Proportional Fair Frequency-Domain Packet Scheduling for 3GPP LTE Uplink

Suk-Bok Lee* Ioannis Pefkianakis* Adam Meyerson* Shugong Xu† Songwu Lu*

*Computer Science Department
UCLA, CA 90095

†Sharp Laboratories of America
Camas, WA 98607

***Abstract*—With the power consumption issue of mobile handset taken into account, Single-carrier FDMA (SC-FDMA) has been selected for 3GPP Long-Term Evolution (LTE) uplink multiple access scheme. Like in OFDMA downlink, it enables multiple users to be served simultaneously in uplink as well. However, its single carrier property requires that all the subcarriers allocated to a single user must be *contiguous* in frequency within each time slot. This contiguous allocation constraint limits the scheduling flexibility, and frequency-domain packet scheduling algorithms in such system need to incorporate this constraint while trying to maximize their own scheduling objectives.**

In this paper we explore this fundamental problem of LTE SC-FDMA uplink scheduling by adopting the conventional time-domain *Proportional Fair* algorithm to maximize its objective (i.e. proportional fair criteria) in the frequency-domain setting. We show the NP-hardness of the frequency-domain scheduling problem under this contiguous allocation constraint and present a set of practical algorithms fine tuned to this problem. We demonstrate that competitive performance can be achieved in terms of system throughput as well as fairness perspective, which is evaluated using 3GPP LTE system model simulations.

I. Introduction

In recent years Orthogonal Frequency Division Multiple Access (OFDMA) has been considered as a strong candidate for the broadband air interface for its robustness to multipath fading, higher spectral efficiency and bandwidth scalability, and it has been selected for 3GPP Long-Term Evolution (LTE) downlink (DL) radio access technology. However, one major disadvantage of OFDMA is that the instantaneous transmitted RF power can vary dramatically within a single OFDM symbol. Such an undesirable high peak-to-average power ratio (PAPR) is a serious concern for the uplink (UL), since power consumption is a key consideration for the mobile handsets. As a result of seeking an alternative to OFDMA, Single-carrier FDMA (SC-FDMA) has been selected for LTE uplink multiple access scheme. While keeping most of the advantages of OFDMA (e.g. the same degree of multipath protection), SC-FDMA has significantly lower PAPR, since the underlying waveform is essentially single-carrier. Thus, lower PAPR of SC-FDMA greatly benefits the mobile terminal in terms of transmit power efficiency.

As in DL OFDMA, multiple access in UL SC-FDMA is achieved by assigning different frequency portions of the system bandwidth to individual users based on their channel conditions. Such simultaneous frequency-domain multiplexing of users (inherently in concert with time-domain scheduling) is performed by *frequency domain packet scheduling* (FDPS). In LTE UL, the system bandwidth is divided into multiple subbands (i.e. groups of subcarriers) denoted as *physical resource blocks* (RBs). In order to achieve large gain from multiuser frequency diversity, a scheduler needs to know the instantaneous radio channel conditions across all users and all RBs, which are fed as input for the frequency-domain adaptive user-to-RB allocation. For example, in LTE UL each user transmits a Sounding Reference Signal (SRS) to the scheduling node (i.e. base station) [1], which is used as *channel quality indicator* (CQI). With CQIs across all users and all RBs, a base station performs RB-to-user assignment at each time slot (e.g. in LTE every 1ms) according to the selected scheduling policy. Thus, in the time-frequency domain, an RB is considered as a minimum scheduling resolution, and also a minimum unit of the data-rate adaptation by *adaptive modulation and coding* (AMC) with a granularity of one sub-frame.

Most of the DL FDPS algorithms proposed so far adopt the well-known time-domain *Proportional Fair* (PF) algorithm as a basic scheduling principle and apply the PF algorithm directly over each RB one-by-one independently. However, such scheduling strategies cannot be employed in the UL SC-FDMA. Due to its single carrier property, SC-FDMA requires that all the RBs allocated to a single user must be *contiguous* in frequency within each time slot (i.e. sub-frame) [5], [6]. Thus, LTE UL FDPS algorithms should respect this constraint while trying to maximize their own scheduling objectives.

In this paper we study this fundamental problem of UL frequency-domain packet scheduling under contiguous RB allocation constraint. We analyze this problem by adopting the widely employed PF algorithm to maximize its objective (i.e. proportional fair criteria) in the frequency-domain setting. The main goal of this paper is to investigate how to adapt the time-domain PF algorithm to this problem framework.

A. The Model

We consider a cellular network whose UL system bandwidth is divided into m RBs, and we have a single base station and n active wireless users. The base station can allocate m RBs to a set of n users. At each time slot multiple RBs (with the contiguity constraint) can be assigned to a single user, each RB however can be assigned to at most one user. In this paper we shall work in an *infinitely backlogged* model in which for each user there is always data available for service. Thus, the base station can schedule all the m RBs every time slot.

We define the indicator variable $x_i^c(t)$ to indicate whether or not RB c is assigned to user i at time slot t. We assume that channel conditions vary across RBs as well as users. The channel conditions typically depends on the channel frequency, so they may be different for different channels;

moreover, they also depends on the user location and the time slot. Therefore, each RB has *user-dependent* and *time-varying* channel condition. We use $r_i^c(t)$ to denote the instantaneous channel rate for user i on RB c at time t. This channel rates are estimated from the CQIs extracted from the UL channel sounding. Thus, if $x_i^c(t) = 1$, then user i can transmit data of size $r_i^c(t)$ on RB c at time slot t.

B. Problem Formulation

In the time-domain context, the well known Proportional Fair (PF) algorithm aims to maximize, over all feasible scheduling rules, the utility function $\sum_i \log R_i$, where R_i is the long-term service rate of user i. This objective is known as *proportional fair criteria*. Maximizing $\sum_i \log R_i$ not only improves overall throughput but also prevents any user from being completely starved since $\log 0 = -\infty$. In order to maximize $\sum_i \log R_i$, we should serve the user who maximizes $r_i(t)/R_i(t)$ at each time slot t (proven in [7], [17], [22]). Note that the PF algorithm achieves high throughput and maintains proportional fairness among all users by giving priority to users with a high-quality channel rate ($r_i(t)$) and a low current average service rate ($R_i(t)$).

