

Fast Algorithms for Resource Allocation in Cellular Networks

Ritesh Madan¹

Stephen P. Boyd²

Sanjay Lall³

Abstract

We consider a 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 using time division multiple access (TDMA) 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. While, in this paper, we focus on TDMA systems, our approach extends to frequency selective channels, and to frequency division multiple access (FDMA), and code division multiple access (CDMA) systems. We briefly describe such extensions.

1 Introduction

We study the problem of resource allocation in wireless cellular networks. In particular, we study the assignment of power and spectral resources to maximize the sum-utility of the achieved data rates. We show that this 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, [1]). However,

¹R. Madan is at Qualcomm-Flarion Technologies, NJ, USA. This work was performed when he was at the Department of Electrical Engineering, Stanford University. rkmadan@stanfordalumni.org

²S. P. Boyd is at the Department of Electrical Engineering, Stanford University. boyd@stanford.edu

³S. Lall is at the Department of Aeronautics and Astronautics, Stanford University. lall@stanford.edu

This work was funded in part by the MARCO Focus Center for Circuit & System Solutions (C2S2, www.c2s2.org), under contract 2003-CT-888, by AFOSR grant AF F49620-01-1-0365, by NSF grant ECS-0423905, by NSF grant 0529426, by DARPA/MIT grant 5710001848, by AFOSR grant FA9550-06-1-0514, DARPA/Lockheed contract N66001-06-C-2021, by AFOSR/Vanderbilt grant FA9550-06-1-0312, the Stanford URI Architecture for Secure and Robust Distributed Infrastructures (AFOSR DoD award number 49620-01-1-0365), and the Sequoia Capital Stanford Graduate Fellowship.

the time-varying nature of wireless channels necessitates re-computation of an optimal resource allocation in an online manner. This requires the design of faster computational algorithms to track the optimal resource allocation.

In this paper, we consider a model where the spectral resources are distributed among multiple users using time division multiple access (TDMA). We exploit the underlying structure of the problem to derive a specialized barrier method that has a complexity of $O(n)$. The methods in this paper extend equally well to frequency division multiple access (FDMA) and code division multiple access (CDMA) systems. In addition, our methods can be generalized to the case of frequency selective fading, in, for example, orthogonal frequency division multiplexing (OFDM) systems.

The rest of the paper is organized as follows. We give the detailed mathematical model and problem formulation in Sec. 2.1. The optimization problem is shown to have a unique positive solution in Sec. 3. We exploit the structure of the underlying optimization problem to obtain an $O(n)$ algorithm in Sec. 4, and illustrate its typical behavior through computational results in Sec. 5. In Sec. 6, we describe extensions to CDMA, FDMA, and OFDM systems, and also compare our algorithm with other possible computational approaches.

2 Problem Formulation

2.1 System Model

We model a wireless cellular network that uses time division multiple access (TDMA) to divide spectrum between communication flows (users) on n links in a cell. We formulate a general problem which is applicable to both the uplink and the downlink. We denote the flow rate (in nats/sec/Hz) for user i by $r_i \geq 0$, and the fraction of time allocated to it by $\tau_i \geq 0$. We denote the associated vectors of rates and time-fractions as $r \in \mathbb{R}^n$ and $\tau \in \mathbb{R}^n$, respectively. Assuming M-Quadrature Amplitude Modulation (MQAM) is used, the average power consumption to support flow $r_i > 0$ can be modeled as

$$p_i(r_i, \tau_i) = a_i \tau_i (e^{r_i/\tau_i} - 1), \quad a_i = N_0 B / (G_i K).$$

Here, G_i is the channel gain over the link to user i , N_0 is the noise power spectral density, B is the bandwidth allo-

cated to the network, and $K = -1.5/\log(5\text{BER})$, where BER is the desired (constant) bit error rate [3]. When $r_i = 0$, the power required is 0. Hence, the power consumption of user i as a function of r_i and τ_i is $a_i f(r_i, \tau_i)$ where the function $f: S \rightarrow \mathbb{R}$ 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

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

where $P(r, \tau)$ 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 station should be kept below some threshold; the weights then represent the power gains to the neighboring base station. We will normalize the power constraint by defining the normalized power $p: S^n \rightarrow \mathbb{R}$ by $p(r, \tau) = \sum_{i=1}^n c_i f(r_i, \tau_i)$ where $c_i = w_i a_i / P_{\max}$. The power constraint is then $p(r, \tau) \leq 1$.

