net utility for the asymptotically optimal bandwidth allocation algorithm is lower than that for the greedy utility maximization algorithm—this is to be expected because the asymptotically optimal bandwidth allocation algorithm is designed for (a) very large time constants, i.e., small values of α, and (b) when the power allocation is restricted to be uniform across the entire bandwidth. In fact, the equal resource allocation algorithm outperforms the asymptotically-optimal bandwidth allocation algorithm.

We show the evolution of the average rate of a single user in Fig. 4. At any time τ, the increase in average rate is due to resources allocated to that user, while the decay is due to the exponential averaging when no resources are allocated. We can see that the greedy utility maximization scheme dominates the equal resource scheme—this is because the equal resource scheme does not take advantage of (a) multiuser diversity by allocating more resources to users which have strong channels at any given time, and (b) the knowledge of difference in the coefficients k_i's in the sum utility function. Also, for most of the time, the greedy utility maximization scheme has a higher average rate than that for the asymptotically optimal bandwidth allocation scheme. This is because the asymptotically-optimal bandwidth allocation scheme allocates resources to only a single user at a time and the resource allocations for a given user are separated by larger times.

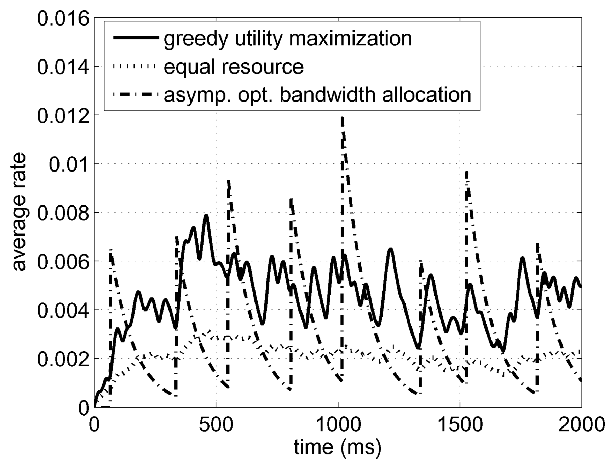

Fig. 4. Evolution of single user's average rate for three different resource allocation schemes.

VI. Discussion: Comparison With Other Computational Methods

Many resource allocation problems in wireless networks are either convex or can be approximated by convex problems (*e.g.*, [25], [26], and [40]). While a general interior point method can be used to solve these problems, in many cases it is possible to exploit the structure of the optimization problem to obtain fast and/or distributed algorithms. Next, we compare our approach with two other such approaches.

A. Dual Subgradient Method

The subgradient method (applied to the dual) can also be used to solve the optimization problem (1) (see [23] for such a method for CDMA systems). Such a method has an economic interpretation where the dual variables act as prices for violating constraints [7]. However, the rate of convergence of this method is highly dependent on the various condition numbers in the problem, and it will typically converge much more slowly than the algorithm presented here. Moreover, each iteration of the subgradient method also has $O(n)$ complexity, which is the same as that for our method. Unlike the subgradient approach, the fast convergence of our method enables it to be used for fading channels, as the number of iterations required for reconvergence after a warm start is small. However, we note that the subgradient method can be used to derive (typically slow) distributed algorithms for resource allocation problems in an adhoc wireless network (e.g., [27]), or the Internet [7]; for such problems exploiting the structure in the computation of the Newton step is typically not possible. Dual decomposition, primal decomposition, or joint primal-dual decomposition can be used (e.g., [14]).

B. Waterfilling

For the special case of log-utility functions, a waterfilling algorithm can be obtained to solve the problem (1), where during each iteration, we adjust a dual variable λ and recompute r_i and b_i. This is similar to the waterfilling algorithm to compute the capacity of a wireless channel—see, for example, [39, Ch. 4]. While this might appear to be a better algorithm, the complexity of this method is quite similar to the complexity of the barrier method described in this paper. In both algorithms: 1) each iteration has a cost that is $O(n)$; 2) around 10–25 or so steps are needed to solve the problem; and 3) a good initial condition gives convergence within fewer steps. We also note that the waterfilling approach can be used to solve the problem in [23].

