# Joint Scheduling and Resource Allocation in CDMA Systems

Rajeev Agrawal[1], Vijay Subramanian[1] and Randall Berry[2⋆]

[1] Mathematics of Communication Networks Group, Motorola Inc.,
Arlington Heights, IL, USA
ragrawa1@email.mot.com, vsubram2@email.mot.com

[2] Department of Electrical and Computer Engineering, Northwestern University,
Evanston, IL 60208, USA
rberry@ece.northwestern.edu

**Abstract.** We consider scheduling and resource allocation for the downlink in a CDMA based wireless network. The scheduling and resource allocation problem is to select a subset of the users for transmission and for each of the users selected, to choose the modulation and coding scheme, transmission power, and number of codes used. We refer to this combination as the physical layer operating point (PLOP). Each PLOP consumes different amounts of code and power resources. The resource allocation task to pick the "optimal" PLOP taking into account both system-wide and individual user resource constraints that can arise in a practical system. In this paper, we tackle this problem as part of a utility maximization problem framed in earlier papers that includes both scheduling and resource allocation. Using an information theoretic model for the achievable rate per code results in a tractable convex optimization problem. By exploiting the structure of this problem, we give algorithms for finding the optimal solution with geometric convergence. We also use insights obtained from the optimal solution to construct low complexity near optimal algorithms that are easily implementable. Numerical results comparing these algorithms are also given.

## 1 Introduction

Efficient scheduling and resource allocation are essential for enabling high-speed wireless data services. A variety of wireless scheduling approaches have been purposed that *opportunistically* exploit the temporal variations of wireless channels to improve system performance, e.g. [1–9]. These approaches attempt to transmit to users during periods when they have good channel quality (and can support higher transmission rates), while maintaining some form of fairness.

We consider wireless scheduling for systems in which the transmitter can simultaneously transmit to multiple users in each scheduling interval by using CDMA. In this setting, in addition to deciding which users to schedule, the available physical layer resources, such as bandwidth and power, must be divided among the users. Examples of such systems include the High Speed Downlink Packet Access (HSDPA) approach developed for W-CDMA or the 1x-EVDV architecture for CDMA2000. In these systems, the physical layer resources and information rate assigned to a user are specified by selecting the number of spreading codes, the fraction of transmission power, and the modulation and coding scheme (MCS). We refer to a combination of these as the physical layer operating point (PLOP).

The main problem to be addressed is to specify the optimal PLOP at each scheduling instant, which in turn specifies the vector of transmission rates for each user. Moreover, this problem must be solved once every time-slot (e.g. 2msec in HSDPA), and so requires a computationally efficient solution. We consider this in the context of the gradient-based scheduling framework presented in [2], where the objective is to choose the transmission rate vector that has the largest projection onto the gradient of the total system utility. The utility is a function of each user's throughput and is used to quantify fairness. Several such gradient-based scheduling algorithms have been studied for TDM systems, e.g. [1, 11].

The problem considered here can be viewed as finding the maximum weighted sum throughput for a downlink channel, where the weights are determined by the gradient of the utility. Our solution is

⋆ The work of R. Berry was supported in part by the Northwestern-Motorola Center for Communications and NSF CAREER award CCR-0238382.

general in that it also applies to other scheduling algorithms which provide these weights in a different manner. Maximizing the weighted sum capacity for the downlink channel has been considered from an information theoretic perspective in [12, 13]. This work assumes the use of information theoretic (multi-user) coding/decoding and focuses on the long-term average throughputs.[3] In [13] several sub-optimal strategies are also considered. Here, we restrict ourselves to CDMA systems where all users are orthogonalized; this is similar to one of the sub-optimal approaches in [13]. However, our focus is on the performance in a specific fading state, which leads to simpler algorithms. We also take into account additional "per-user constraints" that are imposed by the capability of each mobile in a practical system.[4]

In Section 2, we begin by formulating the scheduling and resource allocation problem. Using an analytical formula relating the rate, power, codes, and SINR, we obtain an analytically tractable problem with nice convexity properties. In Sect. 4 we consider these properties in detail. Based on the structure of this problem, we then develop a dual algorithm for determining the optimal PLOP. We obtain analytical formulae for many quantities of interest, for others we have to resort to a numerical search. However these numerical searches are in a single dimension (due to the dual formulation) rather than over the multidimensional PLOP space. Also, thanks to the convexity of the problem, these algorithms converge geometrically fast. Along the way we obtain key structural properties of the optimal solution that help speed up the numerical searches as well as help design low complexity near-optimal solutions.

## 2 Gradient-based scheduling and resource allocation problem

We consider the downlink of a wireless communication system with $K$ users. The channel conditions are time-varying and modeled by a stochastic channel state vector $\mathbf{e}_t = (e_{1,t}, \ldots, e_{K,t})$, where $e_{i,t}$ represents the channel state of the $i$th user at time $t$. Associated with each channel state vector is a rate-region $\mathcal{R}(\mathbf{e}_t) \subset \mathbb{R}_+^K$, which indicates the set of feasible transmission rates $\mathbf{r}_t = (r_{1,t}, \ldots, r_{K,t})$.

Our point of departure is the gradient-based scheduling framework in [2]. In this framework, at each scheduling instant a rate vector $\mathbf{r}_t \in \mathcal{R}(\mathbf{e}_t)$ is selected that has the maximum projection onto the gradient of a system utility function $U(\mathbf{W}_t) = \sum_{i=1}^K U_i(W_{i,t})$, where, for each user $i$, $U_i(W_{i,t})$ is a increasing concave utility function of the user's average throughput, $W_{i,t}$, up to time $t$. In other words, the scheduling and resource allocation decision is the solution to

$$\max_{\mathbf{r}_t \in \mathcal{R}(\mathbf{e}_t)} \nabla U(\mathbf{W}_t)^T \cdot \mathbf{r}_t = \max_{\mathbf{r}_t \in \mathcal{R}(\mathbf{e}_t)} \sum_i \dot{U}_i(W_{i,t}) r_{i,t}. \tag{1}$$

Similar, scheduling rules can be developed when the utility depends on other parameters such as buffer size or packet delay.

