Joint Scheduling and Resource Allocation in Uplink OFDM Systems

Jianwei Huang, Vijay G. Subramanian, Rajeev Agrawal, and Randall Berry

Abstract

Orthogonal Frequency Division Multiplexing (OFDM) with dynamic scheduling and resource allocation is widely considered to be a key component of most emerging broadband wireless access networks, such as WiMAX. However, scheduling and resource allocation in an OFDM system is complicated, especially in the uplink due to two reasons: (1) the discrete nature of subchannel assignments, and (2) the heterogeneity of the users’ subchannel conditions, individual resource constraints and application requirements. We approach this problem using a gradient-based scheduling framework presented in previous work. Physical layer resources (bandwidth and power) are allocated to maximize the projection onto the gradient of a total system utility function which models application-layer Quality of Service (QoS). This is formulated as a convex optimization problem. We present an optimal solution using a dual decomposition. This solution has prohibitively high computational complexity but reveals guiding principles that we use to generate several lower complexity sub-optimal algorithms. We compare the performance of these algorithms via a realistic OFDM simulator.

J. Huang is with the Dept. of Information Engineering, Chinese University of Hong Kong, e-mail: jwhuang@ie.cuhk.edu.hk. V. G Subramanian is with the Hamilton Institute, National University of Ireland, email: vijay.subramanian@nuim.ie. R. Agrawal is with the Advanced Networks and Performance Dept., Motorola Inc., e-mail: Rajeev.Agrawal@motorola.com. R. Berry is with the Dept. of EECS, Northwestern University, e-mail: rberry@ece.northwestern.edu.

Part of this work was done while J. Huang and V. G. Subramanian were at Motorola. J. Huang is supported in part by Direct Grant of the Chinese Univ. of Hong Kong under Grant 2050398. V. Subramanian is supported by SFI grant 03/IN3/I346. R. Berry was supported in part by the Motorola-Northwestern Center for Seamless Communications and NSF CAREER award CCR-0238382. The work was partially presented at the 2007 Asilomar Conference on Signals, Systems and Computers.
I. INTRODUCTION

This paper analyzes the uplink scheduling problem for OFDM wireless access networks. The specific problem is motivated by the WiMAX/802.16e standard\(^1\) where there is a centralized scheduler that knows the QoS classes, queue-lengths and delays of the packets queued on each mobile device. The WiMAX/802.16e standard specifies reserved time-frequency slots for communicating this information to the scheduler and for conveying the scheduling decisions to the mobiles, both with low delays.

Using OFDM on the uplink of an access network with dynamic scheduling and resource allocation has only recently attracted significant attention. Thus the literature on this subject is still in a nascent state [14], [18].\(^2\) This problem is precisely stated in Section II. We highlight two challenging aspects of this problem. First, the discrete nature of subchannel assignments in OFDM systems usually leads to hard integer programming problems. Second, the per-user power constraint that arises in the uplink makes the problem even less tractable. We initially consider a mathematical abstraction in which multiple users can share one subchannel/tone using orthogonalization (e.g. via time-sharing \(^3\)), which relaxes the integer constraints. In Section III we derive an optimal solution to this relaxed problem using a dual decomposition. This provides insight into the structure of an optimal solution; however, due to the per-user power constraints determining this solution has high computational complexity. In Section V we use the insights gained from the optimal solution to propose a family of sub-optimal algorithms that also take into account the integer constraint of one user per subchannel/tone. Finally, in Section VI we present numerical results for these algorithms using a realistic OFDM simulator.

II. PROBLEM STATEMENT

We consider a model for uplink scheduling in an OFDM system that is based on our previous work on downlink scheduling in CDMA systems [3] and OFDM systems [4]. Specifically, in

---

\(^1\)LTE for 3GPP and 3GPP2 and the FLASH OFDM system from Qualcomm Flarion also fit the model we consider in this paper. Furthermore, this model is applicable for both FDD and TDD systems.

\(^2\)The downlink version of this problem has received more attention, but as we discuss later, the uplink version of the problem introduces several new dimensions.

\(^3\)While super-position coding would yield an even larger (and more tractable) capacity region, we do not consider it, as it is still not practical.
TABLE I
KEY NOTATIONS

<table>
<thead>
<tr>
<th>Notation</th>
<th>Physical Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>total number of subchannels</td>
</tr>
<tr>
<td>$\mathcal{N}$</td>
<td>set of all subchannels</td>
</tr>
<tr>
<td>$M$</td>
<td>total number of users</td>
</tr>
<tr>
<td>$\mathcal{M}$</td>
<td>set of all users</td>
</tr>
<tr>
<td>$w_i$</td>
<td>user $i$’s (dynamic) weight</td>
</tr>
<tr>
<td>$e_{ij}$</td>
<td>normalized SINR on subchannel $j$ for user $i$</td>
</tr>
<tr>
<td>$p_{ij}$</td>
<td>power allocated on subchannel $j$ for user $i$</td>
</tr>
<tr>
<td>$x_{ij}$</td>
<td>fraction of subchannel $j$ allocated to user $i$</td>
</tr>
<tr>
<td>$P_i$</td>
<td>maximum transmit power for user $i$</td>
</tr>
</tbody>
</table>

every scheduling epoch the scheduler seeks to maximize a (time-varying) weighted sum of the users’ rates over a given (time-varying) rate-region. We begin by describing this rate-region. The key notations are listed in Table I; we use bold symbols to denote vectors and matrices of these quantities, e.g., $w = \{w_i, \forall i\}$, $e = \{e_{ij}, \forall i, j\}$, $p = \{p_{ij}, \forall i, j\}$, and $x = \{x_{ij}, \forall i, j\}$.

We assume that the scheduler has knowledge of the received Signal-to-noise ratio (SNR) $e_{ij}$ per unit power for every user and tone. In both FDD and TDD systems this can be obtained using a combination of measurements made on the UL pilots as well as past transmissions from the mobiles.  

Let $\mathcal{R}(e_t)$ denote the feasible rate region at time $t$. We model this as

$$
\mathcal{R}(e_t) = \left\{ r \in \mathbb{R}_+^M : r_i = \sum_{j \in \mathcal{N}} x_{ij} \log \left( 1 + \frac{p_{ij} e_{ij}}{x_{ij}} \right), \ \forall i \in \mathcal{M} \right\},
$$

In both FDD and TDD systems this can be obtained using a combination of measurements made on the UL pilots as well as past transmissions from the mobiles.
where \((x, p) \in \mathcal{X}\) are chosen subject to

\[
\sum_i x_{ij} \leq 1, \forall j \in \mathcal{N},
\]

\[
\sum_j p_{ij} \leq P_i, \forall i \in \mathcal{M},
\]

and the set

\[
\mathcal{X} := \left\{ (x, p) \geq 0 : 0 \leq x_{ij} \leq 1, \ p_{ij} \leq \frac{x_{ij}s_{ij}}{e_{ij}} \forall i, j \right\}.
\]

Here, \(s_{ij}\) is a maximum SNR constraint on tone \(j\) for user \(i\), modeling for example a limit on the available modulation order.

In practical OFDM systems, \(x_{ij}\) is constrained to be an integer, in which case we add the additional constraint \(x_{ij} \in \{0, 1\}\) for all \(i, j\). Initially, we ignore this constraint; this corresponds to a system in which users can share each tone. If resource allocation is done on blocks of OFDM symbols, then fractional values of \(x_{ij}\) can be implemented by time-sharing the symbols in a block.\(^5\) We will bring such integer constraints back to the problem in Sections IV and V.