We now adapt this time-domain PF metric to the frequency-domain setting with the utility function $\sum_i \log R_i$ as our objective. Let $\lambda_i^c(t) = r_i^c(t)/R_i(t)$ be the *PF metric value* that user i has on RB c at time slot t. As justified in [10], we can establish a FDPS version of PF objective function when scheduling time slot t as follows:

$$\max \sum_i \sum_c x_i^c(t) \lambda_i^c(t) \qquad (1)$$

Objective (1) above is indeed analogous to the PF algorithm which maximizes $\sum_i x_i(t) \cdot r_i(t)/R_i(t)$ in the time-domain setting. Hence, optimizing the objective (1) makes the utility function $\sum_i \log R_i$ maximized in the frequency-domain setting. For this reason, most of the proposed DL FDPS scheduling algorithms apply the PF algorithm directly over each RB one-by-one, i.e. for RB c the PF algorithm selects the best user who maximizes $r_i^c(t)/R_i(t)$ at time slot t. However, for LTE UL we need to add the contiguous RB constraint into this objective (1) due to the physical layer requirement of SC-FDMA. Accordingly, we can rewrite the objective (1) more precisely as the following optimization problem:

$$\max \sum_i \sum_c x_i^c \lambda_i^c \qquad (1)$$

$$\text{subject to} \sum_i x_i^c \le 1, \quad \forall c \qquad (2)$$

$$\sum_i \sum_c x_i^c \le m \qquad (3)$$

$$\sum_{c=a}^{b} x_i^c = b - a + 1, \quad \forall i, x_i^a = x_i^b = 1 \qquad (4)$$

$$x_i^c \in \{0, 1\} \qquad (5)$$

To simplify notation, the dependence on time t is omitted. Constraint (2) states that each RB can be assigned to at most one user, and constraint (3) just tells that the system has the total of m RBs. The only added is constraint (4),

w/o contiguous requirement — Max = 85

user \ carrier											
A	**8**	7	6	5	4	3	4	5	6	7	**8**
B	1	8	1	**8**	2	**8**	3	**8**	2	7	1
C	6	6	**6**	5	5	6	4	4	6	6	5
D	3	4	5	6	**7**	8	**9**	8	**7**	6	5
E	7	**8**	6	3	6	4	5	8	2	**8**	6

w/ contiguous requirement — Max = 83

user \ carrier											
A	8	7	6	5	4	3	4	5	6	**7**	**8**
B	1	8	1	**8**	2	8	3	8	2	7	1
C	6	6	6	5	5	6	4	4	6	6	5
D	3	4	5	6	**7**	**8**	**9**	**8**	**7**	6	5
E	**7**	**8**	**6**	3	6	4	5	8	2	8	6

Fig. 1 Maximizing the PF objective. The numbers denote the PF metric values λ_i^c. Dark-colored RBs represent assignment strategies maximizing the objective with/without the contiguity constraint.

which enforces the contiguous RB allocation. Now we need to optimize the objective (1) with keeping to those constraints (i.e. choose the value $x_i^c(t)$ to maximize the PF objective (1)). One crucial difference is that we now cannot apply the PF algorithm on each RB one-by-one in isolation. In other words, the isolated local optimization of each RB hardly optimizes the objective (1). Figure 1 exemplifies the case. With the contiguity constraint we may need to serve users with suboptimal PF metric value λ_i^c for some RBs so as to optimize the PF objective (1).

Seeking to maximize the PF objective (1) under this contiguity constraint, we present five variations of PF-FDPS algorithm ($Alg1$ through $Alg5$). In this paper we explore the fundamental nature of this scheduling problem by investigating how well each of these five algorithms fits into the problem framework.

C. Related Work

The Proportional Fair (PF) algorithm was introduced by [15], [22], extensively studied in the research community (e.g. delay [9], [18], instability [7], [8]), and it is widely used as a standard scheduling algorithm in the current single-carrier wireless systems such as CDMA 2000 1xEV-DO [11], [15].

The area of FDPS scheduling is new, and most of studies directly adapt the time-domain PF algorithm into frequency-domain context. Their results show the potential gains of up to 40-60% average system capacity improvement over time-domain only scheduling [19], and moreover [24] shows that the frequency selectivity of FDPS indeed helps significantly improve the short-term fairness. Andrews et al. [10] have proposed the FDPS-version of *MaxWeight* algorithm[1], and addressed the resource wastage problem induced by small-queue condition in DL FDPS context. The objective of the MaxWeight algorithm is the system stability, and the authors have presented the performance from the queue perspective.

Cohen et al. [13] recently studied the DL OFDMA scheduling problem somewhat related to this contiguous allocation requirement in WiMAX. They present several heuristic algorithms for constructing the OFDMA frame matrix as a collection of rectangles which fit into a single matrix. The algorithms, however, assume that 1) at each time slot the base station somehow knows the scheduled data size for each user in advance; 2) the same channel rate is across all RBs as well as all users. In the WLAN context, Yuan et al. [25] have considered a contiguous channel assignment problem to

[1] *MaxWeight* algorithm always serves the user that maximizes $Q_i^s(t) r(i,t)$, where $Q_i^s(t)$ and $r(i,t)$ are the queue size and the instantaneous data rate of user i, respectively.

dynamically allocate the variable-width channel to each access point (AP). The key difference from our problem is that no channel diversity (i.e. they assume the achievable data rate is linear to the available bandwidth) is considered in their WLAN context. That is, an AP with the fixed bandwidth will attain the same throughput regardless of its central frequency assigned, which makes their problem as a special case of ours.

In summary the contiguous RB allocation constraint is a crucial requirement for the LTE UL scheduling algorithms, yet no previous work has been devoted to this fundamental issue of SC-FDMA.

II. Hardness Result

In this section we first show that unfortunately we cannot hope for an efficient algorithm that optimizes the objective (1) under the contiguous RB restriction unless P = NP. We then demonstrate that it is still computationally intractable in the practical systems.

A. Hardness of objective (1)

Theorem 1: LTE UL PF-FDPS problem (i.e. maximization of the PF objective (1) under the contiguous RB allocation constraint) is NP-hard.

Proof: We use a reduction from Hamiltonian Path Problem. Given a directed graph $G = (V, E)$, we say that a path P in G is a *hamiltonian path* if it contains each vertex in V exactly once. The problem asks whether a directed graph G contains a hamiltonian path, and this is NP-complete [16]. As a pre-processing for our reduction, we can transform any given directed graph G into a bipartite graph G', by splitting each node v in G into two nodes v_l and v_r (say, left and right) in G'; All the incoming/outgoing edges to/from v are attached to v_l and v_r, respectively, with adding an edge from v_l to v_r. (See Figure 2.) It is clear that G' contains a hamiltonian path if and only if G contains a hamiltonian path.

We now show that this hamiltonian path problem in bipartite graph (HAM-PATH-BG) is reducible to our problem. A decision version of our problem is to determine whether for a given frequency-domain status S (i.e. a collection of value λ_i^c across all users and all RBs), there exists a contiguous allocation strategy with resulting aggregate value at least k.

Consider an arbitrary instance of HAM-PATH-BG, with $2n$ nodes (n left nodes $v_{l,1}, \ldots, v_{l,n} \in V_l'$ and n right nodes $v_{r,1}, \ldots, v_{r,n} \in V_r'$). We construct our frequency-domain status instance S as follows. A user in S corresponds to each node in G'. For each left node $v_{l,i}$ and right node $v_{r,i}$, we have user $u_{l,i} \in U_l$ and $u_{r,i} \in U_r$, respectively. Thus, we have $|U_l| + |U_r| = n + n = 2n$ users. We partition the RBs into three classes C_l, C_t, and C_r (i.e. left, transit, right). We take a quantity T to be somewhat sufficiently larger than n; say, $T = n^2$. We arrange the RBs such that T contiguous RBs of C_l and C_r alternate with each other via $n+2$ contiguous RBs of C_t. Such a pattern (i.e. $C_l \to C_t \to C_r$) repeats for n times in the frequency-domain, so we have $T \times 2n + (n+2)(2n-1)$ RBs. (See Figure 3.)