We note that (1) must be re-solved at each scheduling instant because of changes in both the channel state and the gradient of the utility. The solution depends on the state dependent capacity region $\mathcal{R}(\mathbf{e}_t)$, which we assume is known.[5] In this paper, we consider a model for this region that is appropriate for a CDMA system, such as HSDPA or 1xEVDV. We model this capacity region as being parameterized by two sets of physical layer parameters: the number of spreading codes, $n_i$ and the transmission power $p_i$ assigned to each user $i$. Each choice of these parameters specifies a PLOP, and must satisfy system constraints on the total number of spreading codes ($\sum_i n_i \leq N$) and the total available power ($\sum_i p_i \leq P$) as well as per user constraints on the number of codes that can be assigned to each user $i$ ($n_i \leq N_i$).

We assume that the channel state $e_i$ for user $i$ indicates the user's received signal to interference plus noise ratio (SINR) per unit power, where we have suppressed the dependence on $t$ for convenience. Assuming that all spreading codes are mutually orthogonal (so that the only interference is from other cells), user $i$'s SINR per code is given by $SINR_i = \frac{p_i}{n_i} e_i$. We assume that the achievable rate per code, $\frac{r_i}{n_i} = \Gamma(\zeta_i \cdot SINR_i)$, where $\Gamma$ corresponds to the Shannon capacity for a Gaussian noise channel with the given SINR, i.e., $\Gamma(x) = B \log(1 + x)$, where $B$ indicates the symbol rate (i.e., the chip rate/spreading factor), and $\zeta_i \in (0, 1]$ is a scaling factor that can be used to model the "gap from capacity" in a practical

[3] In the special case of maximizing the equal weight sum capacity in a flat fading channel, the information theoretic optimal approach is to transmit to only one user in each time-slot [12] and hence, multi-user decoding is not required. However, this is not true if the users are not weighted equally or for other channel models, such a multiple antenna channel. It also does not hold when additional per user constraints are present, as is the case here.

[4] Moreover, these constraints may vary from mobile to mobile. For example, the initial mobile devices for HSDPA can receive up to 5 spreading codes, while future devices may be able to receive up to 15 spreading codes.

[5] While, in a practical system, the exact channel state will not be perfectly known at the transmitter, some estimate of it is usually available, for example, via channel quality feedback.

system. This is a reasonable model for systems that use sophisticated coding techniques, such as Turbo codes. Redefining $e_i$ to be $e_i\zeta_i$, the rate region is then given by

$$\mathcal{R}(\mathbf{e}) = \left\{ \mathbf{r} \geq 0 : r_i = n_i B \log\left(1 + \frac{p_i e_i}{n_i}\right), n_i \leq N_i \; \forall i, \sum_i n_i \leq N, \sum_i p_i \leq P \right\}. \tag{2}$$

Notice that in (2), we allow the number of codes per user to take on a non-integer value. Of course, in a practical system these must be integer valued. However, we show that, in most cases, the solution to this relaxed problem results in integer values for $n_i$.

By defining $w_i := \frac{\dot{U}(W_i)B}{\ln 2}$, we can rewrite the optimization problem in (1) as

$$\begin{aligned} V^* &:= \max_{(\mathbf{n},\mathbf{p})\in\mathcal{X}} \; V(\mathbf{n},\mathbf{p}) \qquad [\textbf{Primal problem}] \\ \text{subject to:} \quad & \sum_i n_i \leq N, \quad \sum_i p_i \leq P, \end{aligned} \tag{3}$$

where $V(\mathbf{n},\mathbf{p}) := \sum_i w_i n_i \ln\left(1 + \frac{p_i e_i}{n_i}\right)$, $\mathcal{X} := \{(\mathbf{n},\mathbf{p}) \geq \mathbf{0} : n_i \leq N_i \quad \forall i\}$, $\mathbf{n}$ is a vector of code allocations, $\mathbf{p}$ is a vector of power allocations, $\mathbf{w}$ is the vector of $w_i$'s, and $\mathbf{e}$ is the vector of $e_i$'s. Note that the constraint set $\mathcal{X}$ is convex. It can also be verified that $V$ is concave in $(\mathbf{n},\mathbf{p})$.

In a practical systems there may also be several additional *per user constraints*, such as:

*i*) peak power constraints:

$$p_i \leq P_i, \qquad \forall i.$$

*ii.*) maximum SINR (per code) constraints:

$$SINR_i = \frac{p_i e_i}{n_i} \leq S_i \Leftrightarrow p_i \leq S_i \frac{n_i}{e_i}, \qquad \forall i.$$

*iii.)* maximum rate per code constraints:

$$\frac{r_i}{n_i} = \ln\left(1 + \frac{p_i e_i}{n_i}\right) \leq (R/N)_i \Leftrightarrow p_i \leq (e^{(R/N)_i} - 1)\frac{n_i}{e_i} \qquad \forall i.$$

*iv.*) maximum rate constraints:

$$r_i = n_i \ln\left(1 + \frac{p_i e_i}{n_i}\right) \leq R_i \Leftrightarrow p_i \leq (e^{R_i/n_i} - 1)\frac{n_i}{e_i} \qquad \forall i.$$

These constraints can arise due to various implementation considerations. For example, a constraint on the rate per code is imposed by the maximum rate of the available modulation and coding schemes. A maximum rate constraint arises because there is only a finite amount of data available to send to each mobile at any time. All of the above constraints can be viewed as special cases of a *per user power constraint* with the form $p_i \leq e_i s_i(n_i) n_i$ for all $i$, where $s_i$ is also dependent on the parameters $P_i, S_i, e_i, R_i, (R/N)_i$. We primarily focus on a *SINR type* of per-user power constraint, where $s_i(n_i) \equiv s_i$ does not depend on $n_i$. This corresponds to a limit on the SINR or rate per code. A further special case of this constraint is $s_i(n_i) \equiv s_i = \infty$ which corresponds to no per-user power constraints.