We first observe that p_i is a convex function of r_i and τ_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., [1, § 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 is therefore also a convex function of r and so, the total power constraint is a convex constraint for $r, \tau > 0$.

2.2 User Utility Functions

The utility for user i is a function of its average 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 $U_i: (0, \infty) \rightarrow \mathbb{R}$ are *thrice* continuously differentiable with

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

for all $x > 0$ and

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

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 $0 < a < 1$.

2.3 Maximum Utility Resource Allocation

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

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

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

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

We now have the equivalent problem

$$\begin{aligned} & \text{maximize} && U(r) \\ & \text{subject to} && (r, \tau) \in D \end{aligned} \tag{2}$$

In the following section we will show that there is a unique optimal allocation (r, τ) which is achieved at a point with $r > 0$ and $\tau > 0$. 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., [1, Ch. 11] or [12]). The algorithm we describe next solves the resource allocation problem much faster by exploiting its special structure. The resulting interior point method converges in about 30 iterations, where each iteration requires $O(n)$ operations with a modest constant.

3 Existence and Uniqueness of a Positive Solution

In this section, we show that the resource allocation problem (1) has a unique solution (r^*, τ^*) , with $r^* > 0$ and $\tau^* > 0$. We will do this by constructing a sequence of points converging to the minimum, 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 time-fraction and positive rate to a user requires infinite power. Hence for every point (r, τ) in the feasible set, we must have $\tau_i > 0$ 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, τ) with $(r_i, \tau_i) = (0, 0)$ for some i cannot be optimal. The idea here is that since U_i has infinite slope at 0, slightly increasing r_i and τ_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, τ^k) is a sequence in S^n with limit*

$$\lim_{k \rightarrow \infty} (r^k, \tau^k) = (r, \tau)$$

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

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

The final lemma needed shows that a point (r, τ) with $r_i = 0$ for some i must also have $\tau_i = 0$. If this were not the case, we could decrease τ_i to zero, spreading this time-fraction among the other users, who can use the extra time-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 $(r, \tau) = 0$.

Lemma 3. *Suppose (r^k, τ^k) is a sequence in S^n with limit*

$$\lim_{k \rightarrow \infty} (r^k, \tau^k) = (r, \tau)$$

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

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

Theorem 1. *There exists a unique $(r^*, \tau^*) \in D$ with $r^*, \tau^* > 0$ such that*

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

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

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

Then U^* is finite, since D is bounded and U is concave. We must show that this optimal value is actually achieved. Suppose (r^k, τ^k) is a maximizing sequence

in D , so that $U(r^k, \tau^k) \rightarrow U^*$. By extracting a subsequence, we can assume that (r^k, τ^k) converges to a point $(\bar{r}, \bar{\tau}) \in \bar{D}$. 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 $\bar{\tau} > 0$. Hence the optimal is achieved in D . Uniqueness now follows from strict concavity of U . \blacksquare

4 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 $O(n)$.

4.1 Barrier Method

We use the barrier method to solve the optimization problem in (2) [1]. The central point $(r^*(t), \tau^*(t))$ for a given value of t are given by the solution of the following problem

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

As t increases, $(r^*(t), \tau^*(t))$ 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 Sec. 4.3.

We collect the variables into one vector $x \in \mathbb{R}^{2n}$, $x = (r_1, \tau_1, \dots, r_n, \tau_n)$. Note that we have interleaved the rate and time-fraction variables here, so that the variables associated with a given user are adjacent. Also, we denote the barrier function as

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

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 $\mathbf{1}^T \tau - 1 = 0$, starting at x .
2. *Update.* $x := x^*(t)$.
3. *Stopping Criterion.* **quit** if $(2n+1)/t < \epsilon$.
4. *Increase t .* $t := \mu t$.

4.2 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

$$\begin{aligned} \begin{bmatrix} -t\nabla^2 U(r) + \nabla^2 \phi(x) & d \\ d^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \nu \end{bmatrix} \\ = \begin{bmatrix} t\nabla U(r) - \nabla \phi_t(x) \\ 0 \end{bmatrix}, \end{aligned} \quad (4)$$

where $d = [0 \ 1 \ \dots \ 0 \ 1]^T$. For the Newton method, we use a backtracking line search to ensure to ensure an adequate decrease in ϕ (see, e.g., [1, Ch.11] or [7]). The method is then as follows:

Given starting point x such that $\mathbf{1}^T \tau = 1$, 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 $\lambda^2/2 \leq \epsilon$
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.* $x := x + s\Delta x$.