We first assign the scheduling metric value λ_i^c for RBs $\in C_l \cup C_r$ such that the intermediate construction has $n!$ different contiguous allocation strategies that correspond naturally to the $n!$ possible hamiltonian paths (in the case of a complete

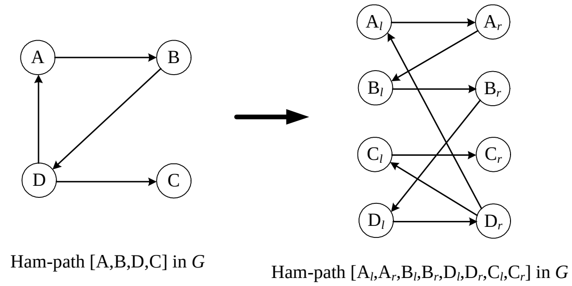


Fig. 2 Equivalence between hamiltonian paths in a given directed graph G and its corresponding bipartite graph G'

graph G). For each user $i \in U_l$ for RB c, we set the value $\lambda_i^c = 1$ if $c \in C_l$, and $\lambda_i^c = 0$ if $c \in C_r$. Similarly, for each user $i \in U_r$ for RB c, we set $\lambda_i^c = 1$ if $c \in C_r$, and $\lambda_i^c = 0$ if $c \in C_l$. (See Figure 3.) At this point, it seems clearly beneficial to allocate RBs $\in C_l$ to users $\in U_l$, and assign RBs $\in C_r$ to users $\in U_r$. It implies that, in order to get as high aggregate value as possible, 1) a user $\in U_l$ and a user $\in U_r$ need to be assigned alternately in the frequency-domain due to the alternate RB placement of C_l and C_r in our construction; 2) every user must be served in the end, since our contiguous allocation constraint prevents once-assigned users from being re-assigned discontiguous RBs.

Now we set the values for RBs $\in C_t$ to model the constraint imposed by the directed edges in G'. Each chunk of RBs $\in C_t$ consists of $n+2$ contiguous RBs, and we denote those RBs as $C_{t(0,l\to r)}, C_{t(1,l\to r)}, \ldots, C_{t(n+1,l\to r)}$ in sequence if the chunk is for transition from C_l to C_r (in opposite, we denote as $C_{t(0,r\to l)}, \ldots, C_{t(n+1,r\to l)}$). For each user $u_{l,i} \in U_l$ on RB $c \in C_{t(j,l\to r)}$, we set $\lambda_{u_{l,i}}^c = i+1$ if $i = j$, and $\lambda_{u_{l,i}}^c = 0$ if $i \neq j$. Similarly, for each user $u_{r,i} \in U_r$ on RB $c \in C_{t(j,r\to l)}$, we set $\lambda_{u_{r,i}}^c = i+1$ if $i = j$, and $\lambda_{u_{r,i}}^c = 0$ if $i \neq j$. We now encode connectivity among nodes in G' into our construction by examining each node's incoming edges. For each user $u_{r,i} \in U_r$ for RB $c \in C_{t(j,l\to r)}$, we first check whether its corresponding node $v_{r,i}$ has incoming edges from any node $v_{l,g}$, and sort, if any, them by g in decreasing order (say, $v_{l,g1}, v_{l,g2}, \ldots$). Then for $c \in C_{t(g+1,l\to r)}$ we set $\lambda_{u_{r,i}}^c = n - g + 1$ if $g = g1$ (i.e. the largest index), and if $g \neq g1$, we set $\lambda_{u_{r,i}}^c = g - g'$ where g' is the next larger index than g (e.g. if $g = g2$ then $g' = g1$). Lastly, we set $\lambda_{u_{r,i}}^c = 0$ for $c \in C_{t(j,l\to r)}$ if $j \neq g+1$. Similarly, the values $\lambda_{u_{l,i}}^c$ for users $\in U_l$ on RB $c \in C_{t(j,r\to l)}$ are set in this way. Finally, we set the target aggregate value $k = T \times 2n + (n+2)(2n-1)$, which is the total number of RBs. This completes the construction of the frequency-domain status S.

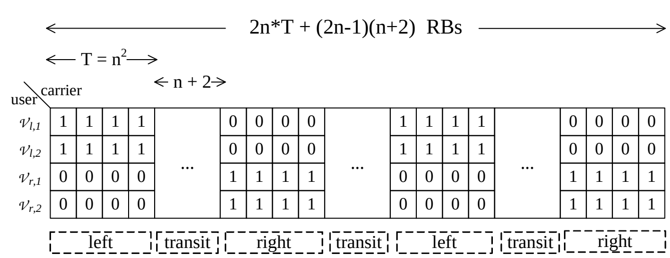


Fig. 3 The intermediate construction reduced from an example HAM-PATH-BG instance G', where G' is of 4 nodes (i.e. $n = 2$).

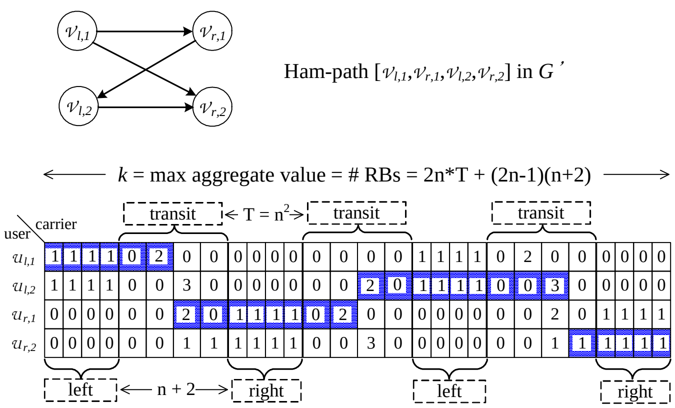


Fig. 4 The reduction from an example HAM-PATH-BG instance G', where G' consists of 4 nodes (i.e. $n = 2$). Dark-colored RBs represent a satisfiable contiguity strategy with aggregate value k.

We claim that our resulting construction S has a feasible allocation strategy if and only if G' contains a hamiltonian path. Indeed, suppose there is a hamiltonian path in G'. The allocation of the contiguous RB chunks to users in order of the sequence of nodes traversing a hamiltonian path achieves exactly the target aggregate value k, since the aggregate value for each "transit" region can be $n + 2$ only when there exists a directed edge untraversed. Such an allocation also conforms to the contiguity constraint, so it is a feasible strategy for S. Conversely, suppose that there is a contiguous allocation strategy C in S. In order to achieve the target value k, every user must be assigned in the end without being re-assigned discontiguous RBs, which forms a hamiltonian path in G. ∎

B. Computational intractability in practice

Since we have proved in Theorem 1 that optimizing the objective (1) is NP-hard, now our last hope for optimizing the objective (1) is probably "brute-force" search in the sense that it may work fine on the relatively small-sized input with help from high computing power. That is, even though this problem itself is NP-hard, we may solve the problem by trying all the possibilities if the size of the typical instance is small in practice. To examine whether or not brute-force search is practicable, we first evaluate the running time of brute-force search on this problem.