With the per user power constraints, the constraint set $\mathcal{X}$ is further restricted to

$$\mathcal{X} := \{(\mathbf{n},\mathbf{p}) \geq \mathbf{0} : n_i \leq N_i, \quad p_i \leq s_i(n_i) n_i / e_i \quad \forall i\}.$$

Note that for a SINR type of per-user power constraint, this set remains convex.

Additionally, there may also be a constraint on the maximum number of users $M$ scheduled in a time-slot.[6] We will prove later that such a constraint will in most cases automatically be satisfied by the optimal solution as long as $M-1$ users can fully utilize the available code budget, i.e. the sum of the $N_i$'s for any subset of $M-1$ users is greater than or equal to $N$.

[6] For example, in HSDPA such a constraint arises because the system can not schedule more users than the number of control channels.

## 3 The dual problem and convex optimization

We begin considering the solution to (3). Note that if $\sum_i N_i \leq N$, then since $n\ln(1+x/n)$ is increasing in $n$, the optimal code allocation must be $n_i = N_i$ for all $i$. In this case, we are left with just a power optimization problem which can be easily solved. Henceforth, we will assume this is not the case, i.e., $\sum_i N_i > N$.

We solve the optimization problem by looking at the dual formulation. Define the Lagrangian, $L(\mathbf{p}, \mathbf{n}, \lambda, \mu)$ for the primal problem (3) by

$$L(\mathbf{p}, \mathbf{n}, \lambda, \mu) = \sum_i w_i n_i \ln\left(1 + \frac{p_i e_i}{n_i}\right) + \lambda\left(P - \sum_i p_i\right) + \mu\left(N - \sum_i n_i\right). \tag{4}$$

Based on this we can define the dual function

$$L(\lambda, \mu) := \max_{(\mathbf{n},\mathbf{p})\in\mathcal{X}} L(\mathbf{p}, \mathbf{n}, \lambda, \mu). \tag{5}$$

The dual problem is then given by:

$$L^* := \min_{(\lambda,\mu)\geq 0} L(\lambda, \mu) \qquad [\textbf{Dual problem}]. \tag{6}$$

Also, with some further abuse of notation, we define

$$L(\lambda) := \min_{\mu\geq 0} L(\lambda, \mu) = \min_{\mu\geq 0} \max_{(\mathbf{n},\mathbf{p})\in\mathcal{X}} L(\mathbf{p}, \mathbf{n}, \lambda, \mu). \tag{7}$$

From standard convex programming (e.g. Prop. 5.1.2 and 5.1.3 of [14]), we have:

**Proposition 1.** *The dual function $L(\lambda, \mu)$ is convex over the set $\{(\lambda, \mu) \geq \mathbf{0}\}$ and $V^* \leq L(\lambda) \leq L(\lambda, \mu)$ for all $\lambda, \mu \geq 0$.*

Also, it can be shown that this problem satisfies Slater's condition [14], and thus we have:

**Proposition 2.** *There exists at least one Lagrange multiplier and there is no duality gap. The set of Lagrange multipliers is equal to the set of optimal dual solutions. Furthermore, $(\mathbf{p}^*, \mathbf{n}^*)$, $(\lambda^*, \mu^*))$ is an optimal primal solution – Lagrange multiplier pair if and only if*

$$(\mathbf{p}^*, \mathbf{n}^*) \in \mathcal{X}, \quad \sum_i n_i^* \leq N, \quad \sum_i p_i^* \leq P \qquad \text{Primal Feasibility} \tag{8}$$

$$(\lambda^*, \mu^*) \geq 0 \qquad \text{Dual Feasibility} \tag{9}$$

$$(\mathbf{p}^*, \mathbf{n}^*) = \arg\max_{(\mathbf{n},\mathbf{p})\in\mathcal{X}} L(\mathbf{p}, \mathbf{n}, \lambda^*, \mu^*) \qquad \text{Lagrangian Optimality} \tag{10}$$

$$\lambda^*(P - \sum_i p_i^*) = 0, \quad \mu^*(N - \sum_i n_i^*) = 0 \qquad \text{Complementary Slackness} \tag{11}$$

## 4 Structure of the primal and dual problems

We give several properties of the dual problem in (6) and the corresponding primal problem in (3).

### 4.1 Computing the dual function

To begin, we compute the dual function, $L(\lambda, \mu)$ in (5) for a given $\lambda$ and $\mu$. To do this, we first optimize the Lagrangian (4) over $\mathbf{p}$, for a fixed $\lambda$, $\mu$, and $\mathbf{n}$. We then optimize over $\mathbf{n}$ to obtain the value of the dual function. For the first step we have:

**Lemma 1.** *For a fixed feasible $\mathbf{n}$ and any $\lambda \geq 0$ and $\mu \geq 0$, the feasible power allocation $\mathbf{p}^*$ that maximizes $L(\mathbf{p}, \mathbf{n}, \lambda, \mu)$ is given by*

$$p_i^* = \frac{n_i}{e_i} s^*\left(\frac{w_i e_i}{\lambda}, s_i(n_i)\right), \tag{12}$$

*where, $s^*\left(\frac{w_i e_i}{\lambda}, s_i(n_i)\right) := \left[\min\left\{\left(\frac{w_i e_i}{\lambda} - 1\right), s_i(n_i)\right\}\right]^+$, and $[x]^+ = \max(0, x)$.*