Next we formulate the scheduling and resource allocation problem. Our approach is based on the gradient-based scheduling framework presented in [2], [10], [11]. Each user \(i\) is assigned a utility function \(U_i(W_{i,t}, Q_{i,t})\) depending on their average throughput \(W_{i,t}\) up to time \(t\) and their queue-length \(Q_{i,t}\) at time \(t\). This is used to quantify fairness and ensure stability of the queues. During each scheduling epoch \(t\), the system objective is to choose a rate vector \(r_t\) in \(\mathcal{R}(e_t)\) that maximizes a (dynamic) weighted sum of the users’ rates, where the weights are determined by the gradient of the sum utility across all users. More precisely, the scheduler seeks to maximize the projection of \(r_t\) onto the gradient

\[
\nabla_w U(W_t, Q_t) - \nabla_q U(W_t, Q_t),
\]

where

\[
U(W_t, Q_t) = \sum_{i=1}^{\mathcal{K}} U_i(W_{i,t}, Q_{i,t}).
\]

We further assume that for each user \(i\),

\[
U_i(W_{i,t}, Q_{i,t}) = u_i(W_{i,t}) - \frac{d_i}{p} (Q_{i,t})^p,
\]

\(^5\)Likewise, if the number of subchannels are large enough so that the subchannel gains do not change dramatically among adjacent subchannels, then the fractional value of \(x_{ij}\) can also implemented by frequency sharing (e.g., [18]).
where $u_i(W_{i,t})$ is an increasing concave function, $d_i \geq 0$ is a QoS weight for user $i$'s queue length, and $p > 1$ is a fairness parameter associated with the queue length. Hence, the scheduling and resource allocation decision is the solution to

\[
\max_{r_t \in \mathcal{R}(e_t)} \left( \nabla_w U(W_t, Q_t)^T - \nabla_q U(W_t, Q_t)^T \right) \cdot r_t = \max_{r_t \in \mathcal{R}(e_t)} \sum_i \left( \frac{\partial u_i(W_{i,t})}{\partial W_{i,t}} + d_i(Q_{i,t})^{p-1} \right) r_{i,t}.
\]

(5)

Several variations of the policy in (5) have been studied. If $d_i = 0$ for all $i \in \mathcal{M}$, the resulting policy has been shown to yield utility maximizing solutions [2], [10], [11]. If $u_i(\cdot) \equiv 0$ with $d_i > 0$ for all $i \in \mathcal{M}$ then this policy has been shown to be stabilizing in a variety of settings [5]–[7]. A specific choice of $d_i$ for “usual” utility functions $u_i(\cdot)$ has been shown to produce utility maximizing solutions subject to stability [9].

As a concrete example, one class of utility functions typically used (e.g. [1], [12]) for $u_i(\cdot)$ is

\[
u_i(W_{i,t}) = \begin{cases} \frac{c_i}{\alpha} (W_{i,t})^{\alpha}, & \alpha \leq 1, \; \alpha \neq 0 \\ c_i \log(W_{i,t}), & \alpha = 0, \end{cases}
\]

(6)

where $\alpha \leq 1$ is a fairness parameter and $c_i \geq 0$ is a QoS weight. In this case, the objective in (5) becomes $\sum_i (c_i(W_{i,t})^{\alpha-1} + d_i(Q_{i,t})^{p-1}) r_{i,t}$. With zero queue weights $d_i$ and equal throughput weights $c_i$, setting $\alpha = 1$ results in a “maximum throughput” scheduling rule that maximizes the total throughput during each slot. For $\alpha = 0$, this results in the proportional fair rule [8].

The optimization in (5) can be written as

\[
\max_{r_t \in \mathcal{R}(e_t)} \sum_i w_{i,t} r_{i,t},
\]

(7)

where $w_{i,t} \geq 0$ is a time-varying weight assigned to the $i$th user at time $t$. Our focus in this paper is on solving such a problem for an uplink OFDM system, i.e., when $\mathcal{R}(e_t)$ is given by (1). We emphasize that (7) must be re-solved at each scheduling instant because of changes in both the subchannel state, $e_t$, and the weights (e.g., the gradient of the utility). We also note that while in the above examples, the weights $w_i$ were given by the gradient of the utility function, our algorithms also apply if other methods for generating these weights are used.
III. OPTIMAL SOLUTION WITH FRACTIONAL ALLOCATIONS

In this section we consider the optimal solution to (7) when \( R(e_t) \) is given by (1). This problem can be written as

\[
\max_{(x,p) \in X} \sum_{i \in M} w_i \sum_{j \in N} x_{ij} \log \left( 1 + \frac{p_{ij}e_{ij}}{x_{ij}} \right)
\]  

(UL)

subject to the per subchannel assignment constraints in (2) and the per user power constraints in (3), where \( X \) is given in (4).

It can be shown that Problem UL has no duality gap and so we solve it by considering a dual formulation. We associate dual variables \( \lambda = (\lambda_i)_{i \in M} \) with constraints (3) and \( \mu = (\mu_j)_{j \in N} \) with constraints (2), resulting in the Lagrangian,

\[
L(\lambda, \mu, x, p) := \sum_{i,j} w_i x_{ij} \log \left( 1 + \frac{p_{ij}e_{ij}}{x_{ij}} \right) + \sum_i \lambda_i (P_i - \sum_j p_{ij}) + \sum_j \mu_j (1 - \sum_i x_{ij}).
\]  

(8)

From duality theory, it follows that the optimal solution to Problem UL is given by

\[
\min_{(\lambda,\mu) \geq 0} \max_{(x,p) \in X} L(\lambda, \mu, x, p).
\]  

(9)

Next we solve this by first analytically finding the optimal \( p \) and \( x \) given fixed values of the dual variables. We then show that the optimal \( \mu \) is given by a performing a search for the maximum value of a per-user metric on each subchannel. The final step is to numerically search for the optimal value of \( \lambda \).

Optimizing \( L(\lambda, \mu, x, p) \) over \( p \) given \( x, \mu \) and \( \lambda \), we get

\[
p^*_{ij} = \frac{x_{ij}}{e_{ij}} \min \left\{ \left( \frac{w_i e_{ij}}{\lambda_i} - 1 \right)^+, s_{ij} \right\},
\]  

(10)

where \( \{x\}^+ = \max\{x, 0\} \). Note that unless \( \sum_{j \in N} \frac{x_{ij}s_{ij}}{e_{ij}} < P_i \), it will always be that \( \sum_{j \in N} p^*_{ij} = P_i \). Assuming this is the case, (10) is a water-filling solution which takes into account the maximum SINR constraint. Substituting \( p^* \) into \( L(\cdot,\cdot,\cdot,\cdot) \) yields

\[
L(\lambda, \mu, x, p^*) = \sum_{i,j} x_{ij} (w_i h(\lambda_i, w_i e_{ij}, s_{ij}) - \mu_j) + \sum_j \mu_j + \sum_i \lambda_i P_i,
\]  

(11)

where we have used the function \( h(\cdot,\cdot,\cdot) \) from [3], namely,

\[
h(a, b, c) = \begin{cases} 
0 & \text{if } a \geq b; \\
\frac{a}{b} - 1 - \log \frac{a}{b} & \text{if } \frac{b}{1+c} \leq a < b; \\
\log(1+c) - \frac{a}{b}c & \text{if } a < \frac{b}{1+c},
\end{cases}
\]  