Lemma 1: The running time of brute-force search for optimizing the objective (1) under the contiguity constraint is $O(n!)$ if $n < m$, and $O(n^m)$ if $n \geq m$. (n users, m RBs)
The proof is given in the Appendix.

Unfortunately both numbers n, m are somewhat large in practice. For example, 3GPP LTE UL is planning to support a scalable bandwidth of 5, 10, 20 and possibly 15 MHz, each corresponds to 25, 50, 100, and 75 RBs, respectively [3], [4]. Moreover, we may have at least several tens of active users in a cell. Even in a sparse cell (say $n = 10$), it takes about 4 secs to complete the search (1 oper. $\approx$ 1 μs), which is too slow to schedule data every 1 ms in the real systems. Thus, we cannot optimize the objective (1) in practice either.

C. Upper bound of objective (1)

We conclude this section with a natural result on the upper bound of objective (1). Let Z and Z^* be algorithms to obtain the optimum OPT and OPT^* for the objective (1) under the contiguity constraint and without the constraint, respectively. Let $u(c)$ and $u'(c)$ be users assigned RB c by Z and Z^*, respectively.

Lemma 2: $OPT^* \geq OPT$

Proof: Since $\lambda^c_{u'(c)} \geq \lambda^c_{u(c)}$ for all c
$OPT^* = \sum_c \lambda^c_{u'(c)} \geq OPT = \sum_c \lambda^c_{u(c)}$ ∎

Therefore, the optimum OPT for the objective (1) under the contiguity constraint is at most the optimum OPT^* without the constraint.

III. Approximation Algorithm

In this section we first present *Alg5* to obtain $1/2$-approximation for this FDPS problem under contiguous RB constraint. This randomized approximation algorithm is however too complex to be used in the practical FDPS, but we present it here since it may give us an implication of the approximable limits of this problem.

We let $x_i^{ab} = 1$ if all the RBs between RB a and b (i.e. contiguous RBs from a to b) are assigned to user i, and $x_i^{ab} = 0$ otherwise. We then could optimize our scheduling problem by solving the following integer program:

$$\max \sum_i \sum_a \sum_{b \geq a} \sum_{t \in [a,b]} x_i^{ab} \lambda_i^t$$
$$\text{subject to} \sum_a \sum_{b \geq a} x_i^{ab} \leq 1 \qquad \forall i$$
$$\sum_i \sum_{a \leq t} \sum_{b \geq t} x_i^{ab} \leq 1 \qquad \forall t$$
$$x_i^{ab} \in \{0, 1\} \qquad \forall (i, a, b) \text{ triples}$$

We cannot solve this integer programming directly, since we proved in Theorem 1 that optimizing our objective is NP-hard, which means this integer program is NP-hard as well. So algorithm *Alg5* finds an approximation solution by using a linear relaxation of the integer programming as follows. We first relax the integrality constraint to read $0 \leq x_i^{ab} \leq 1$, then we can solve the resulting linear program. This gives us fractional values x_i^{ab} and guarantees that the objective is *at least* the integer optimum OPT:

$$\sum_i \sum_a \sum_{b \geq a} \sum_{t \in [a,b]} x_i^{ab} \lambda_i^t \geq OPT$$

We will now devise a *rounding scheme* to obtain integer values for the variables, which we call $\hat{x}^i_{ab}$. These values should satisfy all the constraints and also obtain close-to-optimum value. Suppose we have a small positive real number ϵ. We will do the following:

1) Solve the linear relaxation of the integer program, obtaining variables x^i_{ab}.
2) For each i, t pair initialize $C^i_t \leftarrow 0$
3) Sort the (i, a, b) triples for which $x^i_{ab} > 0$ in increasing order of a.
4) For each (i, a, b) triple:
   a) Define $\rho^i_{ab} \leftarrow \alpha x^i_{ab}$
   b) Let P^i_{ab} be the probability that by the time we consider (i, a, b), we have *already selected an*

interval[2] which shares the same i value or overlaps $[a, b]$.

c) If we have not yet selected any interval for user i nor any interval which overlaps $[a, b]$ then with probability $\rho^i_{ab}/(1 - P^i_{ab})$ select interval (i, a, b).

In order to bound the expected value of this rounding, we need to bound the probability of selecting interval (i, a, b). We will do this via the next two lemmata.

Lemma 3: Provided that $1 - P^i_{ab} \geq \rho^i_{ab}$ at the time we first consider (i, a, b), the overall probability of selecting (i, a, b) will be exactly ρ^i_{ab}.

Proof: The probability of selecting (i, a, b) is the conditional probability that we select (i, a, b) given that we have not yet selected an interval which shares the same i value or overlaps $[a, b]$ times the probability that we have not yet selected an interval which shares the same i value or overlaps $[a, b]$. The former probability is $\frac{\rho^i_{ab}}{1-P^i_{ab}}$ provided this is less than or equal to one, and the latter is $1 - P^i_{ab}$. Multiplying completes the proof. ∎

Lemma 4: As long as $\alpha \leq \frac{1}{2}$, when we consider interval (i, a, b) we will have $1 - P^i_{ab} \geq \rho^i_{ab}$.

Proof: Let (i, a, b) be the first triple considered for which this is not true. Since for every previously considered triple the lemma held, all previously considered (i', a', b') had selection probability exactly $\rho^{i'}_{a'b'}$. In addition, since we consider intervals in order of a value, any overlapping previous interval must include a. So the probability of previously selecting an interval with the same i or an interval overlapping $[a, b]$ will be bounded by:

$$P^i_{ab} \leq \sum\nolimits_{(i,a',b'):a'<a} \rho^i_{a'b'} + \sum\nolimits_{(i',a',b'):a'<a\leq b'} \rho^{i'}_{a'b'}$$
$$\leq (\alpha - \rho^i_{ab}) + (\alpha - \rho^i_{ab})$$

The second line follows from the fact that *all* intervals for i have sum of x^i_{ab} at most one (and similarly all intervals including a have sum of x^i_{ab} at most one). We conclude that if $\alpha \leq \frac{1}{2}$ then:

$$1 - P^i_{ab} \geq 1 - 2\alpha + 2\rho^i_{ab} \geq 2\rho^i_{ab}$$

∎

We can now bound the overall expected value of the solution.

Theorem 2: *Alg5* is a $\frac{1}{2}$-approximation for the PF objective (1).

Proof: Assuming $\alpha = \frac{1}{2}$, the probability of selecting (i, a, b) is at least $\frac{1}{2}x^i_{ab}$ and summing this probability over all i gives an expected value of half the linear program value. ∎

IV. Heuristic Algorithms

Although *Alg5* guarantees theoretically worst-case performance bound, due to its high complexity *Alg5* is impractical for wireless scheduling in the real systems. In this section we present a set of greedy heuristic algorithms for the objective

[2]Here we refer to an interval as a chunk of contiguous RBs.