VII. Conclusion

In this paper, we derived an efficient optimization algorithm to compute the optimal resource allocation in the downlink of an OFDM wireless cellular network. We showed that our algorithm converges to the optimal solution and has a complexity of $O(n)$ for n users. Numerical results show that our algorithm converges very fast in practice. Thus, our algorithm can be implemented in an online manner even for OFDM networks with high-resource granularity. Extension to frequency selective fading and an application to scheduling algorithms with memory are also discussed.

Appendix

Proof [Lemma 1]: Suppose (x^k, y^k) is a sequence of points in D converging to $(r,b) \in \bar{D}$. Now suppose i is such that $b_i = 0$, and $r_i > 0$. Then we have $\lim_{k\to\infty} f(x_i^k, y_i^k) = \infty$ and hence $p(x^k, y^k)$ also tends to infinity, contradicting the assumption that $(x^k, y^k) \in D$. ■

Proof [Lemma 2]: If $\lim_{k\to\infty} U(r^k) = -\infty$ then we are done. Suppose not, and let $T = \{\, i \mid r_i = 0 \,\}$. For $\epsilon > 0$ define $y(\epsilon)$ by

$$y_i(\epsilon) = \begin{cases} \epsilon, & \text{if } i \in T \\ b_i - \frac{\epsilon |T|}{n - |T|}, & \text{otherwise.} \end{cases}$$

Then $\mathbf{1}^T y(\epsilon) = 1$ for all $\epsilon > 0$. Also define $x(\epsilon)$ by

$$x_i(\epsilon) = \begin{cases} \alpha\epsilon, & \text{if } i \in T \\ r_i - \beta\epsilon, & \text{otherwise} \end{cases}.$$

where $\alpha > 0$ and $\beta > 0$. For $\beta > 0$ sufficiently large we have for all $i \notin T$

$$\left.\frac{df\big(x_i(\epsilon), y_i(\epsilon)\big)}{d\epsilon}\right|_{\epsilon=0} < 0.$$

Pick such a β. Hence

$$\frac{dp\big(x(\epsilon), y(\epsilon)\big)}{d\epsilon} = |T|(e^{\alpha} - 1) + \sum_{i \notin T} \frac{d}{d\epsilon} f\big(x_i(\epsilon), y_i(\epsilon)\big)$$

and therefore for $\alpha > 0$ sufficiently small

$$\left.\frac{dp\big(x(\epsilon), y(\epsilon)\big)}{d\epsilon}\right|_{\epsilon=0} < 0$$

and hence for $\epsilon > 0$ sufficiently small we have $p\big(x(\epsilon), y(\epsilon)\big) < 1$ and hence $\big(x(\epsilon), y(\epsilon)\big) \in D$. Now we have

$$U\big(x(\epsilon)\big) - \lim_{k\to\infty} U(r^k) = \epsilon \sum_{i=1}^{n} \frac{U_i\big(p_i(\epsilon)\big) - \lim_{k\to\infty} U_i(r_i^k)}{\epsilon}.$$

Now if $i \in T$, as $\epsilon \to 0^+$ we have

$$\frac{U_i\big(x_i(\epsilon)\big) - \lim_{k\to\infty} U_i(r_i^k)}{\epsilon} \to \infty$$

and if $i \notin T$ then as $\epsilon \to 0^+$

$$\frac{U_i\big(x_i(\epsilon)\big) - \lim_{k\to\infty} U_i(r_i^k)}{\epsilon} \to \beta U_i'(r_i).$$