(12)
where \( a \geq 0, b > 0 \) and \( c \geq 0 \). Optimizing (11) over \( x \) such that \( x_{ij} \in [0, 1] \) yields

\[
L(\lambda, \mu, x^*, p^*) = \sum_{ij} (w_i h(\lambda_i, w_i e_{ij}, s_{ij}) - \mu_j)^+ + \sum_j \mu_j + \sum_i \lambda_i P_i,
\]

where the subchannel allocation has the following structure

\[
x^*_{ij}(\mu_j) = \begin{cases} 
1, & \text{if } w_i h(\lambda_i, w_i e_{ij}, s_{ij}) > \mu_j; \\
[0, 1], & \text{if } w_i h(\lambda_i, w_i e_{ij}, s_{ij}) = \mu_j; \\
0, & \text{if } w_i h(\lambda_i, w_i e_{ij}, s_{ij}) < \mu_j.
\end{cases}
\]

(14)

Since the cost function in (13) is separable, minimizing \( L(\lambda, \mu, x^*, p^*) \) to obtain the optimal \( \mu^*_j(\lambda) \) requires a simple sort per subchannel similar as that in [3], namely,

\[
\mu^*_j(\lambda) = \max_i \mu_{ij}(\lambda_i),
\]

where \( \mu_{ij}(\cdot) := w_i h(\cdot, w_i e_{ij}, s_{ij}) \).

From (14) and (15), it is clear that \( x^*_{ij}(\mu^*_j(\lambda)) \equiv 0 \) if \( i \not\in \arg \max_{i \in M} \mu_{ij}(\lambda_i) \), i.e., there is a per subchannel metric such that any user who does not maximize this metric on a given subchannel will not be allocated the subchannel. There will be ties when multiple users achieve the same value of \( \mu^*_j \) on subchannel \( j \). These can be broken arbitrarily to obtain the correct value for \( L(\lambda, \mu^*, x^*, p^*) \). Now substituting \( \mu^* \) into \( L(\lambda, \mu, x^*, p^*) \), and noticing that \( \mu, x^*, p^* \) are all functions of \( \lambda \), we have

\[
L(\lambda) := L(\lambda, \mu^*, x^*, p^*) = \sum_j \max_i \mu_{ij}(\lambda_i) + \sum i \lambda_i P_i.
\]

The solution to (9) is given by minimizing \( L(\lambda) \) over \( \lambda \geq 0 \). For this we use a sub-gradient-based search, i.e.,

\[
\lambda_i(t + 1) = \left[ \lambda_i(t) - \kappa(t) \left( P_i - \sum_j p^*_i(t) \right) \right]^+, \forall i \in M.
\]

The algorithm will converge when \( \kappa(t) \) is chosen sufficiently small [15]. The detailed algorithm is given in Appendix A. Given an optimal \( \lambda^* \), by duality, \( L(\lambda^*) \) is the optimal objective value to Problem UL.

However, to implement this algorithm, the scheduler must specify the corresponding optimal primal values of \( (x^*, p^*) \). Here, as in [4], more care is required. Specifically, when ties occur
in (15), it is often needed to split the subchannel among several users (i.e., allowing fraction values of $x^*$). Following a similar approach as in [3], the optimal fraction values can be found by solving a linear program whose size increases with the number of users and tones involved in each tie. As we discuss below, this number can be quite large in the uplink setting. Moreover, as noted above one is typically interested in an integer allocation in practice, i.e. each subchannel can be allocated to at most one user. We consider this problem next.

IV. INTEGER SUBCHANNEL ALLOCATION BASED ON OPTIMAL ALGORITHM

We now address the problem:

$$\max_{(x,p) \in X, x_{ij} \in \{0,1\}, \forall i,j} \sum_{i \in M} w_i \sum_{j \in N} x_{ij} \log \left(1 + \frac{p_{ij}e_{ij}}{x_{ij}}\right)$$

(UL-Integer)

Initially, consider the following heuristic for Problem UL-Integer: (i) Solve Problem UL as in the previous section, and (ii) “break” any ties on all subchannels, i.e., whenever there is a fractional $x^*_{ij}$ value, choose one user in the tie and allocate subchannel $j$ to that user only. Clearly, if there are no ties to break, this algorithm gives the optimal solution to Problem UL-Integer. After all ties are broken, we can then re-optimize the power allocation for each user so that each individual power constraint is tight.6

In [4] we used a similar algorithm to solve a downlink OFDM scheduling problem (i.e., we solved the dual problem by allowing fractional subchannel allocation first, and then broke any ties to find an integer subchannel allocation). However, there are several major differences between the uplink and downlink setting which make this approach less appealing for implementation in the uplink setting. First, in the downlink case there is a single power constraint $\sum_{i,j} p_{ij} \leq P$ for the base station instead of the per-user power constraints in (3). Hence, in the downlink case $L(\lambda)$ is a function of only a single dual variable $\lambda$, which simplifies the numerical search for the optimal $\lambda$. In the uplink setting, the convergence of the subgradient search is too slow to be useful for scheduling on a fast time-scale.

Second, even if the optimal $\lambda$ can be found tie breaking is more difficult than in the downlink case. One reason for this is that having a single dual variable provides simple tie-breaking rules for the down-link based on the (scalar) subgradients of $L(\lambda)$ (see [4]); in the uplink case, the

6This can be accomplished using a finite time “water-filling” algorithm as in [4].
subgradients are vectors, complicating such an approach. Also, the uplink case can be more sensitive to how ties are resolved. For example, if two users, \( i \) and \( l \), have the same weights \((w_i = w_l)\) and the same gains on subchannel \( j \) \((e_{ij} = e_{lj})\), then allocating subchannel \( j \) to either user yields the same total weighted rate and the same total power usage in the downlink case. On the other hand, different allocations lead to different individual power consumptions in the uplink case, and thus may lead to different solutions.

Finally, the number of ties is typically much larger in the uplink case than in the downlink case. Consider a simple scenario with two users and two subchannels. Each user has the same gain over both subchannels, i.e., \( e_{i1} = e_{i2} = e_i \) for \( i = 1, 2 \), and \( P = P_1 = P_2 \), where \( P \) is the total power constraint in the downlink case. Assume user 2 has a much better subchannel than user 1 so that in the downlink case, the unique optimal solution is to allocate both subchannels to user 2, and there is no tie. However, in the uplink case, it can be shown that at the optimal dual solution, \( \lambda_1 \) and \( \lambda_2 \) will satisfy

\[
\mu_{1j}(\lambda_1) = \mu_{2j}(\lambda_2) \quad \text{for} \quad j = 1, 2,
\]

i.e., there is a tie in each subchannel and we have to compare four possible subchannel allocations to find the optimal solution. This can be easily extended to \( M \) users and \( N \) subchannels, with each user having the same gain over all its subchannels. This results in \( M^N \) ties, independent of the variation in gains across users.

V. LOW COMPLEXITY SUBOPTIMAL ALGORITHMS

Due to the effects discussed in the previous section, solving for the optimal dual solution to Problem UL and breaking any ties is to get an integer allocation is not computationally feasible for even a moderately sized system. We now present a family of sub-optimal algorithms (SOA’s) that try to reduce this complexity while sacrificing little in performance. These algorithms seek to exploit the problem structure revealed by the optimal algorithm. Furthermore, these sub-optimal algorithms all enforce an integer tone allocation during each scheduling interval.