This follows directly from the Kuhn-Tucker conditions for the optimization problem. Note the solution is similar to a "water-filling" power allocation across the users [15]. Substituting (12) into Lagrangian, we have

$$L(\mathbf{p}^*, \mathbf{n}, \lambda, \mu) = \sum_i (w_i n_i h(w_i e_i, s_i(n_i), \lambda) - \mu n_i) + \lambda P + \mu N, \tag{13}$$

where

$$h(w_i e_i, s_i(n_i), \lambda) = \ln\left(1 + s^*\left(\frac{w_i e_i}{\lambda}, s_i(n_i)\right)\right) - \frac{1}{w_i e_i} s^*\left(\frac{w_i e_i}{\lambda}, s_i(n_i)\right).$$

With a SINR type per-user power constraint, $h(w_i e_i, s_i(n_i), \lambda)$ does not depend on $n_i$. In this case, the Lagrangian can be easily optimized over $\mathbf{n}$, yielding:

**Lemma 2.** *With a SINR type per-user power constraint, the vector of code allocations, $\mathbf{n}^*$ that maximizes (13) is given by*

$$n_i^* = \begin{cases} 0, & w_i h(w_i e_i, s_i, \lambda) < \mu, \\ N_i, & \mu < w_i h(w_i e_i, s_i, \lambda). \end{cases} \tag{14}$$

*If $\mu = w_i h(w_i e_i, s_i, \lambda)$, every choice of $n_i$ such that $0 \le n_i \le N_i$ maximizes the Lagrangian.*

Substituting this back into the Lagrangian, we have:

**Lemma 3.** *With a SINR type per-user power constraint, the dual function is given by*

$$L(\lambda, \mu) = \sum_i [\mu_i(\lambda) - \mu]^+ N_i + \mu N + \lambda P, \tag{15}$$

*where*

$$\mu_i(\lambda) = w_i h(w_i e_i, s_i, \lambda). \tag{16}$$

### 4.2 Optimizing the dual function

Next we turn to optimizing the dual function. First we consider optimizing over $\mu$, then we consider optimizing over $\lambda$. To begin, we sort the users in decreasing order of $\mu_i(\lambda)$ in (16). Using this ordering, let $j^*$ be the smallest integer such that $\sum_{i=1}^{j^*} N_i \ge N$, and let $N'_{j^*} := N - \sum_{i=1}^{j^*-1} N_i$. Then we have:

**Lemma 4.** *With a SINR type per-user power constraint,*

$$L(\lambda) := \min_{\mu \ge 0} L(\lambda, \mu) = \sum_{i=1}^{j^*-1} \mu_i(\lambda) N_i + \mu_{j^*}(\lambda) N'_{j^*} + \lambda P, \tag{17}$$

*and the minimizing $\mu$ is given by $\mu^*(\lambda) := \mu_{j^*}(\lambda)$.*

Note that $\mu_j(\lambda) \ge \mu_{j+1}(\lambda)$ by the above ordering. Thus $\mu^*(\lambda)$ is a threshold; any user $i$ with $\mu_i(\lambda) > \mu^*(\lambda)$, gets its full code allocation, and those with $\mu_i(\lambda) < \mu^*(\lambda)$ get none.

**Remark:** When $w_i \ge w_j$, $e_i > e_j$, and $s_i \ge s_j$ then $\mu_i(\lambda) \ge \mu_j(\lambda)$, for all $\lambda$. Thus user $i$ will be always be given a full code allocation before allocating any codes to user $j$. Furthermore, assume the scheduling rule is to simply maximize the total throughput. In this case, packing users into the code budget in order of their $e_i$'s is optimal.

We next consider optimizing $L(\lambda)$ over $\lambda \ge 0$. For this we have:

**Lemma 5.** *With a SINR type per user power constraint, $L(\lambda)$ is convex in $\lambda$.*

Since $L(\lambda)$ is a univariate convex function, it can be easily minimized numerically using a bisection type of search with a geometric convergence rate. Also note that, from (12), if $\lambda > w_i e_i$, then user $i$ will be allocated zero power. Therefore, the optimal $\lambda$ must be in the interval $[0, \max_i w_i e_i]$. This provides a starting point for any numerical search. As $\lambda$ decreases from its maximum value, users will initially receive a positive code allocation based on the ordering of $w_i e_i$. Let $\mathbf{n}_0$ denote the resulting code allocation. A simple check can be done to see if $\mathbf{n}_0$ is optimal.

### 4.3 Finding a Lagrangian Optimal Primal Solution

Next, we examine finding primal values $(\mathbf{n}^*, \mathbf{p}^*)$ such that

$$(\mathbf{n}^*, \mathbf{p}^*) = \arg \max_{(\mathbf{n},\mathbf{p})\in\mathcal{X}} L(\mathbf{p}, \mathbf{n}, \lambda, \mu^*(\lambda)), \tag{18}$$

for a given $\lambda > 0$. Here $\mu^*(\lambda)$ is given in Lemma 3. Given the optimal $\lambda = \lambda^*$, then from Prop. 2, such a $(\mathbf{n}^*, \mathbf{p}^*)$ will be optimal for the primal problem if they also satisfies primal feasibility and complimentary slackness. We give a procedure for finding such a pair in the following. If the optimal $\lambda$ is not found, we also use this procedure to find a candidate feasible $\tilde{\mathbf{n}}$. This can serve two purposes. First, we can also construct a primal feasible $\tilde{\mathbf{p}}$ corresponding to $\tilde{\mathbf{n}}$. From Prop. 1,

$$V^* - V(\tilde{\mathbf{n}}, \tilde{\mathbf{p}}) \leq L(\lambda) - V(\tilde{\mathbf{n}}, \tilde{\mathbf{p}}).$$

We can use this as a stopping criteria in the algorithms discussed below. The other use for $\tilde{\mathbf{n}}$ is to find a subgradient for $L(\lambda)$ which can aid in searching for the optimal $\lambda$; this will be discussed in the following.