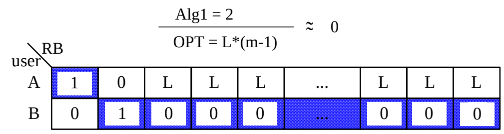


Fig. 5 Bad example (2 users, m RBs) for $Alg1$. Dark-colored RBs represent a resulting assignment by $Alg1$. L is a very large number.

(1) under contiguous RB constraint. Our greedy heuristics do not give guaranteed error bound, and moreover we believe that no practical greedy algorithms can give an approximation to this particular problem (we will show it by giving bad examples[3]). We however note that the approximation guarantee only reflects the performance of the algorithm on the most pathological instance which is generally not common in practice. Our heuristics fine-tuned to the typical instances of the problem might not perform well in their worst case scenarios, yet their overall performance is very good in practice, as shown in Section V

A. Alg1: carrier-by-carrier in turn

As a starter, our first greedy heuristic $Alg1$ is a very natural yet coarse adaptation of algorithm Z^* that optimizes objective (1) without the contiguity constraint (i.e. Z^* applies PF over each RB one-by-one in isolation). Emulating Z^*, $Alg1$ schedules data from RB1 to RBm in sequence, and for each RB c it assigns the best user i who 1) has the maximum PF metric value λ^c_i on c and 2) satisfies the contiguity constraint.

Algorithm 1 : Carrier-by-carrier in turn

Let U be the set of schedulable users
Let $A[m]$ be RB-to-user assignment status
for RB $c = 1$ to m **do**
  pick the best user $i \in U$ with largest value λ^c_i
  assign RB c to user i (i.e. $A[c] \leftarrow i$)
  Let I be RBs already assigned to user i
  if $I = \emptyset$ **then**
    $U = U - \{A[c-1]\}$
  end if
end for

Since $Alg1$ schedules data from one end side RB, it is not likely to even have a chance to try users' high metric value frequency portions. Figure 5 shows such a undesirable case (this also demonstrates $Alg1$ cannot give an approximation). Assignment of user B to RB2 prevents user A from being scheduled on subsequent RBs, which would otherwise greatly improve the result. We note that although such an extremely bad case above is not realistic (or might not exist in practice), this approach gives poor performance in general.

B. Alg2: largest-metric-value-RB-first

We have shown from $Alg1$ that scheduling RBs in sequence from one end side does not much help the problem. So, viewing this scheduling problem as simply a packing problem, adhering to its rule of thumb "pack large items first" may help

[3]In this section we mean by a "bad" example a problem instance where the heuristic will lead to very bad results.

in our case. Adopting such a quite intuitive judgement, $Alg2$ schedules RBs with largest metric value first, with the idea that later, RBs with small value may only do so much damage. And in fact, this approach can lead to an optimal result on the bad example in Figure 5, with nicely avoiding a snare in preceding RBs in order.

However, the contiguity constraint makes this problem much harder than the well-studied packing problem; it is uncertain how our action should be in the case that, for a certain user i a candidate RB is not adjacent to RBs already assigned to i (e.g. RB3 is first assigned to i, then the next largest value one is RB5 of i. If RB4 is already assigned to other user, then the contiguity constraint prohibits i from being assigned to RB5. Should we however assign RB5 to i if RB4 is still unoccupied?). Strictly adhering to the argument "pack large items first", $Alg2$ assigns those candidate RBs anyway unless it clearly violates the contiguity constraint (i.e. it assigns RB5 to i).

Algorithm 2 : largest-metric-value-RB-first

```
Let V be the sorted list of all the metric values λ_i^c in decreasing order
Let S be the set of not-yet-assigned RBs
k ← 1
while S ≠ ∅ do
    pick RB c with k^th largest metric value λ_i^c ∈ V, c ∈ S
    Let I be RBs already assigned to user i
    if none is yet assigned to RBs between I and c then
        Let C' be all RBs located between I and c
        C' = C' ∪ {c}
        assign all RBs ∈ C' to user i
        S = S − C';   V = V − {λ_i^{C'}};   k ← 1
    else
        k ← k + 1
    end if
end while
```

The price we pay for this a bit aggressive strategy is that we have to assign all the "in-between" RBs to a candidate user (i.e. it assigns RB5 to i, which as a result comes with assignment of RB4 to i, since i is already assigned RB3). The downside of this approach comes from this by-product assignment. Since the length of such "in-between" RBs is arbitrary, a potential improvement in those RBs is likely to be cancelled. Figure 6 exemplifies such a case (this also shows $Alg2$ cannot give an approximation). Assignment of user A to the largest value RBs (each in the end sides) obstructs assignment of user B on the "in-between" RBs, which would otherwise greatly improve the result. It turns out that $Alg2$'s strategy is too aggressive to attain the potential multiuser frequency diversity gain, which incurs performance penalty.

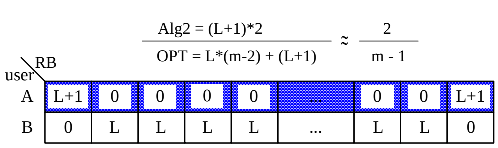


Fig. 6 Bad example (2 users, m RBs) for $Alg2$. Dark-colored RBs represent a resulting assignment by $Alg2$. L is a very large number.

C. Alg3: riding peaks

Learned lesson from the drawback of $Alg2$, we would like to utilize each user's high valued RBs as much as possible.

Let's look at the PF metric values ($\lambda_i^c(t) = r_i^c(t)/R_i(t)$) at time slot t. One key observation is that, for each user i the denominator ($R_i(t)$) is constant for all RBs, so the resulting value for each RB c is dominated by channel rate ($r_i^c(t)$) only scaled down/up to the current service rate. Thus, at time slot t each user's RB values fluctuate exactly as the channel rate changes between RB to RB. However, another fundamental physical layer characteristic is that in multi-carrier systems the channel SNR values (i.e. CQI) are correlated in both time and frequency (depending on the Doppler effect and the delay spread) [12], [20], [23]. In other words, if for each user i RB c has good channel rate, then the neighboring RBs ($c-1, c+1$) have high channel rate as well with high probability[4].

So the key idea of $Alg3$ is to "ride users' peaks" in frequency domain, by exploiting such correlations. Recall that the conventional PF algorithm rides peaks in time domain. $Alg3$, in fact, extends $Alg2$'s rule of thumb: 1) look at large value RBs first; 2) make them augmented by one neighbor RB. This second rule enforces a bit conservative contiguity condition (i.e. for a certain user i a candidate RB must be *adjacent* to RBs already assigned to i).

We first have all metric values λ_i^c sorted in decreasing order, then pick the largest value element (i.e. a user-RB pair) that does not violate the adjacency augmentation rule. This makes the algorithm much sensitive to the metric value fluctuation among RBs. In the bad example of Figure 6, $Alg3$ can attain an optimal result by allocating the contiguous high value RBs to user B (user A will be assigned to only one of end side RBs, since they are not adjacent each other). Figure 7 illustrates the "peak riding" of $Alg3$. In the beginning user A is first assigned to its high value RBs, while user B and C are assigned to their peak RBs a little bit later. In the end they are all assigned to the RBs around their peaks according to the rules. Note that $Alg2$ fails to allocate user B to its high value RBs, since B's peak RB is surrounded by a bit higher A's peak RBs.