Hence for $\epsilon > 0$ sufficiently small

$$\lim_{k\to\infty} U(r^k) < U\big(x(\epsilon)\big)$$

as desired. ■

Proof [Lemma 3]: If $\lim_{k\to\infty} U(r^k) = -\infty$ then we are done. Suppose not, and let $T = \{\, i \mid r_i = 0 \text{and } b_i > 0 \,\}$. Define $y \in \mathbb{R}^n$ by

$$y_i = \begin{cases} 0, & \text{if } i \in T \\ b_i + \frac{\sum_{j\in T} b_j}{n-|T|}, & \text{otherwise.} \end{cases}$$

Then $\mathbf{1}^T y = 1$ and $y \geq 0$. For any $x > 0$ we have

$$f(x, z_1) > f(x, z_2) \quad \text{if } 0 < z_1 < z_2.$$

If $r \neq 0$ then for some $i \notin T$ we have $r_i > 0$ and hence $p(r,y) < p(r,b) \leq 1$. Also clearly if $r = 0$ then $p(r,y) < 1$. Now for $\epsilon > 0$ define $x(\epsilon)$ by

$$x_i(\epsilon) = \begin{cases} r_i + \epsilon, & \text{if } r_i > 0 \text{ and } b_i > 0 \\ r_i, & \text{otherwise.} \end{cases}$$

Since p is continuous, there exists $\epsilon > 0$ sufficiently small so that $p\big(x(\epsilon), y\big) < 1$. Pick such an ϵ. Then since U_i is increasing we have

$$U\big(x(\epsilon)\big) > \lim_{k\to\infty} U(r^k).$$

Now either $x > 0$ and $y > 0$, in which case the proof is complete, or there is some i such that $(x_i(\epsilon), y_i) = (0,0)$. In this case the conditions of Lemma 2 hold, and this then gives the desired result. ■

Proof [Lemma 4]: $x^{(0)}$ is strictly feasible by assumption. Now we use induction to prove the lemma.

Consider iteration $k+1$, and assume that $x^{(k)} = (r^{(k)}, b^{(k)})$ is strictly feasible. Denote the Newton step by $(\Delta r^{(k)}, \Delta b^{(k)})$. Now, let $\hat{l}$ be the minimum value of l such that for some i, we have $r_i^{(k)} + \hat{l}\Delta r_i^{(k)} = 0$ or $b_i^{(k)} + \hat{l}\Delta b_i^{(k)} = 0$, or $p(r^{(k)} + \hat{l}\Delta r^{(k)}, b^{(k)} + \hat{l}\Delta b^{(k)}) = 1$. Thus, $\hat{l}$ is the minimum value of l for which $(r^{(k)} + l\Delta r^{(k)}, b^{(k)} + l\Delta b^{(k)})$ is not strictly feasible. We claim that as $l \to \hat{l}$, $f(r^{(k)} + l\Delta r^{(k)}, b^{(k)} + l\Delta b^{(k)}) \to \infty$, i.e., the step length returned by the line search algorithm is less than $\hat{l}$, which implies that the $(k+1)$th iterate is strictly feasible.

Note that $r_i^{(k)} + \hat{l}\Delta r_i^{(k)}$ and $b_i^{(k)} + \hat{l}\Delta b_i^{(k)}$ are finite for all i. Now assume that $l < \hat{l}$. Then $U(r^{(k)} + l\Delta r^{(k)})$ is upper bounded. Similarly $\log(r_i^{(k)} + l\Delta r_i^{(k)})$ and $\log(b_i^{(k)} + l\Delta b_i^{(k)})$ are upper bounded for all i. Also, $(1 - p(r^{(k)} + l\Delta r^{(k)}, b^{(k)} + l\Delta b^{(k)}))$ is upper bounded by 1. Hence, it follows from the definition of $f(r,b)$ that that as $l \to \hat{l}$, $f(r^{(k)} + l\Delta r^{(k)}, b^{(k)} + l\Delta b^{(k)}) \to \infty$, as claimed above. ■