In the optimal algorithm, given the optimal \( \lambda^* \), the optimal subchannel allocation up to any ties is determined by sorting the users on each tone according to the metric \( \mu_{ij}(\lambda) \) as in (14). Given an optimal subchannel allocation, the optimal power allocation is given by a per-user water-filling allocation as in (10). In each SOA, we use the same two phases with some modifications.
to reduce the complexity of computing $\lambda^*$ and the optimal subchannel allocation. Specifically, we begin with a subChannel Allocation (CA) phase in which we assign each sub-subchannel to at most one user. We consider two different SOAs that implement the CA phase differently. In SOA1, instead of using the metric given by the optimal $\lambda$, we consider metrics based on a constant power allocation over all subchannels assigned to a user. In SOA2, we find the subchannel allocation again through a dual based approach, but here we first determine the number of subchannels assigned to each user and then match specific subchannels and users. After the subchannel allocation is done in both SOAs, we perform a Power Allocation (PA) phase in which each user’s power is allocated across the assigned subchannels using a water-filling allocation as in the optimal algorithm.

A. CA in SOA1: Progressive subchannel Allocation based on Metric Sorting

In this family of SOAs, subchannels are assigned sequentially in one pass based on a per user metric for each subchannel, i.e. we iterate $N$ times, where each iteration corresponds to the assignment of one subchannel. Let $K_i(n)$ denote the set of subchannels assigned to user $i$ after the $n$th iteration. Let $g_i(n)$ denote user $i$’s metric during the $n$th iteration and let $l_i(n)$ be the subchannel index that user $i$ would like to be assigned if he is assigned the $n$th subchannel. The resulting CA algorithm is given in Algorithm 1. Note that the user metrics are updated after each subchannel is assigned.

We consider several variations of Algorithm 1 which correspond to different choices for steps 4 and 5. The choices for step 4 are:

(4A): Sort all of the subchannels based on the best normalized SINR among the users, i.e., find a subchannel permutation $\{\alpha_j\}$ such that $\max_i e_{i\alpha_1} \geq \max_i e_{i\alpha_2} \geq \cdots \geq \max_i e_{i\alpha_N}$, and set $l_i(n) = \alpha_n$ for each user $i$. Each $\max$ operation has complexity of $O(M)$, and the sorting operation has a complexity of $O(N \log(N))$. The total complexity is $O(NM + N \log N)$. We note that this is only one-time “pre-processing” that needs to be done before the CA phase starts. During the subchannel allocation iterations, the users just choose the subchannel index from the sorted list.

(4B): For each user $i$, set $l_i(n)$ to be the subchannel with the largest gain among all unassigned subchannels, i.e., $l_i(n) = \arg \max_{j \in N \setminus \bigcup_{n \leq i} K_i(n-1)} e_{ij}$. This requires $M$ sorts (one per user); these also need to be performed only once (since each subchannel assignment does not change a user’s
Algorithm 1 CA Phase for SOA1

1: Initialization: set \( n = 0 \) and \( \mathcal{K}_i(n) = \emptyset \) for each user \( i \).

2: while \( n < N \) do
3: \( n + 1 \).
4: Update subchannel index \( l_i(n) \) for each user \( i \).
5: Update metric \( g_i(n) \) for each user \( i \).
6: Find \( i^*(n) = \arg \max_i g_i(n) \) (break ties arbitrarily).
7: Assign the \( n \)th subchannel to user \( i^*(n) \):

\[
\mathcal{K}_i(n) = \begin{cases} 
\mathcal{K}_i(n-1) \cup \{l_i(n)\}, & \text{if } i = i^*_n; \\
\mathcal{K}_i(n-1), & \text{otherwise}.
\end{cases}
\]

8: end while

ordering of the remaining subchannels) and can be done in parallel. The total complexity of the \( M \) sorting operations is \( O(MN \log N) \), which is higher than that in (4A).

Let \( k_i(n) = |\mathcal{K}_i(n)| \). The choices for Line 5 are:

(5A): Set \( g_i(n) \) to be the total increase in user \( i \)'s utility if assigned subchannel \( l_i(n) \), assuming constant power allocation over all assigned subchannels, i.e.,

\[
g_i(n) = w_i \left[ \sum_{j \in \mathcal{K}_i(n-1) \cup \{l_i(n)\}} \log \left( 1 + \frac{P_i e_{ij}}{k_i(n-1) + 1} \right) - \sum_{j \in \mathcal{K}_i(n-1)} \log \left( 1 + \frac{P_i e_{ij}}{k_i(n-1)} \right) \right].
\] (16)

(5B): Set \( g_i(n) \) to be user \( i \)'s gain from only subchannel \( l_i(n) \), assuming constant power allocation, i.e.

\[
g_i(n) = w_i \log \left( 1 + \frac{P_i}{k_i(n-1) + 1} e_{i,l_i(n)} \right).
\]

Compared with (5A), this metric ignores the change in user \( i \)'s utility due to the decrease in power allocated to any subchannels in \( \mathcal{K}_i(n-1) \).

The complexity of running the \( N \) iterations would be \( O(N) \). The total complexity for the CA phase would be \( O(NM + N \log N) \) (if (4A) is chosen) or \( O(MN \log N) \) (if (4B) is chosen).

B. CA in SOA2: subchannel Number Assignment & subchannel User Matching

SOA2 implements the CA phase through two steps: subchannel number assignment (CNA) and subchannel user matching (CUM). The algorithm is summarized in Algorithm 2.
Algorithm 2 CA Phase of SOA2

1) subChannel Number Assignment (CNA) step: determine the number of subchannels \( n_i \) allocated to each user \( i \) such that \( \sum_{i \in \mathcal{M}} n_i \leq M \).

2) subChannel User Matching (CUM) step: determine the subchannel assignment \( x_{ij} \in \{0, 1\} \) for all users \( i \) and subchannels \( j \), such that \( \sum_{j \in \mathcal{N}} x_{ij} = n_i \).

1) subChannel Number Assignment (CNA): In the CNA step, we determine the number of subchannels \( n_i \) assigned to user \( i \) for each \( i \in \mathcal{M} \). The assignment is calculated based on the approximation that each user sees a wideband flat fading subchannel. Notice that here we do not specify which subchannel is allocated to which user; such mapping will be determined in the CUM step. The CNA step is further divided into three stages: a basic assignment stage, an improvement assignment stage, and an integer approximation stage.

1) Basic Assignment: assignment based on normalized SINR averaged over all subchannels:

In this stage we model each user \( i \) as having a flat fading subchannel with normalized SINR \( \bar{\tau}_i = \frac{1}{N} \sum_{j \in \mathcal{N}} e_{ij} \), and then determine a subchannel number assignment \( n_i \) for all \( i \) by solving:

\[
\max_{\{n_i \geq 0, i \in \mathcal{M}\}} \sum_{i \in \mathcal{M}} w_i n_i \log \left( 1 + \frac{P_i}{n_i \tau_i} \right) \quad \text{(SOA2-CNA)}
\]

subject to: \( \sum_{i \in \mathcal{M}} n_i \leq N \).