It can be shown that a solution to (18) is equivalent to finding

$$\mathbf{n}^* = \arg \max_{\{\mathbf{n}\geq 0: n_i \leq N_i \ \forall i\}} \sum_i [\mu_i(\lambda) - \mu^*(\lambda)]^+ n_i, \tag{19}$$

and setting $\mathbf{p}^*$ as in Lemma 1. If there is only one user $i$ such that $\mu_i(\lambda) = \mu^*(\lambda)$, there will be exactly one solution to (19) which satisfies $\sum n_i^* = N$; this can be found by again sorting the users based on $\mu_i(\lambda)$ in (16). The desired solution is then to set $n_i^* = N_i$ for all $i < j^*$, $n_{j^*}^* = N'_{j^*}$, and $n_i^* = 0$ for all $i > j^*$, where $j^*$ and $N'$ are as in Lemma 4. Note that at optimality, if $\mu^*(\lambda^*) > 0$, then to satisfy complementary slackness, it must be that $\sum n_i^* = N$. Even if $\mu^*(\lambda^*) = 0$, an optimal solution satisfying $\sum n_i^* = N$ can still be constructed, however some users will be allocated zero power. Also note that $n_i^*$ in (19) is always an integer allocation.

A scalar $d \in \mathbb{R}$ is a *subgradient* of $L(\lambda)$ at $\lambda$ if

$$L(\tilde{\lambda}) \geq L(\lambda) + (\tilde{\lambda} - \lambda)d \quad \forall\, \tilde{\lambda} \geq 0.$$

For an arbitrary $\lambda$, a solution to (18) that also satisfies $\sum n_i^* = N$ can be used to find a subgradient of $L(\lambda)$.

**Proposition 3.** *Let $(\hat{\mathbf{n}}, \hat{\mathbf{p}})$ satisfy (18) for a given $\lambda$ and $\sum \hat{n}_i = N$. Then $P - \sum_i \hat{p}_i$ is a subgradient of $L(\lambda)$ at $\lambda$.*

When there is a unique $\mathbf{n}^*$ that satisfies the conditions of this proposition, it can be shown that $L(\lambda)$ is differentiable and so it has only one subgradient (its derivative). When there is a tie and more than one $\mu_j(\lambda) = \mu^*(\lambda)$, then there will be multiple $\mathbf{n}^*$ that optimize (19) and satisfy $\sum n_i^* = N$. However, for the optimal $\lambda^*$, every such $\mathbf{n}^*$ may not result in a power allocation that is feasible and satisfies complimentary slackness. For an arbitrary $\lambda$, different choices in $\mathbf{n}^*$ will result in different subgradients for $L(\lambda)$. Next we consider resolving such ties.

Let $\mathcal{I}_\lambda$ denote the set of users involved in a tie for a given $\lambda$, i.e., for all $i \in \mathcal{I}_\lambda$, $\mu_i(\lambda) = \mu^*(\lambda)$. The objective in (19) will not depend on $n_i$, for $i \in \mathcal{I}_\lambda$. First we consider resolving this tie to find the maximum subgradient of $L(\lambda)$ at $\lambda$. It follows from Lemma 3 that this is the solution to the following linear program:

$$\max_{\{n_i | i\in\mathcal{I}_\lambda\}} P_{\text{res}} - \sum_{i\in\mathcal{I}_\lambda} s^*\left(\frac{w_i e_i}{\lambda}, s_i\right)\frac{n_i}{e_i} \qquad [\textbf{LPmax}]$$

$$\text{subject to:} \quad 0 \leq n_i \leq N_i, \quad i \in \mathcal{I}_\lambda$$

$$\sum_{i\in\mathcal{I}_\lambda} n_i = N_{\text{res}}$$

Here, $P_{\text{res}}$ and $N_{\text{res}}$ are the residual power and codes available for the users in the tie. The minimum subgradient can also be found via a linear program given by

$$\min_{\{n_i | i\in\mathcal{I}_\lambda\}} P_{\text{res}} - \sum_{i\in\mathcal{I}_\lambda} s^*\left(\frac{w_i e_i}{\lambda}, s_i\right)\frac{n_i}{e_i}. \qquad [\textbf{LPmin}]$$

subject to the same constraints as in LPmax.

In either of these cases, the structure of the linear program permits a simple greedy solution. Specifically, given an ordering of the users in $\mathcal{I}_\lambda$, we define a *greedy code allocation* for that ordering to be one which sequentially takes each user in the ordering and assigns that user the maximum possible codes until all $N_{\text{res}}$ codes are assigned.

**Lemma 6.** *The code allocation that solves LPmax (LPmin) is given by a greedy code allocation based on ordering the users in $\mathcal{I}_\lambda$ in increasing (decreasing) order of $s^*\left(\frac{w_i e_i}{\lambda}, s_i\right)\frac{1}{e_i}$.*

Let $\hat{\mathbf{n}}$ ($\check{\mathbf{n}}$) be the greedy code allocation for LPmax (LPmin). Finding both of these solutions involves a sort of $\mathcal{I}_\lambda$, and thus each have a complexity of $O(|\mathcal{I}_\lambda| \log(|\mathcal{I}_\lambda|))$. Typically, if a tie occurs, only a small number of users will be involved. Indeed, assuming the parameters $w_i$ and $e_i$ are independently chosen according to an absolutely continuous distribution, then with probability one a tie will not involve more than two users.