[4] A time-delayed channel model for the duration of an SC-FDMA symbol is given by a channel vector h. By adding the cyclic prefix, channel in frequency domain is given by $H_f = FT(h)$. Normally the number of subcarriers, N, is much larger than the number of resolvable paths, L. H_f has N elements that are made by linear combination of L independent random variables. Hence, at most only L of them are independent, and the rest can be written as linear combination (correlation) of others.

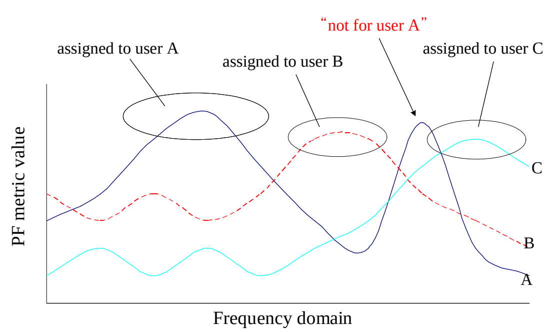


Fig. 7 $Alg3$ rides peaks.

Algorithm 3 : riding peaks

Let V be the sorted list of all the metric values λ_i^c in decreasing order
Let S be the set of not-yet-assigned RBs
$k \leftarrow 1$
while $S \neq \emptyset$ **do**
  pick RB c with k^{th} largest metric value $\lambda_i^c \in V$, $c \in S$
  Let I be RBs already assigned to user i
  if (c is adjacent to I) or ($I = \emptyset$) **then**
    assign RB c to user i
    $S = S - \{c\}; \quad V = V - \{\lambda_i^c\}; \quad k \leftarrow 1$
  else
    $k \leftarrow k + 1$
  end if
end while

This "peak riding" approach so far seems quite good. There exist, of course the cases where it can lead to very bad solutions. If for a certain user the channel rate across RBs changes arbitrarily, then sticking to peaks is not likely a good strategy. As mentioned earlier, we however can find typical instances displaying the frequency-domain correlation among RBs, and in fact, this approach can lead to a measurable improvement on both throughput and short-term fairness in the realistic UL SC-FDMA scenarios as shown in Section V.

Figure 8 shows an bad example for $Alg3$, which also demonstrates it cannot give an approximation. In this example we assign user A and B to their peak RBs with the frequency-domain correlation in mind (i.e. in the hope that their *adjacent* RBs also have high metric values), but no such correlation here leads to a very bad solution.

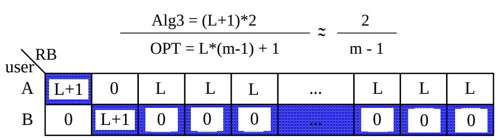


Fig. 8 Bad example (2 users, m RBs) for $Alg3$. Dark-colored RBs represent a resulting assignment by $Alg3$. L is a very large number.

D. Alg4: RB grouping

Given that the frequency domain exhibits a correlation (more precisely, correlation between two adjacent RBs), $Alg3$ is expected to yield good performance. As mentioned in Section IV-C, the channel quality values are indeed correlated in both time and frequency. However, in general the correlation in the frequency-domain is not as strong as the one in the time-domain (frequency-selective fading distortion) [20], [21]. That implies that we have the overall frequency correlation but its granularity may not be as small as one RB (i.e. the smooth lines in Figure 7 may need to be changed to the uneven ones). Figure 9 (overall fluctuation similar to Figure 8 but with some jitters) shows that such a condition incurs poor results by $Alg3$. Since $Alg3$ relies on the strong frequency-domain correlation, it is easily cheated by the small-scale variation. In the figure, user B is falsely assigned to the abrupt peak, user A is trapped by the sudden drop, and in the end user C expands its region to that point.

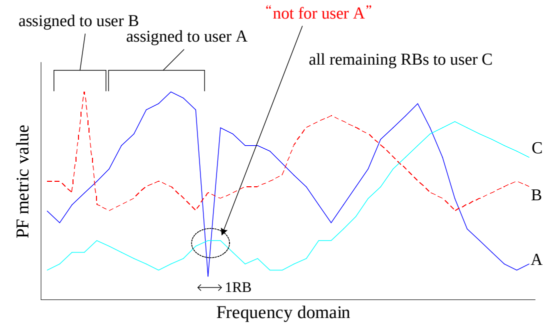


Fig. 9 $Alg3$ suffers from small-scale variation.

To deal with such small-scale variation, it would help to extend our unit of consideration (i.e. the number of contiguous RBs that we view at a time). In the example of Figure 8 and 9, if we consider a group of x contiguous RBs (say $x = 3$) instead of one RB, then we have a wider view enough to obtain an optimal solution. Thus, this RB grouping seems helpful to catch a bit large-scale fluctuation. $Alg4$ makes use of RB grouping to manage the weak frequency-domain correlation. The following questions may arise: "how big should a group be?", "is it a variable size?", and "freedom of positioning?". The harder we try to set up good criteria regarding those questions, it becomes more a quagmire due to the NP-hard nature. Here we set up simple rules: 1) divide m RBs into n groups; 2) apply the "peak riding" over those RB groups. Thus, $Alg4$ is an RB-grouping version of $Alg3$; $Alg4$ "rides peaks" with the granularity of RB groups (one group = $\lceil \frac{m}{n} \rceil$ RBs). Notice that as n (i.e. the number of users) grows, the group size gets smaller (i.e. we see the smaller-scale fluctuation). As a ground for our choice of $\lceil \frac{m}{n} \rceil$, we argue that it would be beneficial to see the small-scale fluctuation with large number of users, since high multiuser frequency diversity can facilitate the potential improvement from the small-scale peaks.

One can easily find a bad example for $Alg4$ and its inapproximability as well (example structure is similar to Figure 8). However, such extremely bad instances are unlikely to happen in practice, and in fact, $Alg4$ exhibits constantly better performance over $Alg3$ on the real traces, particularly when the number of users is not large (as n grows, $\lceil \frac{m}{n} \rceil$ RBs becomes 1 RB).

V. SIMULATIONS

To evaluate the performance of our heuristics, SC-FDMA uplink system level simulations have been conducted based on 3GPP LTE system model. We use traces generated as specified in 3GPP deployment evaluation [2], based on Typical Urban channel model. Table 1 summarizes a list of the default simulation parameters and assumptions.

We analyze the performance of the algorithms in terms of throughput as well as short-term fairness[5], and assess how well they emulate the proportional fair criteria in this FDPS setting. However, since it is NP-hard to optimize objective (1) under the contiguity constraint, we do not have such an optimal algorithm in our hand. Thus, we use an algorithm that

[5]A well-known problem of the conventional time-domain PF scheduling is its poor short-term fairness.