Proof [Lemma 5]: For all $(r,b) \in L$, $\mathbf{1}^T b = 1$. By the above lemma, all iterates are strictly feasible. Since $b > 0$ for all $(r,b) \in L$, the b_is are bounded above by 1, which implies that $\sum_{i=1}^n \log b_i$ is bounded above. Also, $0 < p(r,b) < 1$ for all $(r,b) \in L$, i.e., $\log(1 - p(r,b))$ is bounded above by zero. Since $p(r,b)$ is an increasing function of the r_is and decreasing function of the b_is, and $b_i \leq 1$ for all $(r,b) \in L$, it follows that r_is are bounded above by a constant for all $(r,b) \in L$. This also implies that $U(r)$ is bounded above by some $\bar{U}$ for $(r,b) \in L$.

Now, we show that r_is and b_is are bounded away from zero for all $(r,b) \in L$. To see this, first note that $U(r)$, $\sum_{i=1}^n \log b_i$, $\sum_{i=1}^n \log r_i$, and $\log(1 - p(r,b))$ are all bounded above for all $(r,b) \in L$. Thus, it follows that $\psi_t(r,b) \to \infty$ as $r_i \to 0$ or $b_i \to 0$ for any i. Then, the claim follows since the Newton method is a descent method, i.e., $\psi_t(r^{(k)}, b^{(k)}) \leq \psi_t(r^{(0)}, b^{(0)})$ for any iteration k. ■

Proof [Lemma 6]: We show that the complement of D, i.e., D^C is open. Note that D^C is the union of the following sets:

$$\begin{aligned} O^1 &= \{(x,y) \in \mathbb{R}^{2n} \mid \mathbf{1}^T y \neq 1\} \\ O^2 &= \{(x,y) \in \mathbb{R}^{2n} \mid x < 0\} \\ O^3 &= \{(x,y) \in \mathbb{R}^{2n} \mid x > 0, y \leq 0\} \\ O^4 &= \{(x,y) \in \mathbb{R}^{2n} \mid x = 0, y < 0\} \\ O^5 &= \{(x,y) \in \mathbb{R}^{2n} \mid x \geq 0, p(x,y) > 1, y > 0\}. \end{aligned}$$

It is easy to see that O^1 and O^2 are open. Since, the union of open sets is open, it is sufficient to show that $O^3 \cup O^4 \cup O^5$ is open. To do this, consider a point $(x,y) \in O^3 \cup O^4 \cup O^5$. Hence, either $(x,y) \in O^3$ or $(x,y) \in O^4$ or $(x,y) \in O^5$ – in each of these cases there exists an ϵ–ball around (x,y) which is contained in $O^3 \cup O^4 \cup O^5$. ■

REFERENCES

[1] S. Shakkottai, T. Rappaport, and P. Karlsson, "Cross-layer design for wireless networks," *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 74–80, Oct. 2003.

[2] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Found. Trends Netw.*, vol. 1, no. 1, pp. 1–144, 2006.

[3] H. J. Kushner and P. A. Whiting, "Convergence of proportional-fair sharing algorithms under general conditions," *IEEE Trans. Wireless Commun.*, vol. 3, no. 4, pp. 1250–1259, Jul. 2004.

[4] A. Stolyar, "Greedy primal-dual algorithm for dynamic resource allocation in complex networks," *Queue. Syst.*, vol. 54, pp. 203–220, 2006.

[5] S. Shakkottai and R. Srikant, "Scheduling real-time traffic with deadlines over a wireless channel," *ACM/Baltzer Wireless Netw. J.*, vol. 8, no. 1, pp. 13–26, 2002.

[6] S. Shakkottai and A. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," *Amer. Math. Soc.*, ser. 2A,, vol. 207, Memory of F. Karpelevich, pp. 185–202, 2002.

[7] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, 1998.