4.3 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

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

The Hessian of the barrier function is given by

$$\begin{aligned} \nabla^2 \phi(x) &= \begin{bmatrix} 1/r_1^2 & & & & \\ & 1/\tau_1^2 & & & \\ & & \ddots & & \\ & & & 1/r_n^2 & \\ & & & & 1/\tau_n^2 \end{bmatrix} \\ &+ \frac{1}{(1 - p(r, \tau))^2} \nabla p(r, \tau) \nabla p(r, \tau)^T + \frac{1}{1 - p(r, \tau)} \nabla^2 p(r, \tau). \end{aligned}$$

Hence, it follows that

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

where the blocks not shown are all zero, and

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

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

$$\begin{aligned} \frac{\partial p(r, \tau)}{\partial r_i} &= c_i e^{r_i/\tau_i} \\ \frac{\partial p(r, \tau)}{\partial \tau_i} &= c_i e^{r_i/\tau_i} (1 - 1/\tau_i) - c_i. \end{aligned}$$

Let us denote

$$\begin{aligned} a &= \frac{1}{(1 - p(r, \tau))} \nabla p(r, \tau), \\ b &= t\nabla U(r) - \nabla \phi_t(x). \end{aligned}$$

Then we have

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

It is easy to show that $H_i > 0$. Since $aa^T \geq 0$, 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 equation (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 $H = \text{diag}(H_1, \dots, H_n)$. In particular, we have (see, for example, [1, App. C])

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

where

$$\begin{bmatrix} H & d \\ d^T & 0 \end{bmatrix} u = \begin{bmatrix} b \\ 0 \end{bmatrix}, \quad (5)$$

and

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

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

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

Substituting these back in (5), it follows that

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

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

4.4 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 time fractions, and powers to users such that the total power is less than 1, i.e., $p(r^{(0)}, \tau^{(0)}) < 1$; 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 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 $(r, \tau) \in \bar{L}$, r_i, τ_i s are bounded above and bounded away from zero.*

Since the KKT matrix on the left in equation (4) is invertible, and is a continuous function of (r, τ) , 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, τ) 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, [1, 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.

4.5 Warm Start

The Newton method can be initialized with $\tau = (1/n)\mathbf{1}$, and $r = \epsilon\mathbf{1}$, where $\epsilon > 0$ such that (r, τ) is strictly feasible, i.e., $p(r, \tau) < 1$. 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.

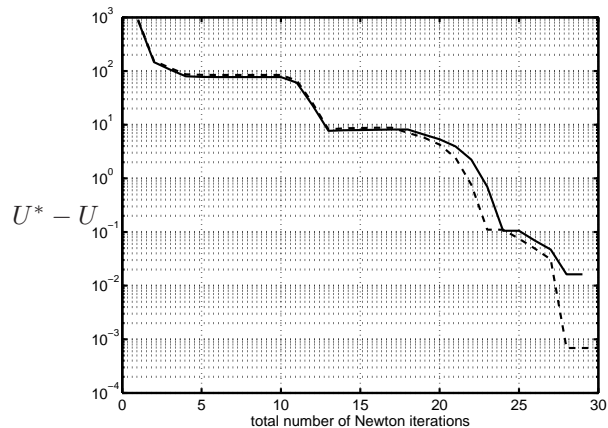
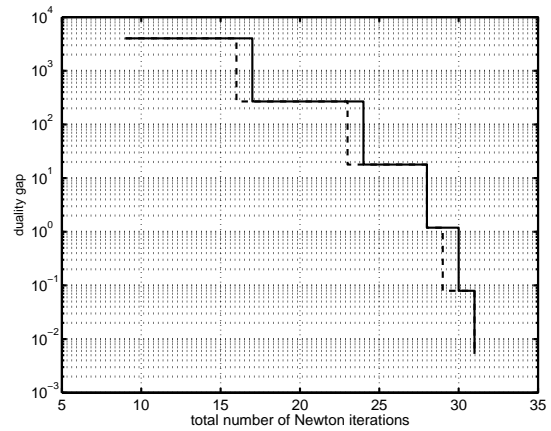


Figure 1: Typical convergence of the barrier method. *Top.* Norm of residual versus iteration for two different instances. *Bottom.* Convergence of $U^* - U$ versus iteration.