If the optimal solution to either LPmax or LPmin is zero, then the corresponding code allocation must be optimal. If the solution to LPmax is negative, then all the subgradients of $L(\lambda)$ at $\lambda$ are negative. Likewise, if the solution to LPmin is positive, then all the subgradients are positive. However, if the solution to LPmax is positive and the solution to LPmin is negative, then $L(\lambda)$ will have a zero subgradient at $\lambda$; a feasible code allocation corresponding to this zero subgradient will be primal optimal. Assuming this is true, there must exist an $\alpha \in [0,1]$ such that

$$\alpha\left(\sum_{i\in\mathcal{I}_\lambda} s^*\left(\frac{w_i e_i}{\lambda}, s_i\right)\frac{\hat{n}_i}{e_i}\right) + (1-\alpha)\left(\sum_{i\in\mathcal{I}_\lambda} s^*\left(\frac{w_i e_i}{\lambda}, s_i\right)\frac{\check{n}_i}{e_i}\right) = P_{\text{res}}.$$

Solving for $\alpha$ and setting $\tilde{n}_i = \alpha\hat{n}_i + (1-\alpha)\check{n}_i$, for all $i \in \mathcal{I}_t$ will gives a primal optimal code allocation.

A special case of the above construction is when

$$N_i \geq N_{\text{res}}, \qquad \forall\, i \in \mathcal{I}_\lambda. \tag{20}$$

This implies that the per-user code constraints will be inactive for any solution to LPmax or LPmin.[7] In this case, the solution to LPmax and LPmin will involve one user each and the above combination will involve only these two users.

**Lemma 7.** *For a SINR type power constraint, an optimal code allocation can be found with the following properties:*

1. *For the case of $N_i = N$ at most two users will be scheduled.*
2. *If (20) holds or at most two users are involved in a tie, then at most $\lceil N/N_{min}\rceil + 1$ users will be scheduled, where $N_{min} := \min_i N_i$. All but two users will have their full code allocation.*

Using typical parameter values for a HSDPA system, this implies that the number of users to be scheduled will be on the order of 1-4.

### 4.4 Optimizing over the powers

Finally, we consider finding the optimal power allocation, $\mathbf{p}$, in the primal problem given a fixed code allocation $\mathbf{n}$, i.e., we want to solve

$$V^*(\mathbf{n}) := \max_{\{\mathbf{p}\geq 0: p_i \leq e_i s_i(n_i) n_i\ \forall i\}} V(\mathbf{n}, \mathbf{p}) \tag{21}$$

subject to $\sum_i p_i \leq P$. This can be solved by finding $\lambda^*(\mathbf{n})$ using a dual formulation and then computing the optimal $\mathbf{p}^*(\mathbf{n})$ as in Lemma 1. The optimal $\lambda$ can be shown to satisfy:

**Lemma 8.** *A given $\lambda$ is the solution to the dual problem of (21) if and only if*

$$\lambda = \frac{\sum_i n_i w_i 1_{\{\frac{w_i e_i}{1+s_i(n_i)} \leq \lambda < w_i e_i\}}}{P - \sum_i \frac{n_i}{e_i} s_i(n_i) 1_{\{\lambda < \frac{w_i e_i}{1+s_i(n_i)}\}} + \sum_i \frac{n_i}{e_i} 1_{\{\frac{w_i e_i}{1+s_i(n_i)} \leq \lambda < w_i e_i\}}}.$$

Based on this lemma, we can develop an iterative search for finding the optimal $\lambda$ for a given code allocation. The algorithm has a complexity of $O(M \log M)$ due to a required sort of the $M$ users with positive code allocations.

[7] In practical system, this condition will often be satisfied. For example, in a HSDPA system with $N = 15$ and $N_i = 15$, 10, or 5 (the same value for all $i$), then whenever only two users are involved in a tie, this condition will be satisfied.

## 5 Algorithms

In this section we discuss algorithms for solving (3). First, we present several variations of optimal algorithms all with a geometric convergence rate. We also discuss several suboptimal algorithms with lower complexity.

### 5.1 Optimal Algorithm

The optimal algorithms we consider are all based on finding the dual optimal solution, $L^*$ in (6), by solving $\min_{\lambda \geq 0} L(\lambda)$, where $L(\lambda)$ is defined in (7). By strong duality this gives us the optimal primal value, $V^*$, and, given the dual optimal $(\lambda^*, \mu^*)$, the primal optimal $(\mathbf{p}^*, \mathbf{n}^*)$ are given by optimizing the Lagrangian as discussed in Sect. 4.3.

For a SINR type per-user power constraint, $L(\lambda)$ is given by Lemma 4. We have shown that this is a univariate convex function, and so can be minimized using a convex search technique. To begin, we consider a bisection method, where at the $k$th iteration, the algorithm identifies a range $[\lambda_k^{LB}, \lambda_k^{UB}]$ known to contain the optimal $\lambda^*$. We also identify an estimate of $\lambda^*$ given by $\lambda_k \in [\lambda_k^{LB}, \lambda_k^{UB}]$. These parameters are updated from iteration to iteration, by considering a candidate $\lambda_k^{cand}$ in either $[\lambda_k^{LB}, \lambda_k]$ or $[\lambda_k, \lambda_k^{UB}]$, and then updating these parameters, depending on the relative values of $L(\lambda)$. Choosing $\lambda_k^{cand}$ as the midpoint of the larger sub-interval ensures geometric convergence to the optimal dual solution. Each iteration requires evaluating $L(\lambda)$; using Lemma 4, this has a complexity of $O(K \log(K))$ due to the required sort based on $\mu_i(\lambda)$. As discussed in Section 4.2, we can use the points $\lambda_{min} = 0$ and $\lambda_{max} = \max_i w_i e_i$ to begin our search. This provides a basic optimal algorithm; next we discuss several enhancements, which further exploit the structure of the problem.

The first enhancement we consider is based on initially checking if the code allocation $\mathbf{n}_0$ discussed in Section 4.2 is optimal. If so, we need simply calculate the optimal primal power allocation, $\mathbf{p}^*(\mathbf{n}_0)$ and we are done. If $\mathbf{n}_0$ is not optimal, this provides a tighter upper-bound on $\lambda$ for beginning our search.