[8] D. Tse and P. Viswanath, *Fundamentals of Wireless Communcation*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[9] J. M. Holtzman, "Asymptotic analysis of proportional fair algorithm," in *Proc. 2001 12th IEEE Int. Symp. Pers., Indoor, Mobile Radio Commun.*, , Sep./Oct. 2001, vol. 2, pp. F–33–F–37.

[10] S. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," in *Proc. IEEE INFOCOM*, Mar. 30–Apr. 3, 2003, vol. 1, pp. 321–331.

[11] A. Stolyar, “On the asymptotic optimality of the gradient scheduling algorithm for multi-user throughput allocation,” *Oper. Res.*, vol. 53, pp. 12–25, 2005.
[12] J. Huang, V. G. Subramanian, R. Agrawal, and R. Berry, “Downlink scheduling and resource allocation for OFDM systems,” in *Proc. CISS*, 2006, pp. 1272–1279.
[13] A. Eryilmaz and R. Srikant, “Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control,” in *Proc. IEEE INFOCOM*, 2005, vol. 3, pp. 1794–1803.
[14] B. Johansson and M. Johansson, “Mathematical decomposition techniques for distributed cross-layer optimization of data networks,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1535–1547, Aug. 2006.
[15] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, “Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks,” in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–13.
[16] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, “Scheduling in a queueing system with asynchronously varying service rates,” *Probab. Eng. Inf. Sci.*, vol. 18, pp. 191–217, 2004.
[17] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” in *Proc. 29th IEEE CDC*, Dec. 5–7, 1990, vol. 4, pp. 2130–2132.
[18] M. Neely, E. Modiano, and C. Rohrs, “Power and server allocation in a multi-beam satellite with time-varying channels,” in *Proc. IEEE INFOCOM*, 2002, pp. 1451–1460.
[19] M. Neely, E. Modiano, and C. Rohrs, “Dynamic power allocation and routing for time-varying wireless networks,” in *Proc. IEEE INFOCOM*, 2003.
[20] S. Shakkottai and A. Stolyar, “Scheduling algorithms for a mixture of real-time and non-real-time data in HDR,” in *Proc. ITC-17*, 2001, pp. 793–804.
[21] N. Chen and S. Jordan, “Throughput in processor-sharing queues,” *IEEE Trans. Autom. Control*, vol. 52, no. 2, pp. 299–305, Feb. 2007.
[22] N. Chen and S. Jordan, “Downlink scheduling with probabilistic guarantees on short-term average throughputs,” in *Proc. IEEE WCNC*, 2008, pp. 1865–1870.
[23] P. Tinnakornsrisuphap and C. Lott, “On the fairness and stability of the reverse-link MAC layer in CDMA2000 1xEV-DO,” in *Proc. IEEE GLOBECOM*, 2004, vol. 1, pp. 144–148.
[24] P. Hande, S. Rangan, and M. Chiang, “Distributed uplink power control for optimal SIR assignment in cellular data networks,” in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–13.
[25] D. O’Neill, D. Julian, and S. Boyd, “Adaptive management of network resources,” in *Proc. IEEE Veh. Technol. Conf.*, 2003, vol. 3, pp. 1929–1933.
[26] 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. IEEE INFOCOM*, 2004, vol. 2, pp. 1446–1456.
[27] M. Johansson, L. Xiao, and S. Boyd, “Simultaneous routing and resource allocation in CDMA wireless data networks,” in *Proc. IEEE ICC*, 2003, vol. 1, pp. 51–55.
[28] J. Jang and K. B. Lee, “Transmit power adaptation for multiuser OFDM systems,” *IEEE J. Sel. Areas Commun.*, vol. 21, no. 2, pp. 171–178, 2003.
[29] L. M. C. Hoo, B. Halder, J. Tellado, and J. M. Cioffi, “Multiuser transmit optimization for multicarrier broadcast channels: Asymptotic FDMA capacity region and algorithms,” *IEEE Trans. Commun.*, vol. 52, no. 6, pp. 922–930, Jun. 2004.
[30] Y. Zhang and K. Letaief, “Multiuser adaptive subcarrier and bit allocation with adaptive cell selection for OFDM systems,” *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1566–1575, Sep. 2004.
[31] H. Yin and H. Liu, “An efficient multiuser loading algorithm for OFDM-based broadband wireless systems,” in *Proc. IEEE GLOBECOM*, 2000, vol. 1, pp. 103–107.
[32] K. Seong, M. Mohseni, and J. M. Cioffi, “Optimal resource allocation for OFDMA downlink systems,” in *Proc. ISIT*, 2006, pp. 1394–1398.
[33] “Ultra mobile broadband (UMB),” [Online]. Available: http://www.3gpp2.org/Public_html/specs/tsgc.cfm
[34] “UTRA-UTRAN long term evolution (LTE),” [Online]. Available: http://www.3gpp.org/Highlights/LTE/LTE.htm
[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[36] G. J. Foschini and J. Salz, “Digital communications over fading radio channels,” *Bell Syst. Tech. J.*, pp. 429–456, 1983.
[37] S. Wright, *Primal-Dual Interior-Point Methods*. Singapore: SIAM, 2003.
[38] C. T. Kelley, *Solving Nonlinear Equations with Newton’s Method*. Singapore: SIAM, 2003.
[39] A. J. Goldsmith, *Wireless Communcations*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[40] S. Cui, R. Madan, A. J. Goldsmith, and S. Lall, “Joint routing, MAC, and link layer optimization in sensor networks with energy constraints,” in *Proc. IEEE ICC*, 2005, vol. 2, pp. 725–729.