5 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 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 dB to 20dB. Figure 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

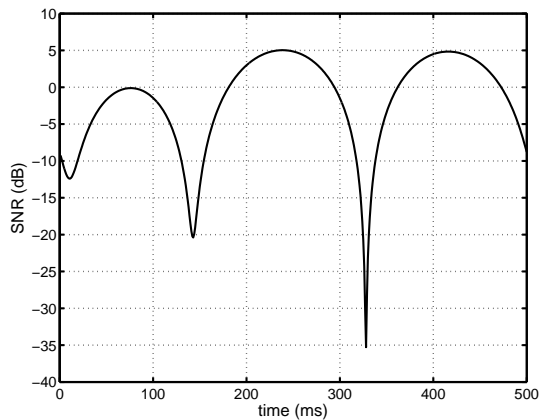


Figure 2: Typical realization of a Rayleigh fading channel with mean SNR of 0 dB and Doppler frequency 5 Hz.

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. Each user's channel undergoes mutually independent Rayleigh fading with a Doppler frequency of 5Hz and mean SNR of 0dB. We re-computed the optimal resource allocation at every time step of 1ms. Thus, the channel completely de-correlates after 200 time-steps or so. Also, the variation in channel gains over time is very high. A typical realization of such a channel is shown in Fig. 2 – the channel can easily swing over a range of 30 dB.

Figure 3 shows the number of Newton steps required to re-converge 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 time-steps we used a larger value of $t^{(0)}$ such that only 2 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 re-convergence is less than 15. A larger number of Newton iterations is occasionally required at times when the rate of change of channel is high; for example during deep fades.

6 Discussion

6.1 Comparison with Other Approaches

Many resource allocation problems in wireless networks are either convex or can be approximated by convex problems (*e.g.*, [2, 9, 10]). 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.

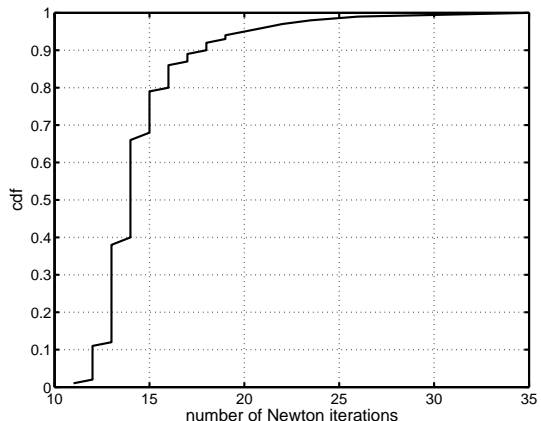
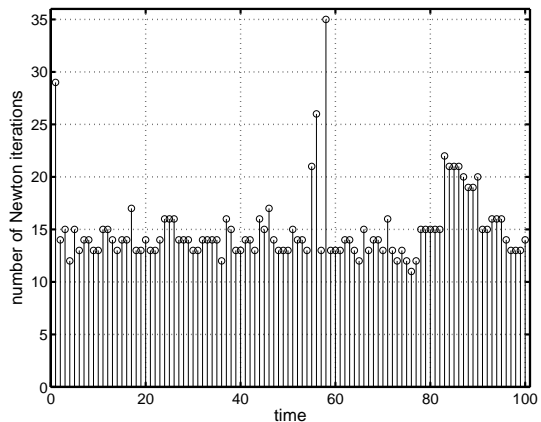


Figure 3: Number of Newton iterations needed for re-convergence with Rayleigh fading channels. *Top.* Number of Newton iterations for re-convergence during the first 100 time steps. *Bottom.* CDF of number of Newton iterations for re-convergence over 500 time-steps.

6.1.1 Dual Subgradient Method

The subgradient method (applied to the dual) can also be used to solve the optimization problem (1) (see [11] for such a method for CDMA systems). 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 re-convergence 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., [6]), or the internet [8]; 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., [5]).

6.1.2 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 τ_i . 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, (i) each iteration has a cost that is $O(n)$, (ii) around 10–25 or so steps are needed to solve the problem, and (iii) a good initial condition gives convergence within fewer steps.

6.2 Other System Models

We modeled the constraint that the average power for a TDMA system should be less than a given maximum average power. Adding instantaneous power constraints, or modeling only instantaneous power constraints is straightforward. The methods in this paper can also be used almost immediately to obtain $O(n)$ algorithms for utility maximization in FDMA systems with flat fading, and in CDMA systems [11]. For frequency division, the resulting optimization problem is convex in the rates and the bandwidths allocated to the users. The power constraint would now be an instantaneous power constraint. More generally, for a channel with frequency selective fading with m frequency sub-bands of bandwidth less than the coherence bandwidth [4], we can use time division for each of these sub-bands to maximize the total utility of the system. Here, the Hessian is the sum of a block arrow matrix and a rank one matrix, where each block is the sum of a rank one matrix and a $2m \times 2m$ block diagonal matrix with blocks of size 2×2 . Such

a matrix can be inverted efficiently to give an $O(nm)$ barrier method. Both instantaneous and average power constraints can be modeled.