The next enhancement we consider is evaluate a feasible primal solution $\mathbf{n}_k = \mathbf{n}^*(\lambda_k)$ as in Section 4.3, for each iteration $k$. This serves two the following two purposes:

**1. Stopping Criterion**: It can be used for a stopping criteria. Two possibilities are:

*a.*) Calculate a primal feasible $\mathbf{p}_k = \mathbf{p}^*(\mathbf{n}_k)$, as in Section 4.4 and stop when the primal value and the dual value are sufficiently close.

*b.*) Calculate a power allocation $\mathbf{p}_k$ as given by Lemma 1 and stop when $|P - \sum_i p_{i,k}| < \epsilon$. From Prop. 3, this stopping criteria checks if the subgradient is near zero. Note that $\mathbf{p}_k$ is different from $\mathbf{p}^*(\mathbf{n}_k)$.

**2. Update** $\lambda_k$: The second use for $\mathbf{n}_k$ is as a guide for picking the next $\lambda_k^{cand}$. Once again there are several possibilities; we give two that correspond to the cases (a.) and (b.) above.

*a.*) For case (a.), we consider $\lambda_k^{cand} = \lambda^*(\mathbf{n}^*(\lambda_k))$, where $\lambda^*(\mathbf{n})$ is the optimal $\lambda$ for the given code allocation, and $\mathbf{n}^*(\lambda)$ is the optimal code allocation for the given $\lambda$. If $\lambda_k^{cand} \in [\lambda_k^{LB}, \lambda_k^{UB}]$, we can consider it instead of the bisection point of a sub-interval.[8] Evaluating this map using the iteration discussed after Lemma 8 has a complexity of $O(M \log M)$.

*b.*) For case (b.), we can use the subgradient $d_k = P - \sum_i p_{i,k}$ to aid in choosing the next candidate $\lambda$. In particular, if $d_k < 0$ then the optimal $\lambda$ must lie in $[\lambda_k^*, \lambda_k^{UB}]$, and if $d_k > 0$ then the optimal $\lambda$ must lie in $[\lambda_k^{LB}, \lambda_k^*]$.

Combining these steps, we have an optimal algorithm with the basic structure shown in Fig. 1. The stopping criterion check and updating steps can be performed in either of the ways discussed above.

### 5.2 Suboptimal Algorithms

We briefly mention two sub-optimal approaches. The first, we refer to as a *truncated optimal* algorithm begins by generating several initial code allocations based on packing the code budget using different heuristic sort metrics, (e.g., ordering the users based on $w_i e_i$). Given the set of initial feasible code allocations, we then select the allocation with the maximum primal value. This allocation is then updated by applying the optimal algorithm in Fig. 1 for a fixed number of iterations; in the following simulations, we used only 1 iteration.

The second sup-optimal approach we refer to as the *greedy baseline algorithm*. This algorithm is based on splitting the scheduling decision and the resource allocation into two parts. First a scheduling order for the users is found, using some heuristic sort metric. Given the scheduling order, the resource allocation

[8] Geometric convergence can still be guaranteed by only considering $\lambda_k^{cand}$ is chosen such that it is sufficiently in the interior $[\lambda_k^{LB}, \lambda_k^{UB}]$ so the current interval will be reduced by a given percentage.

1. IF $\mathbf{n}_0$ is optimal, THEN END.
2. Initialize $\lambda_0^{LB}, \lambda_0^{UB}, \lambda_0$.
3. Set $k = 0$, $\mathbf{n}_k = \mathbf{n}^*(\lambda_0)$, and **Choose** $\mathbf{p}_k$.
4. WHILE **Stopping Criterion** fails DO
   i k = k +1;
   ii **Update** $\lambda_k$ and thereafter update $\lambda_k^{LB} and \lambda_k^{UB}$;
   iii Calculate $\mathbf{n}_k = \mathbf{n}^*(\lambda_k)$.
5. END WHILE

**Figure 1.** Basic structure of optimal algorithm.

is then done by taking each user in order and choosing a PLOP that maximizes the transmission rate the user can receive, using the residual power and codes that are available.

## 6 Numerical Results

In this section we provide simulation results for the optimal and sub-optimal algorithms discussed above. We consider a single cell system with 40 users and other parameters chosen to match a HSDPA system; in particular, we set $N = 15$, $N_i = 5$, and $P = 11.9W$. As in [1], we assign each user a utility of the form $U_i(W_i) = \frac{1}{\alpha}(W_i)^\alpha$, where $\alpha \leq 1$ is a fairness parameter. When $\alpha = 0$, we set $U_i(W_i) = \log(W_i)$; this corresponds to a proportional fair rule. We then simulate the combined scheduling and resource allocation algorithms using a realistic single cell model that includes both large-scale and small-scale fading. In Table 1, we give several performance metrics for each algorithm and for different choices of the fairness parameter $\alpha$. Shown are the time average utility, the time-average log utility (this can be used to compare the throughputs of different utility functions), the average number of users $M$ scheduled per time-slot, the average number of codes used $N_s$, the average power used per time-slot, $P_s$, and the sector throughput. Also, in Figure 1, we show the empirical CDF of the user throughput for each algorithm in the $\alpha = 0$ case.

**Table 1.** Simulation Results

| $\alpha$ | Algorithm | Utility | Log Utility | $M$ | $N_s$ | $P_s$ | Sector Throughput (Mbps) |
|---|---|---|---|---|---|---|---|
| 0.0 | Optimal | 231.944 | 231.944 | 3.35461 | 15 | 11.8997 | 8.8145 |
| 0.0 | Truncated optimal | 229.282 | 229.282 | 3 | 15 | 11.2689 | 7.87875 |
| 0.0 | Greedy baseline | 222.222 | 222.222 | 3 | 15 | 10.9659 | 6.36075 |
| 0.25 | Optimal | 173.646 | 231.669 | 3.33331 | 15 | 11.8998 | 9.28545 |
| 0.25 | Truncated optimal | 170.275 | 228.886 | 3 | 15 | 10.7793 | 8.54505 |
| 0.25 | Greedy baseline | 163.798 | 222.663 | 3 | 15 | 10.6948 | 7.2903 |
| 0.5 | Optimal | 806.085 | 228.404 | 3.36408 | 15 | 11.899 | 11.1392 |
| 0.5 | Truncated optimal | 749.531 | 224.379 | 3 | 15 | 9.83421 | 9.127 |
| 0.5 | Greedy baseline | 725.4 | 220.801 | 3 | 15 | 9.72985 | 8.6008 |
| 0.75 | Optimal | 4129.16 | 213.411 | 3.36341 | 15 | 11.8903 | 12.6934 |
| 0.75 | Truncated optimal | 3579.71 | 207.866 | 3 | 15 | 7.82554 | 10.1799 |
| 0.75 | Greedy baseline | 3538.96 | 201.87 | 3 | 15 | 7.79743 | 10.2524 |