We can show that Problem SOA2-CNA is a standard concave maximization problem over a convex set, which can be solved by a dual relaxation method. Consider the Lagrangian

\[
L(n, \lambda) := \sum_{i \in \mathcal{M}} w_i n_i \log \left( 1 + \frac{P_i}{n_i \tau_i} \right) - \lambda \left( \sum_{i \in \mathcal{M}} n_i - N \right).
\]

Optimizing \( L(n, \lambda) \) over \( n = (n_i, i \in \mathcal{M}) \) for a given \( \lambda \) is equivalent to solving the following \( M \) subproblems,

\[
n_i(\lambda) = \arg \max_{n_i} \left( w_i n_i \log \left( 1 + \frac{P_i}{n_i \tau_i} \right) - \lambda n_i \right), \forall i.
\]  

(17)

Problem (17) can be solved by a simple line search over the range of \((0, N]\). Substituting the corresponding results into the Lagrangian yields

\[
L(\lambda) := \sum_{i \in \mathcal{M}} w_i n_i^* (\lambda) \log \left( 1 + \frac{P_i}{n_i^* (\lambda) \tau_i} \right) - \lambda \left( \sum_{i \in \mathcal{M}} n_i^* (\lambda) - N \right),
\]

January 16, 2008
which is a convex function of $\lambda$ [15]. The optimal value

$$\lambda^* = \arg \min_{\lambda \geq 0} L(\lambda)$$  \hspace{1cm} (18)

can be found by a line section search over $[0, \max_i (w_i \log(1 + P_i \bar{\epsilon}_i))]$\(^7\). For a given search precision, the maximum number of iterations needed to solve either (17) or (18) is fixed\(^8\), thus the worst case complexity of the basic assignment step is independent of $M$ or $N$. Since each user is allocated only a subset of the subchannels, the normalized SINR $\bar{\epsilon}_i = \frac{1}{N} \sum_{j \in N} e_{ij}$ is typically a pessimistic estimate of the averaged subchannel conditions over the allocated subset. This motivates us to consider the following assignment improvement stage of CNA.

2) Assignment Improvement: Iterative calculations with normalized SINR averaged over the best subchannel subset. In this stage we iteratively solve the following variation of Problem SOA2-CNA:

$$\max_{\{n_i(t) \geq 0, i \in M\}} \sum_{i \in M} w_i n_i(t) \log \left(1 + \frac{P_i}{n_i(t)} \bar{\epsilon}_i(t)\right)$$  \hspace{1cm} (SOA2-CNA-t)

subject to: $\sum_{i \in M} n_i(t) \leq N$,

for $t = 1, 2, \ldots$. During the $t$-th iteration, $\bar{\epsilon}_i(t)$ is a refined estimate of normalized SINR based on the best $n_i(t - 1)$ subchannels of user $i$. The iteration will stop when the subchannel number allocation converges or the maximum number of iterations is reached.

3) Integer Approximation. Let $n^*$ denote the optimal solution of SOA2-CNA-t after the last iteration. This may not be integer-valued, in which case we then find a feasible integer solution $\hat{n}$ to approximate $n^*$.

The complete algorithm of the CNA phase of SOA2 is given in Algorithm 3, where $\lceil x \rceil$ denotes the smallest integer that is no smaller than $x$ and $\lfloor x \rfloor$ denotes the maximum integer that is no larger than $x$. Lines 1 to 5 correspond to the basic assignment ($t = 1$) and improvement assignment stages ($t > 1$), and the lines 6 to 9 correspond to integer approximation stage. In order to perform the assignment improvement, we need to perform $M$ sorting operations, with

\(^7\)The upperbound of the search interval is obtained by examining the first order optimality condition of (17).

\(^8\)For example, if we use bi-section search to solve (17) and stop when the relative error of the solution is less than $N/2^{10}$, then we only need a maximum of ten search iterations.
a total complexity $O(MN \log(N))$. Note that this only needs to be done once. Each iteration of finding the subchannel allocation has complexity of $O(M)$ due to solving $M$ subproblems for a fixed dual variable. The maximum number of iterations is fixed and thus is independent of $N$ or $M$. The integer approximation stage requires a sorting with the complexity of $O(M \log(M))$. So the total complexity for the CNA phase of SOA2 is $O(MN \log(N))$.

**Algorithm 3 subchannel Number Assignment (CNA) Phase of SOA2**

1) Initialization: integer MaxIteration > 0, $t = 0$, and $n_i(0) = N$ for all user $i$.
2) $t = t + 1$.
3) For each user $i$, calculate $\bar{e}_i(t)$ as the average subchannel condition of the best $\lceil n_i(t - 1) \rceil$ subchannels of that user.
4) Solve Problem SOA2-CNA-t to determine the optimal $n_i(t)$ for all $i$.
5) If $n_i(t) \neq n_i(t - 1)$ for some $i$ and $t < \text{MaxIteration}$, go to line 2. Otherwise, let $n_i^* = n_i(t)$ for all $i$.
6) Order users in descending order of the mantissa of $n_i^*$, $fr(n_i^*) = n_i^* - \lfloor n_i^* \rfloor$. That is, find a user permutation subset $\{\alpha_k, 1 \leq k \leq N\}$ such that 
   \[ fr(n_{\alpha_1}^*) \geq fr(n_{\alpha_2}^*) \geq \cdots \geq fr(n_{\alpha_M}^*) \].
7) For each user $i$, let $n_i^* = \lfloor n_i^* \rfloor$.
8) Calculate the number of unallocated subchannels, $N^A = N - \sum_i n_i^*$.
9) Adjust the users with the large mantissas such that all the subchannels are allocated, i.e., $n_{\alpha_i}^* = n_{\alpha_i}^* + 1$ for all $1 \leq i \leq N^A$.

2) **subChannel User Matching (CUM) Step:** After the CNA step, we know how many subchannels are to be allocated to each user. However, we still need to determine which specific subchannels are assigned to which user. This is accomplished in the CUM step by finding a subchannel assignment that maximizes the weighted-sum rate assuming each user employs a flat
power allocation, i.e. we solve the problem:

$$\max_{x_{ij} \in \{0, 1\}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} x_{ij} w_i \log \left( 1 + \frac{P_i}{n_i}e_{ij} \right)$$

subject to: $$\sum_{j \in \mathcal{N}} x_{ij} = n_i^*, \forall i \in \mathcal{M},$$

$$\sum_{i \in \mathcal{M}} x_{ij} = 1, \forall j \in \mathcal{N},$$

where $\mathbf{n}^* = (n_i^*, i \in \mathcal{M})$ is the integer subchannel allocation obtained in the CNA step.

Problem SOA2-CUM is an integer Assignment Problem whose optimal solution can be found by using the Hungarian Algorithm [13]. To use the Hungarian algorithm in our problem, we need to perform “virtual user splitting” as explained next. For user $i$, let $r_{ij} = w_i \log \left( 1 + \frac{P_i}{n_i}e_{ij} \right)$, and let

$$\mathbf{r}_i = [r_{i1}, r_{i2}, \ldots, r_{iN}]$$

be user $i$’s achievable rates over all possible subchannels. We can then form a $M$ by $N$ matrix $\mathbf{R} = [\mathbf{r}_1^T, \mathbf{r}_2^T, \ldots, \mathbf{r}_M^T]^T$. Next, we split each user $i$ into $n_i^*$ virtual users by adding $n_i^* - 1$ copies of the row vector $\mathbf{r}_i$ to the matrix $\mathbf{R}$. This expands $\mathbf{R}$ into a $N$ by $N$ square matrix. Solving Problem SOA2-CUM is then equivalent to finding a permutation matrix $\mathbf{C}^* = [c_{ij}]_{N \times N}$ such that

$$\mathbf{C}^* = \arg \min_{\mathbf{C} \in \mathcal{C}} -\mathbf{C} \cdot \mathbf{R} = -\sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} r_{ij}. \quad (19)$$

Here $\mathcal{C}$ is the set of permutation matrices, i.e., for any $\mathbf{C} \in \mathcal{C}$, we have $c_{ij} \in \{0, 1\}$, $\sum_i c_{ij} = 1$ and $\sum_j c_{ij} = 1$ for all $i$ and $j$. This problem can be solved by the standard Hungarian algorithm which has a computational complexity of $O(N^3)$, where $N$ is the total number of subchannels. The detailed algorithm can be found in [13]. After obtaining $\mathbf{C}^*$, we can calculate the corresponding subchannel allocation $\mathbf{x}^*$. For example, if $c_{kj}^* = 1$ and virtual user $k$ corresponds to the actual user $i$, then we know $x_{ij}^* = 1$, i.e., subchannel $j$ is allocated to user $i$ only.