TABLE I Simulation parameters

Parameter	Setting
System bandwidth	20 MHz
Subcarriers per RB	12
RB bandwidth	180 kHz
Number of RBs	96
Cell-level user distribution	Uniform
Number of active users in cell	10, 20, 30, 40, 50
Traffic model	Infinitely backlogged
Transmission time interval (TTI)	1 ms
Channel model	Typical Urban
User speed	3, 30, 120 km/h
User receiver	1x2/MMSE/ZF
Modulation/coding rate settings	QPSK: 1/3, 1/2, 2/3, 3/4 16QAM: 1/2, 2/3, 3/4
HARQ model	Ideal chase combining
HARQ Aak/Nack delay	8 ms
Max. number of HARQ retransmission	3

optimizes objective (1) without the constraint as our reference, and we refer to this algorithm as OPT^*. Note that, as shown in Lemma 2, OPT^* offers an upper bound of the optimum. We use Jain's fairness index [14], measured by the data-rate fairness criterion[6]:

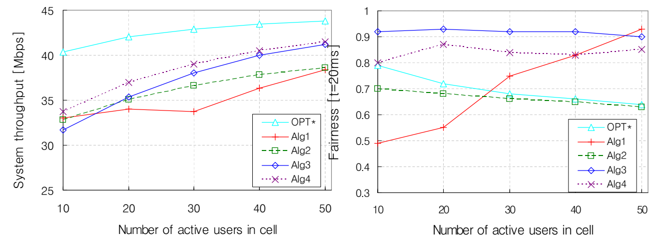
$$F_\phi(\Delta t) = \frac{[\sum_{i=1}^{N} \phi_i(\Delta t)]^2}{[N \cdot \sum_{i=1}^{N} \phi_i(\Delta t)^2]},$$

where $\phi_i(\Delta t)$ denotes the actual data-rate user i achieved in time interval Δt, with N users in the system.

We first measure the system throughput of our algorithms with varying the number of active users in the cell. As shown in Figure 10(a), $Alg4$ results in the highest throughput among our heuristics, followed by $Alg3$, $Alg2$, and $Alg1$. This trend seems to match with our expectation, since $Alg4$ and $Alg3$ contain more advanced heuristic idea than the other two. In general, $Alg3$ performs better than $Alg1$ and $Alg2$ because $Alg3$ seeks to take advantage of each users' peak while both $Alg1$ and $Alg2$ are not so fine-tuned enough to effectively utilize multiuser frequency diversity. However, as seen from Figure 10(a), $Alg3$ displays the poor performance with small number of active users (e.g. when $n = 10$, it yields even lower throughput than $Alg1$ and $Alg2$). Such a result shows the implication of the weak frequency-domain correlation, by which $Alg3$ is easily misled into bad solutions. On the other hand, $Alg4$ contantly outperforms the other three algorithms in all scenarios. $Alg4$ deals with this small-scale variations by widening its view to $\lceil \frac{m}{n} \rceil$ RBs. In the case of small number of active users, $Alg4$ expands the RB-group size, and it rides each users' aggregated peak by catching a bit large-scale fluctuation (it attains 84% of OPT^* while $Alg3$ gets 77%.). As n grows, $Alg4$ adaptively lessens the view so as to exploit the small-scale fluctuation, and its performance gets similar to $Alg3$ (when $n = 50$, $Alg4$ and $Alg3$ reach 95% of OPT^* while the other two get around 86%). It is worth stressing again that OPT^* does not represent the optimum of our objective but simply shows an upper bound of it, where the actual optimum lies between $Alg4$ and OPT^* in general.

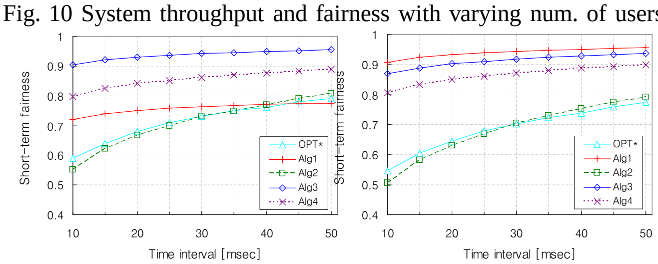
We now evaluate the short-term fairness of our algorithms

[6] $F_\phi(\Delta t)$=1 implies that all users received equal data-rate within time Δt.



(a) cell throughput (b) fairness index (t=20ms)

Fig. 10 System throughput and fairness with varying num. of users



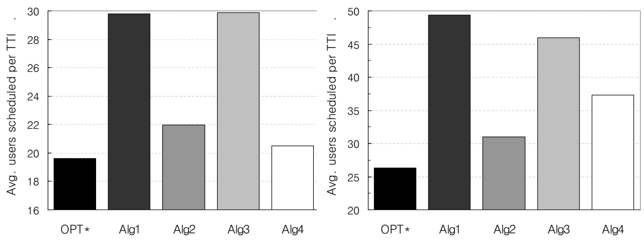
(a) 30 active user case (b) 50 active user case

Fig. 11 Short-term fairness with varying time interval

with varying the number of active users. Figure 11(a) shows the short-term data-rate fairness $F_\phi(\Delta t)$, in the cell of 30 active users, with extending the time interval window Δt from 10 ms (i.e. 10 TTI) to 50 ms. In this setting, $Alg3$ consistantly outperforms other algorithms in all intervals, followed by $Alg4$, $Alg1$, and $Alg2$. To understand why $Alg3$ provides better short-term fairness than others in this setting, we record the number of users scheduled per one TTI for each algorithm. Figure 12(a) plots the average number of users scheduled per one TTI when 30 users are active in the cell. We can see that all of 30 users are likely assigned to all 96 RBs by $Alg3$ and $Alg1$.[7] However, the crucial difference is that $Alg1$ is likely to allocate arbitrary rate on each user while $Alg3$ seeks to assign users their peak RBs, which helps short-term "fair share" of the frequency resource. Figure 10(b) presents the short-term fairness of 20 ms interval window with increasing number of active users. Interestingly, $Alg1$ offers the best fairness when the number of users is large (e.g. $n = 50$). See also Figure 11(b) and 12(b) for fairness and the average number of users scheduled per a TTI with 50 users. With the large number of users, $Alg1$ is able to balance users' rates, but those are not likely from peak RBs.

At this point we note that comparison by each single metric separately, however, does not provide us much meaningful insight on the performance. Moreover, achieving both high throughput and fairness is a somewhat conflicting goal in general. For example, $Alg1$ performs even better than OPT^* in terms of the short-term fairness while OPT^* yields 127% greater throughput over $Alg1$. Hence, we need to compare the algorithms by a comprehensive metric that takes both throughput and fairness into account. Such a balance is pursued by the proportional fair criteria (i.e. maximizing $\sum_i \log R_i$, where R_i is the long-term service rate for user i), which in fact is our

[7] This result seems quite intuitive in the sense that $Alg3$ and $Alg1$ make assignment decision on one single RB at a time while $Alg2$ and $Alg4$ assign potentially multiple RBs to a certain user at a time.