Ritesh Madan received the B.Tech. degree from the Indian Institute of Technology (IIT) Bombay, Bombay, India, in 2001, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 2003 and 2006, respectively, all in electrical engineering.

Currently, he is with Qualcomm-Flarion Technologies, Bridgewater, NJ. At Stanford University, he was a recipient of the Sequoia Capital Stanford Graduate Fellowship. He has held visiting research positions at Mitsubishi Electric Research Labs (MERL), Cambridge, MA, and at the Tata Institute of Fundamental Research (TIFR), Mumbai, India. His research interests include wireless networks, convex optimization, networked control, and dynamic programming.

Stephen P. Boyd (S’82–M’85–SM’97–F’99) received the A.B. degree in mathematics from Harvard University, Cambridge, MA, in 1980, and the Ph.D. degree in electrical engineering and computer science from the University of California, Berkeley, in 1985.

In 1985, he joined the faculty at Stanford University, Stanford, CA, where he is currently the Samsung Professor of Engineering and a Professor of electrical engineering in the Information Systems Laboratory. His current research focus is on convex optimization applications in control, signal processing, and circuit design.

Sanjay Lall (S’92–M’96–SM’09) received the B.A. degree in mathematics and the Ph.D. degree in engineering from the University of Cambridge, Cambridge, U.K.

Currently, he is an Associate Professor of electrical engineering, Associate Professor of aeronautics and astronautics, and Vance D. and Arlene C. Coffman Faculty Scholar at Stanford University, Stanford, CA. Until 2000, he was a Research Fellow at the California Institute of Technology, Pasadena, in the Department of Control and Dynamical Systems, and prior to that, he was a NATO Research Fellow at Massachusetts Institute of Technology, Cambridge, in the Laboratory for Information and Decision Systems. His research focuses on the development of advanced engineering methodologies for the design of control systems which occur in a wide variety of aerospace, mechanical, electrical and chemical systems.

Prof. Lall received the George S. Axelby Outstanding Paper Award by the IEEE Control Systems Society in 2007, the NSF CAREER Award in 2007, the Presidential Early Career Award for Scientists and Engineers (PECASE) in 2007, and the Graduate Service Recognition Award from Stanford University in 2005.