A similar idea has been used to solve various downlink OFDM resource allocation problems (e.g., [17]) as well as to find user coalitions for Nash Bargaining in an uplink OFDM system [16].
C. Power Allocation (PA) phase with Single-user Water-filling

After the subchannel allocation (CA) phase in either SOA1 and SOA2, each user still needs to allocate its transmission power across all subchannels that are allocated to it. For user $i$, this can be formulated as the following problem

$$
\max_{\mathbf{p}_i \in P_i} \ x_i^* \log \left( 1 + p_{ij} e_{ij} \right),
$$

where

$$
P_i = \left\{ \mathbf{p}_i \geq 0 : p_{ij} \leq \frac{s_{ij}}{e_{ij}}, \sum_{j \in N} p_{ij} \leq P_i \right\}.
$$

This can be done by using a water-filling type of power allocation, taking into account of the per subchannel power constraint. Specifically, if $\sum_{j \in N} x_{ij}^* s_{ij} e_{ij} \leq P_i$, we let $p_{ij}^* = \frac{s_{ij}}{e_{ij}}$. Otherwise, the optimal power allocation is determined by

$$
p_{ij}^* = \left( \frac{x_{ij}^*}{\nu_i} - 1 \right)^+ \land \frac{s_{ij}}{e_{ij}},
$$

where the constant $\nu_i$ is chosen so that $\sum_{j \in N} p_{ij}^* = P_i$. Note that It is possible that some subchannel is allocated to user $i$ but gets zero transmission power due to its poor subchannel gain. The optimal value of $\nu_i$ can be found through a simple line search, with a constant worst case complexity given a fixed search precision similar as solving (17). Note that this typically leads to an approximate value of the $v_i$, although in practice it will be good enough if the search precision is set to be small enough.

Alternatively, we can exploit the structure of our problem and give an alternative characterization for the exact value $v_i$ as in the following lemma.

Lemma 1: A $v_i$ solves the PA problem of user $i$ if and only if

$$
v_i = \frac{\sum_j x_{ij}^* 1 \left\{ \frac{s_{ij}}{e_{ij}} \leq v_i < x_{ij}^* e_{ij} \right\}}{P_i - \sum_j \frac{s_{ij}}{e_{ij}} 1 \left\{ v_i < \frac{x_{ij}^* s_{ij}}{\nu_i + s_{ij}} \right\} + \sum_j \frac{1}{e_{ij}} 1 \left\{ \frac{x_{ij}^* s_{ij}}{\nu_i + s_{ij}} \leq v_i < x_{ij}^* e_{ij} \right\}},
$$

where $1\{\}$ is the indicator function.

By using equation (20), we can derive an algorithm that finds the exact value of $v_i$ in a maximum of $2n_i^*$ steps. Details of the algorithm can be found in our previous work [4], where we used the same algorithm to address the power allocation problem in the downlink. The key difference here is that we need to run the algorithm up to $M$ times, once for each user which is allocated a subchannel. Since $\sum n_i^* = N$, it follows that the total worst case computational complexity for this approach is $O(MN)$. 

January 16, 2008
TABLE II
Worst Case Computational Complexity of Suboptimal Algorithms

<table>
<thead>
<tr>
<th>Suboptimal Algorithm</th>
<th>Worst Case Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>subChannel Allocation (CA)</td>
<td></td>
</tr>
<tr>
<td>4A &amp; 5A</td>
<td>$O(NM + N \log N)$</td>
</tr>
<tr>
<td>4A &amp; 5B</td>
<td>$O(NM + N \log N)$</td>
</tr>
<tr>
<td>4B &amp; 5A</td>
<td>$O(MN \log N)$</td>
</tr>
<tr>
<td>4B &amp; 5B</td>
<td>$O(MN \log N)$</td>
</tr>
<tr>
<td>Power Allocation (PA)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$O(MN)$</td>
</tr>
<tr>
<td>Total (CA + PA)</td>
<td>$O(MN \log N)$</td>
</tr>
<tr>
<td>subChannel Allocation (CA)</td>
<td></td>
</tr>
<tr>
<td>CNA</td>
<td>$O(MN \log N)$</td>
</tr>
<tr>
<td>CUM</td>
<td>$O(N^3)$</td>
</tr>
<tr>
<td>Power Allocation (PA)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$O(MN)$</td>
</tr>
<tr>
<td>Total (CA+PA)</td>
<td>$O(N^3 + MN \log N)$</td>
</tr>
</tbody>
</table>

D. Complexity Summary of Suboptimal Algorithms

The worst case computational complexities of SOA1 and SOA2 are summarized in Table II. Notice that if we further make the assumption that $N > M$ (i.e., number of subchannels is larger than the number of users, which is almost always satisfied in practical OFDM systems), then the total complexity of SOA2 is $O(N^3)$. Notice that both algorithms have polynomial time complexities, which is much better than the tie-breaking integer allocation algorithm proposed in Section IV. In that case, the tie breaking procedure has a worst case exponential complexity of $O(M^N)$.

VI. Simulation Results

We report simulation results for the following four algorithms:

1) Integer-Dual: integer subchannel allocation (with tie breaking) based on optimal algorithm (as in Section IV) and power control as in Section V-C. Since in general many ties exist after solving Problem UL, it is too complicated to compare all possible ties and to break them with the best one. To limit the complexity when ties occur, we inspect up to 128 ways of breaking the ties with an integer allocation and select the allocation among these with the largest weighted sum rate (before reallocating the power).
2) SOA1: subchannel allocation as in Section V-A and power control as in Section V-C. There are four versions of SOA1, depending on how steps 4 and 5 in Algorithm 1 are implemented; we present results for each.

3) SOA2: subchannel allocation as in Section V-B (with upto 10 iterations) and power control as in Section V-C.

4) Base-line: each subchannel $j$ is allocated to the user $i$ with the highest $e_{ij}$, without considering the weights $w_i$’s and the power constraints. Each user’s power is then allocated as in Section V-C.

All results are for a single OFDM cell with 40 users. Each user’s subchannel gains are the product of a constant location-based term, picked using an empirically obtained distribution, and a fast fading term, generated using a block-fading model and a standard mobile delay-spread model with a delay spread of $10\,\mu$sec. The fast-fading component for each multi-path component is held fixed for 2msec and an independent value is generated for the next block, which corresponds to a 250MHz Doppler. The system bandwidth is 5MHz corresponding to 512 OFDM tones. Resource allocation is performed using adjacent groups of 8 tones.\footnote{This corresponds to the “Band AMC mode” of 802.16 d/e.} The symbol duration is $100\,\mu$sec with a cyclic prefix of $10\,\mu$sec. All users are infinitely-backlogged with the same utility function of $U_i(W_{i,t}) = (W_{i,t})^\alpha/\alpha$, where $W_{i,t}$ is the long term average throughput of user $i$. All users have the same maximum power constraint of $P_i = 2W$. Each simulation run is for 1000 fading blocks.