In these results, the optimal algorithm gives a higher utility as well as a higher sector throughput compared to the other algorithms. For the $\alpha = 0$ case (proportional fair) we get a 34% improvement over the greedy baseline algorithm. The truncated optimal algorithm is close to optimal and usually also gives a higher sector throughput than the greedy baseline algorithm. For $\alpha = 0$, we get a 23.87% improvement over the greedy baseline algorithm. Furthermore, not only is sector throughput higher for the optimal algorithm, but in fact, from Fig. 2 we see that all user throughputs are larger (in a stochastic ordering

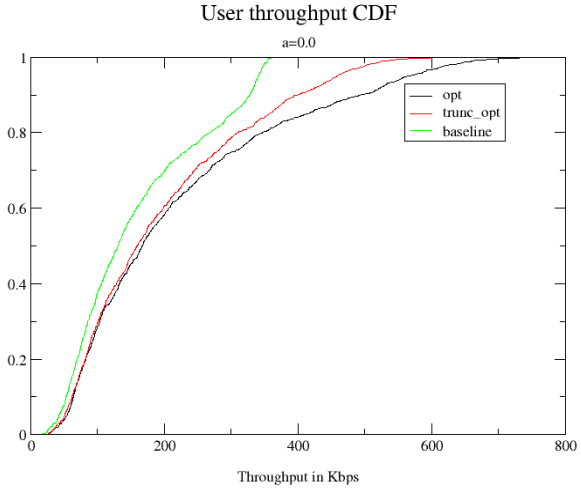


**Figure 2.** Empirical CDF of users throughputs for $\alpha = 0$.

sense). In general, the optimal is better than truncated optimal which, in turn, is better than the greedy baseline when we compare user throughputs. We also observe that the optimal algorithm schedules 3 or 4 users whereas the other algorithms only schedule 3 users. From Table 1, we see that the optimal algorithm does a better job of filling the power budget and that all algorithms used up all the codes.

## 7 Conclusions

In this paper we studied optimally allocating codes and power for the downlink of a CDMA system. The objective was to maximize the weighted sum throughput, where the weights were determined by a gradient-based scheduling algorithm. By formulating this as a convex optimization problem, we were able use a dual approach to characterize the optimal solution and develop efficient optimal and sub-optimal algorithms. Numerical results show that these algorithms can yield better performance than a greedy baseline approach which splits the scheduling and resource allocation into two steps.

## References

1. R. Agrawal, A. Bedekar, R. La, V. Subramanian, "A Class and Channel-Condition based Weighted Proportionally Fair Scheduler," *Proc. of ITC 2001*, Salvador, Brazil, Sept. 2001.
2. R. Agrawal and V. Subramanian, "Optimality of Certain Channel Aware Scheduling Policies," *Proc. of 2002 Allerton Conference on Communication, Control and Computing*, Oct. 2002.
3. L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queue with randomly varying connectivity", in *IEEE Transactions on Information Theory*, Vol. 39, pp. 466-478, March 1993.
4. R. Leelahakriengkrai and R. Agrawal, "Scheduling in Multimedia CDMA Wireless Networks," *IEEE Trans. on Vehicular Technology*, 2002.
5. M. Andrews, K. Kumaran, K. Ramanan, A. L. Stolyar, R. Vijayakumar and P. Whiting, "Providing Quality of Service over a Shared Wireless Link", in *IEEE Communications Magazine*, pp.150-154, 2001, Vol.39, No.2.
6. S. Shakkottai and A. L. Stolyar, "Scheduling algorithms for a mixture of real-time and non-real-time data in HDR", in *Proceedings of the 17th International Teletraffic Congress*, pp. 793-804, Salvador da Bahia, Brazil, 24-28 Sept., 2001.
7. Y. Liu and E. Knightly, "Opportunistic Fair Scheduling over Multiple Wireless Channels", in *Proc. of IEEE INFOCOM*, San Francisco, CA, March 2003.
8. X. Liu, E. K. P. Chong, and N. Shroff, "Opportunistic transmission scheduling with resource sharing constraints in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, Oct. 2001.
9. P. Liu, R. Berry, and M. Honig, "Delay-Sensitive Packet Scheduling in Wireless Networks," *Proc. of IEEE WCNC 2003*, New Orleans, LA, March 16-20, 2003.
10. P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana and A. Viterbi. "CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users", in *IEEE Commun. Mag.*, pp. 70-77, July 2000
11. A. Jalali, R. Padovani, R. Pankaj, "Data throughput of CDMA-HDR a high efficiency - high data rate personal communication wireless system.", in *Proc. VTC '2000*, Spring, 2000.
12. D. Tse, "Optimal Power Allocation over Parallel Gaussian Broadcast Channels," *Proc. of ISIT*, 1997.
13. L. Li and A. Goldsmith, "Optimal Resource Allocation for Fading Broadcast Channels- Part I: Ergodic Capacity," *IEEE Trans. on Information Theory*, March 2001.
14. D. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.
15. R. Gallager, *Information Theory and Reliable Communication*, John Wiley and Sons, 1968.