Appendix

Proof. [Lemma 1] Suppose (x^k, y^k) is a sequence of points in D converging to $(r, \tau) \in \bar{D}$. Now suppose i is such that $\tau_i = 0$, and $r_i > 0$. Then we have $\lim_{k \rightarrow \infty} f(x_i^k, y_i^k) = \infty$ 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 $\lim_{k \rightarrow \infty} U(r^k) = -\infty$ then we are done. Suppose not, and let $T = \{i \mid r_i = 0\}$. For $\epsilon > 0$ define $y(\epsilon)$ by

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

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 $i \notin T$

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

Pick such a β . Hence

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

and therefore for $\alpha > 0$ sufficiently small

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

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

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

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

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

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

$$\frac{U_i(x_i(\epsilon)) - \lim_{k \rightarrow \infty} U_i(r_i^k)}{\epsilon} \rightarrow \beta U'_i(r_i)$$

Hence for $\epsilon > 0$ sufficiently small ■

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

as desired. ■

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

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

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 $p(r, y) < p(r, \tau) \leq 1$. Also clearly if $r = 0$ then $p(r, y) < 1$. Now for $\epsilon > 0$ define $x(\epsilon)$ by

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

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

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

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 $x^{(k)} = (r^{(k)}, \tau^{(k)})$ is strictly feasible. Denote the Newton step by $(\Delta r^{(k)}, \Delta \tau^{(k)})$. 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 $\tau_i^{(k)} + \hat{l} \Delta \tau_i^{(k)} = 0$, or $p(r^{(k)} + \hat{l} \Delta r^{(k)}, \tau^{(k)} + \hat{l} \Delta \tau^{(k)}) = 1$. Thus, \hat{l} is the minimum value of l for which $(r^{(k)} + l \Delta r^{(k)}, \tau^{(k)} + l \Delta \tau^{(k)})$ is not strictly feasible. We claim that as $l \rightarrow \hat{l}$, $f(r^{(k)} + l \Delta r^{(k)}, \tau^{(k)} + l \Delta \tau^{(k)}) \rightarrow \infty$, 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 $\tau_i^{(k)} + \hat{l} \Delta \tau_i^{(k)}$ 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 $\log(\tau_i^{(k)} + l \Delta \tau_i^{(k)})$ are upper bounded for all i . Also, $(1 - p(r^{(k)} + l \Delta r^{(k)}, \tau^{(k)} + l \Delta \tau^{(k)}))$ is upper bounded by 1. Hence, it follows from the definition of $f(r, \tau)$ that that as $l \rightarrow \hat{l}$, $f(r^{(k)} + l \Delta r^{(k)}, \tau^{(k)} + l \Delta \tau^{(k)}) \rightarrow \infty$, as claimed above.

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

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

References

- [1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [2] S. Cui, R. Madan, A. J. Goldsmith, and S. Lall. Joint routing, MAC, and link layer optimization in sensor networks with energy constraints. *ICC*, 2005.
- [3] G. J. Foschini and J. Salz. Digital communications over fading radio channels. *Bell Systems Technical Journal*, pages 429–456, 1983.
- [4] A. J. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.
- [5] B. Johansson and M. Johansson. Mathematical decomposition techniques for distributed cross-layer optimization of data networks. To appear, *IEEE Journal on Selected Areas in Communications*, 2006.
- [6] M. Johansson, L. Xiao, and S. Boyd. Simultaneous routing and resource allocation in CDMA wireless data networks. *ICC*, 2003.
- [7] C. T. Kelley. *Solving Nonlinear Equations with Newton's Method*. SIAM, 2003.
- [8] F. P. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.
- [9] 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. *INFOCOM*, 2004.

- [10] D. O'Neill, D. Julian, and S. Boyd. Adaptive management of network resources. *IEEE Vehicular Technology Conference*, 2003.
- [11] P. Tinnakornrisuphap and C. Lott. On the fairness and stability of the reverse-link MAC layer in CDMA2000 1xEV-DO. *Globecom*, 2004.
- [12] S. Wright. *Primal-dual interior-point methods*. SIAM, 2003.