In all tables, the “Utility” column denotes $\sum_i U_i(W_{i,t})$ where $t$ is the end of simulation time. The “Log U” column denotes the logarithmic utility function, $\sum_i \log(U_i(W_{i,t}))$, which provides a characterization of fairness among users. The “User Scheduled” column denotes the average number of users who receive positive rates within one scheduling interval. The “Optimality Ratio” for the Integer-Dual algorithm denotes the ratio between the objective value of Problem UL-Integer achieved with the algorithm and the maximum objective value of Problem UL (i.e., by looking at the optimal dual value) averaged over all scheduling instances. This gives us an idea of how good Integer-Dual algorithm performs. Note that the optimality ratio is a conservative estimation of the algorithm performance, since the optimal objective value of the relaxed Problem UL typically is strictly larger than the optimal objective function of Problem UL-Integer (i.e., the...
TABLE III
ALGORITHM PERFORMANCE FOR SCHEDULING EVERY 20 OFDM SYMBOLS, α = 0.5

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Utility</th>
<th>Log U</th>
<th>Total Rate (Mbps)</th>
<th>User Scheduled</th>
<th>Optimality Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer-Dual</td>
<td>53922</td>
<td>514.0</td>
<td>21.56</td>
<td>37.5</td>
<td>0.9412</td>
</tr>
<tr>
<td>SOA 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4A &amp; 5A</td>
<td>52494</td>
<td>510.7</td>
<td>22.86</td>
<td>34.6</td>
<td>N/A</td>
</tr>
<tr>
<td>4A &amp; 5B</td>
<td>51697</td>
<td>509.2</td>
<td>20.22</td>
<td>28.1</td>
<td>N/A</td>
</tr>
<tr>
<td>4B &amp; 5A</td>
<td>54165</td>
<td>513.3</td>
<td>22.25</td>
<td>35.0</td>
<td>N/A</td>
</tr>
<tr>
<td>4B &amp; 5B</td>
<td>53156</td>
<td>511.4</td>
<td>21.43</td>
<td>28.6</td>
<td>N/A</td>
</tr>
<tr>
<td>SOA 2</td>
<td>54316</td>
<td>513.6</td>
<td>22.33</td>
<td>35.1</td>
<td>N/A</td>
</tr>
<tr>
<td>Base Line</td>
<td>21406</td>
<td>-1960.5</td>
<td>16.13</td>
<td>2.66</td>
<td>N/A</td>
</tr>
</tbody>
</table>

ratio might never reach 1 even if we solve Problem UL-Integer optimally. Furthermore, we emphasize that this value is based on the average performance of solving problem UL in each scheduling instance; this does not translate directly into a bound on the average utility under an optimal scheduling rule and under the given policy. Since other suboptimal algorithms do not find the optimal dual value of Problem UL at each scheduling instance, the "Optimality Ratio" does not apply to them.

Table III gives the results of the algorithms (summed over all users) when scheduling decisions are made every 20 OFDM symbols (i.e., a fading block of 2msec). The utility parameter α = 0.5. In Table III, SOA1 (with 4B & 5A) and SOA2 achieve the best performances in terms of total utility. Their performance is even better than the Integer-Dual approach, which was obtained based on the optimal value of the relaxed problem. This is likely because only 128 ways to break ties are considered (this is typically not sufficient). Since the Integer-Dual algorithm achieves an optimality ratio of 0.9412, this suggests that SOA1 and SOA2 achieve very close to optimal performance as well. The base-line algorithm always has poor performance.

Table IV shows the performance of each algorithm when scheduling is performed every 80 OFDM symbols, with all other parameters the same as in Table III. It is clear that this coarser allocation leads to poorer performance compared with Table III. This shows the trade-off between system performance and resource allocation frequency (and thus algorithm complexity).

\[11\] In particular, the trajectory of the scheduling “weights” will be different under the two policies.
TABLE IV
ALGORITHM PERFORMANCE WITH SCHEDULING EVERY 80 OFDM SYMBOLS, $\alpha = 0.5$

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Utility</th>
<th>Log U</th>
<th>Total Rate (Mbps)</th>
<th>User Scheduled</th>
<th>Optimality Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer-Dual</td>
<td>47843</td>
<td>503.58</td>
<td>17.26</td>
<td>37.1</td>
<td>0.9318</td>
</tr>
<tr>
<td>SOA 1 4A &amp; 5A</td>
<td>46851</td>
<td>500.4</td>
<td>16.98</td>
<td>34.5</td>
<td>N/A</td>
</tr>
<tr>
<td>4A &amp; 5B</td>
<td>42901</td>
<td>492.5</td>
<td>14.38</td>
<td>25.3</td>
<td>N/A</td>
</tr>
<tr>
<td>4B &amp; 5A</td>
<td>47716</td>
<td>502.0</td>
<td>17.62</td>
<td>35.1</td>
<td>N/A</td>
</tr>
<tr>
<td>4B &amp; 5B</td>
<td>43999</td>
<td>494.6</td>
<td>15.14</td>
<td>26.3</td>
<td>N/A</td>
</tr>
<tr>
<td>SOA 2</td>
<td>47682</td>
<td>502.1</td>
<td>17.55</td>
<td>35.1</td>
<td>N/A</td>
</tr>
<tr>
<td>Base Line</td>
<td>15887</td>
<td>-2116.5</td>
<td>11.65</td>
<td>2.64</td>
<td>N/A</td>
</tr>
</tbody>
</table>

TABLE V
ALGORITHM PERFORMANCE FOR SCHEDULING EVERY 20 OFDM SYMBOLS, $\alpha = 0$

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Utility</th>
<th>Log U</th>
<th>Total Rate (Mbps)</th>
<th>User Scheduled</th>
<th>Optimality Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer-Dual</td>
<td>515</td>
<td>515</td>
<td>19.80</td>
<td>39.0</td>
<td>0.9715</td>
</tr>
<tr>
<td>SOA 1 4A &amp; 5A</td>
<td>511</td>
<td>511</td>
<td>18.47</td>
<td>36.9</td>
<td>N/A</td>
</tr>
<tr>
<td>4A &amp; 5B</td>
<td>509</td>
<td>509</td>
<td>21.56</td>
<td>28.6</td>
<td>N/A</td>
</tr>
<tr>
<td>4B &amp; 5A</td>
<td>515</td>
<td>515</td>
<td>20.04</td>
<td>37.4</td>
<td>N/A</td>
</tr>
<tr>
<td>4B &amp; 5B</td>
<td>512</td>
<td>512</td>
<td>17.80</td>
<td>29.3</td>
<td>N/A</td>
</tr>
<tr>
<td>SOA 2</td>
<td>515</td>
<td>55</td>
<td>19.91</td>
<td>37.4</td>
<td>N/A</td>
</tr>
<tr>
<td>Base Line</td>
<td>-1961</td>
<td>-1961</td>
<td>16.13</td>
<td>2.66</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Tables V and VI give the results of the algorithms when the utility parameter $\alpha$ is equal to 0 (proportional fair allocation) and 1 (maximum rate allocation), respectively. In each scheduling decisions are made every 20 OFDM symbols, It is clear that there is a trade-off between fairness (measured by Log U) and efficiency (measured by total rate). $\alpha = 0$ is the most fair and the least efficient, while $\alpha = 1$ is the most efficient and least fair. Once again we note that all of the heuristics have good performance with SOA1 (with 4B & 5A) and SOA2 achieving the best performances in terms of total utility.