(a) 30 active user case (b) 50 active user case

Fig. 12 Average num. of users scheduled per 1 TTI

ultimate objective function. We recall that the conventional PF algorithm optimizes the proportional fair criteria so that it maximizes long-term throughputs of the users *relative* to their channel conditions, and our main goal is to maximize the PF criteria in the FDPS context. Now we assess how well our heuristics emulate the proportional fair objective in our problem framework. In the following table we show the values of the PF criteria with 30 active users in the cell.

	$\sum_i \log R_i$
OPT*	223.1
Alg1	216.5
Alg2	218.9
Alg3	220.6
Alg4	221.6

We can see that $Alg4$ has the highest value of $\sum_i \log R_i$, followed by $Alg3$, $Alg2$ and $Alg1$. We obtain the same trend (with similar gaps between values) in all other scenarios. As we underlined earlier, OPT^* simply represents an upper bound of the optimum of our objective, so the actual optimum has a value of $\sum_i \log R_i$ between $Alg4$ and OPT^*. Therefore, among our heuristics $Alg4$ has the value of the PF criteria closest to the actual optimum, and it emulates best the PF criteria in UL FDPS setting.

VI. Conclusions

Due to its single carrier property of SC-FDMA, LTE UL requires the RBs allocated to a single user to be contiguous in frequency. In this paper we explored this fundamental problem of frequency-domain scheduling under contiguous RB allocation constraint. We investigated how to adapt the time-domain PF algorithm to this problem framework. We first showed the NP-hard nature of this problem, then presented a set of practical algorithms fine tuned to this problem. Among them, an algorithm that exploits the frequency-domain correlations in concert with an adaptive RB grouping technique emulates best the PF criteria in the LTE UL FDPS context.

Finally we believe that no practical wireless scheduling algorithms can give an approximation to this particular problem, but whether there actually exists such an algorithm or not still remains as an open problem.

References

[1] Physical layer aspects for evolved Universal Terrestrial Radio Access (UTRA). 3GPP TR 25.814
[2] Technical specification group radio access networks - Deployment aspects. 3GPP TR 25.943
[3] Requirements for Evolved UTRA and UTRAN (Release 7). 3 GPP TR 25.913 V7.0.0, June 2005.
[4] TSG-RAN WG1 #42, R1-050737, “Bandwidth of resource blocks for DL OFDMA”, London, UK, Sep, 2005.
[5] Moray Rumney. 3GPP LTE: Introducing SIngle-Carrier FDMA Agilent Measurement Journal, 2008.
[6] 3GPP TSG-RAN WG2 Meeting #57, R2-070585, “Resource fragmentation in LTE uplink”, St. Louis, USA, Feb, 2007.
[7] M. Andrews. A survey of scheduling theory in wireless data networks. *IMA*, 2005.
[8] M. Andrews. Instability of the proportional fair scheduling algorithm for HDR. *IEEE Transactions on WIreless Communications*, 2004.
[9] M. Andrews, L. Qian, and A. Stolyar. Optimal utility based multi-user throughput allocation subject to throughput constraints. *IEEE INFOCOM*, 2005.
[10] M. Andrews and L. Zhang. Scheduling algorithms for multi-carrier wireless data systems. *ACM MobiCom*, 2007.
[11] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and S. Viterbi. CDMA/HDR: A bandwidth efficient high speed data service for nomadic users. *IEEE Communications Magazine*, 2000.
[12] E. Biglieri, J. Proakis, and S. Shamai. Fading channels: Information-theoretic and communications aspects. *IEEE Trans. on Information Theory*, 2000.
[13] R. Cohen and L. Katzir. Computational analysis and efficient algorithms for micro and macro OFDMA scheduling. *IEEE INFOCOM*, 2008.
[14] R. Jain, D. M. Chiu, and W. Hawe. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems. DEC Research Report TR-301.
[15] A. Jalali, R. Padovani, and R. Pankaj. Data throughput of CDMA-HDR a high efficiency -high data rate personal communication wireless system. *IEEE VTC*, 2000.
[16] J. Kleinberg and E. Tardos. Algorithm Design. *Addison Wesley*, 2005.
[17] H. Kushner and P. Whiting. Asymptotic properties of proportional-fair sharing algorithms. *Allerton*, 2002.
[18] X. Liu, E. Chong, and N. B. Shroff. A framework for opportunistic scheduling in wireless networks. *Computer Networks*, 2003.
[19] A. Pokhariyal, T. E. Kolding, and P. E. Mogensen. Performance of Downlink Frequency Domain Packet Scheduling for the UTRAN Long Term Evolution. *IEEE PIMRC*, 2006.
[20] B. Sklar. Rayleigh fading channels in mobile digital communication systems, Part I: Characterization. *IEEE Communications Magazine*, 1997.
[21] B. Sklar. Rayleigh fading channels in mobile digital communication systems, Part II: Mitigation. *IEEE Communications Magazine*, 1997.
[22] D. Tse. Multiuser diversity in wireless networks. http://www.eecs.berkeley.edu/ dtse/stanford416.ps , 2002.
[23] W. Wang, T. Ottosson, M. Sternad, A. Ahlen, and A. Svensson. Impact of multiuser diversity and channel variability on adaptive OFDM. *IEEE VTC*, 2003.
[24] C. Wengerter, J. Ohlhorst, and A. G. E. Elbwart. Fairness and Throughput Analysis for Generalized Proportional Fair Frequency Scheduling in OFDMA. *IEEE VTC*, 2005.
[25] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, and Y. Wu. UnChannelize the Channels in WLAN. *ACM MobiCom*, 2007.

Appendix

A. Proof of Lemma 1

Proof: First, we consider the number of possible ordering when k users are assigned to m RBs under the contiguity constraint where $n \geq m$ (i.e. the number of users is greater than that of RBs). We initially pre-position k users in sequence: $n(n-1)(n-2)\cdots(n-k+1) = \prod_{j=0}^{k-1}(n-j)$.

With keeping those orders, we fill the remaining $m-k$ spots by adding some of those k users under contiguity constraint. We have m spots and k users, which is represented by a positive integer-valued vector $(x_1, x_2, \cdots, x_k)$:

$$x_1 + x_2 + \cdots + x_k = m \quad x_i \geq 1, i = 1, \cdots, k$$

So, there are $\binom{m-1}{k-1}$ distinct vectors satisfying the condition. Hence, the number of possible ordering is: $\binom{m-1}{k-1}\prod_{j=0}^{k-1}(n-j)$. We have $1 \leq k \leq m$ (we cannot assign more than m users at a time) users, so the total search space is:

$$T(n,m) = \sum_{i=1}^{m}\left[\binom{m-1}{i-1}\prod_{j=0}^{i-1}(n-j)\right] = O(n^m)$$

In the case when $n < m$, we have $1 \leq k \leq n$ users available to be scheduled, then the total search space is: $T(n,m) = \sum_{i=1}^{n}\left[\binom{m-1}{i-1}\prod_{j=0}^{i-1}(n-j)\right] = O(n!)$ ■