In each case simulated all of the SOA’s have good performance with SOA1 (with 4A & 5B) and SOA2 consistently achieving the best performance in terms of total utility. From the
TABLE VI
ALGORITHM PERFORMANCE FOR SCHEDULING EVERY 20 OFDM SYMBOLS, \( \alpha = 1 \)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Utility</th>
<th>Log U</th>
<th>Total Rate (Mbps)</th>
<th>User Scheduled</th>
<th>Optimality Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer-Dual</td>
<td>23243349</td>
<td>472.56</td>
<td>23.37</td>
<td>20.43</td>
<td>0.82541</td>
</tr>
<tr>
<td>4A &amp; 5A</td>
<td>23186687</td>
<td>448.99</td>
<td>22.28</td>
<td>22.6</td>
<td>N/A</td>
</tr>
<tr>
<td>4A &amp; 5B</td>
<td>23109607</td>
<td>-136.0251</td>
<td>23.20</td>
<td>15.6</td>
<td>N/A</td>
</tr>
<tr>
<td>4B &amp; 5A</td>
<td>24318459</td>
<td>-444.02</td>
<td>24.42</td>
<td>32.7</td>
<td>N/A</td>
</tr>
<tr>
<td>4B &amp; 5B</td>
<td>23953809</td>
<td>-195.10</td>
<td>24.05</td>
<td>15.6</td>
<td>N/A</td>
</tr>
<tr>
<td>SOA 1</td>
<td>24464874</td>
<td>372.95</td>
<td>24.57</td>
<td>21.8</td>
<td>N/A</td>
</tr>
<tr>
<td>SOA 2</td>
<td>16077515</td>
<td>-1961</td>
<td>16.13</td>
<td>2.66</td>
<td>N/A</td>
</tr>
</tbody>
</table>

In Section V-D, we note that these have slightly higher complexity than some of the other SOA's. Hence if lower complexity is needed, this can be achieved with only a slight loss in performance. We also note that in each case the SOAs and the integer-dual algorithm schedule a large number of users on average in each time-slot. A potential cost from this is that it may increase the needed signaling overhead. One way to reduce this cost is to add a penalty term to our objective which increases with the number of users scheduled.

VII. CONCLUSIONS

We presented an optimization-based formulation for scheduling and resource allocation in the uplink of an OFDM access network. Compared to the downlink, we argued that the uplink was computationally more challenging due in part to the per-user power constraints. A (high complexity) optimal algorithm was given as well as a family of low complexity heuristics. The heuristics were shown to have good performance via simulations for a range of different user utilities and scheduling time-scales. Two algorithms from this family consistently achieved the best performance, but had a slightly higher complexity than some of the other algorithms, enabling complexity to be traded off with performance within this family of algorithms.
APPENDIX

A. Subgradient Algorithm for the Optimal Algorithm with Fractional Allocation (Sect. III)

Given a vector of dual variables $\lambda$, the dual value is

$$L(\lambda) = \sum_j \max_i w_i h(\lambda_i, w_i e_{ij}, s_{ij}) + \sum_i \lambda_i P_i.$$ 

We will use the subgradient search method as in exercise 6.3.2 of [15] to search for the optimal dual variables $\lambda^* = \arg\min_{\lambda} L(\lambda)$. The $\lambda$ is updated as

$$\lambda_i(t + 1) = \left[\lambda_i(t) - \kappa(t) \left(P_i - \sum_j p_{ij}^* (\lambda(t))\right)\right]^+,$$

where $\kappa(t)$ is chosen as

$$\kappa(t) = \frac{L(\lambda(t)) - \tilde{L}}{\sum_i \left(P_i - \sum_j p_{ij}^* (\lambda(t))\right)^2}.$$

Here $\tilde{L}$ is an adjustable estimate of the optimal dual value (i.e., $\min_{\lambda} L(\lambda)$). To be specific, define two positive scalars $\delta(1)$ and $V$. The Algorithm operates in cycles during which $\tilde{L}$ is kept constant. Cycle 1 begins with the starting point $\lambda(1)$. At the beginning of the typical cycle $t$, we have a vector $\lambda(t)$ and we set $\tilde{L} = L(\lambda(t)) - \delta(t)$. Cycle $t$ consists of successive subgradient iterations that start with $\lambda^1(t) = \lambda(t)$ and continue with $\lambda^k(t)$ (for $k = 1, 2, ...$) until one of the following two events occurs:

1) The dual value falls below $L(\lambda(t)) - \delta(t)/2$.

2) The length of the path traveled starting from $\lambda(t)$, given by

$$\sum_k \kappa(t) \left(\sum_i \left(P_i - \sum_j p_{ij}^* (\lambda^k(t))\right)^2\right)^{1/2} = \sum_k \left(\frac{L(\lambda^k(t)) - \tilde{L}}{\sum_i \left(P_i - \sum_j p_{ij}^* (\lambda^k(t))\right)^2}\right)^{1/2},$$

exceeds $V$.

Cycle $t + 1$ begins with $\lambda(t + 1)$ equal to the vector that has the minimum dual value within cycle $t$, i.e., $\lambda(t + 1) = \arg\min_{\lambda^k(t)} L(\lambda^k(t))$, and either $\delta(t + 1) = \delta(t)$ or $\delta(t + 1) = \delta(t)/2$, depending on whether cycle $t$ terminated with case (1) or (2), respectively. The complete algorithm is given in Algorithm 4.
Algorithm 4 Subgradient Search of Optimal $\lambda$ in the Optimal Algorithm

1) Initialization: choose $\varepsilon > 0$, $V > 0$, $\delta (1) > 0$, $\lambda (1) \geq 0$ and $t = 0$.
2) Let $t = t + 1$, $\bar{L} = L (\lambda (t)) - \delta (t)$, $k = 1$ and $\lambda^1 (t) = \lambda (t)$.
3) Update the dual variables
   \[
   \lambda_i^{k+1} (t) = \left[ \lambda_i^k (t) - \kappa^k (t) \left( P_i - \sum_j p_{ij}^* (\lambda^k (t)) \right) \right]^+, \forall i,
   \]
   where
   \[
   \kappa^k (t) = \frac{L (\lambda^k (t)) - \bar{L}}{\sum_i \left( P_i - \sum_j p_{ij}^* (\lambda^k (t)) \right)^2}.
   \]
   If $\max_i |\lambda_i^{k+1} (t) / \lambda_i^k (t) - 1| \leq \varepsilon$, stop. Otherwise, let $k = k + 1$.
4) Calculate the optimal power allocation ($p_{ij}^* (\lambda^k (t)), \forall i, j$) and the dual value $L (\lambda^k (t))$.
   - If $L (\lambda^k (t)) < L (\lambda (t)) - \delta (t) / 2$, go to step (5).
   - If $\sum_k \kappa^k (t) \left( \sum_i \left( P_i - \sum_j p_{ij}^* (\lambda^k (t)) \right)^2 \right)^{1/2} > V$, go to step (6).
   Otherwise, go to step (3).
5) Let $\delta (t + 1) = \delta (t)$ and $\lambda (t + 1) = \arg \min_k L (\lambda^k (t))$. Go to step (2).
6) Let $\delta (t + 1) = \delta (t) / 2$ and $\lambda (t + 1) = \arg \min_k L (\lambda^k (t))$. Go to step (2).

REFERENCES


