Frequency-Domain Packet Scheduling for 3GPP LTE Uplink

Hongkun Yang, Fengyuan Ren, Chuang Lin, Jiao Zhang
Department of Computer Science and Technology, Tsinghua University
Email: {yanghk07, renfy, chlin, zhangjiao08}@csnet1.cs.tsinghua.edu.cn

***Abstract*—In this paper, we investigate the frequency-domain packet scheduling (FDPS) problem for 3GPP LTE Uplink (UL). Instead of studying a specific scheduling policy, we provide a unified approach to tackle this issue. First we formalize a general LTE UL FDPS problem which is suitable for various scheduling policies. Then we prove that the problem is MAX SNP-hard, which implies that approximation algorithms with constant approximation ratios are the best that we can hope for. Therefore we design two approximation algorithms, both of which have polynomial runtime. Subsequently, we analyze the two algorithms and find their approximation ratios. The first algorithm is easy to follow, since it is based on a simple greedy method. The second one is based on the local ratio technique and it can approximately solve the LTE UL FDPS problem with a approximation ratio of 2.**

I. Introduction

Third Generation Partnership Project (3GPP) has been studying the 3G Long-Term Evolution (LTE) system ever since 2004. The LTE standardization aims at developing future cellular technologies which can provide reduced latency, higher user data rates, improved system capacity and coverage, and reduced cost [1]. In order to achieve this goal, an evolution of the radio interface as well as the radio network architecture should be considered.

Orthogonal Frequency Division Multiplexing (OFDM) radio technology has been selected for the LTE downlink radio access scheme because of its robustness against multi-path fading, higher spectral efficiency and bandwidth scalability [2]. On the other hand, requirements for the LTE uplink differ from those for the LTE downlink in several aspects, among which power consumption is a primary concern for User Equipment (UE) terminals [3].

However, OFDM has a high peak-to-average power ratio (PAPR). Such an undesirable high PAPR requires expensive and inefficient power amplifiers with high requirements on linearity, which subsequently increases the cost of the UE terminal and drains the battery faster [4]. Thus, given the high PAPR and related loss of efficiency associated with OFDM signaling, we have to seek an alternative to OFDM for use in the LTE uplink radio access scheme.

In the uplink, LTE uses a pre-coded version of OFDM, which is called Single Carrier Frequency Division Multiple Access (SC-FDMA) [4]–[6]. SC-FDMA retains the multi-path resistance and flexible sub-carrier frequency allocation offered by OFDM. Moreover, it has a significantly low PAPR like traditional single-carrier formats such as GSM, which compensates for the drawback with normal OFDM.

In the LTE uplink, the bandwidth (e.g. the number of SC-FDMA sub-carriers) is divided into separable chunks denoted as resource blocks (RBs). An RB is considered as the minimum scheduling resolution in the time-frequency domain. Frequency domain packet scheduling (FDPS) allocates different RBs to individual users according to their information such as channel conditions and lengths of queues, which achieves simultaneous frequency-domain multiplexing in concert with time-domain scheduling [7]. In order to obtain the most advantageous schedule, the base station needs to know the information of all users and all RBs, then performs the RB-to-user assignment according to the selected scheduling policy in each Transmission Time Interval (in LTE, TTI=1ms).

Moreover, in the LTE uplink, all the RBs allocated to a single user must be contiguous in frequency domain within each time slot, because the underlying waveform of SC-FDMA is essentially single-carrier [5]. This contiguous constraint introduces intricacy to the LTE uplink frequency-domain packet scheduling (LTE UL FDPS) problem, and the constraint is sufficient to make the problem hard. Thus, new scheduling schemes and tractable algorithms need to be customized for the LTE UL FDPS problem.

The LTE UL FDPS problem has been addressed in the literature. J. Lim and H.G. Myung et al. propose two utility-based scheduling schemes for SC-FDMA systems [8]. Afterwards, this work is improved by taking the delayed channel state information into consideration [9]. Recently, S. Lee et al. formalize the proportional fair FDPS problem (PF-FDPS) for the LTE uplink as a combinatorial optimization problem and prove that the problem is NP-hard [7]. They propose four approximation algorithms based on heuristics and conduct an extensive simulation study on the performance of the four heuristic algorithms.

All of the above work assumes an infinitely backlogged model in which for each user there are always packets available for service. However, this is not always the case in practical systems. In reality, packets will be generated for each user according to an arrival process, and hence there may be some users which do not have packets in each time slot. Furthermore, M. Andrews et al. point out that the proportional fair scheduling does not work so well when the queues are fed by an admissible arrival process [10]. In particular it can result in the unstability of queues [11]. Therefore, it might be preferable that we consider scheduling policies which combine system utility maximization and queue stability.

However, the selection of the scheduling policy for a specific LTE uplink system depends on a case by case analysis and this is not the focus of our work. In this paper, rather than a particular scheduling policy, we propose a universal solution for the LTE UL FDPS problem. We define a profit function which indicates the profit gained by allocating a set of contiguous RBs to an active user. The profit function is capable of expressing various scheduling policies, including the proportional fair scheduling and scheduling policies that combine utility maximization and queue stability. Based on the profit function, we formalize a general scheduling problem for the LTE uplink which can cover many LTE uplink scheduling policies.

Furthermore, we address the hardness of LTE UL FDPS. We prove that the scheduling problem is MAX SNP-hard, which indicates that it is very unlikely to efficiently approximate the problem within a certain ratio, and that approximation algorithms with constant approximation ratios are the best that we would have.

Afterwards we provide two approximation algorithms for LTE UL FDPS in company with provable approximation ratios. The first algorithm is based on a greedy method. It is intuitive and easy to follow, but its approximation ratio is bounded by a slowly increasing function of the number of active users. The second one is designed on the basis of the local ratio technique [12]. It is relatively delicate, but achieves a constant approximation ratio of 2.

The remainder of the paper is organized as follows. Section II introduces a necessary background in the theory of approximation and complexity. Section III gives the system model we study and formalizes the LTE UL FDPS problem. Section IV proves that LTE UL FDPS is MAX SNP-hard and shows that polynomial time approximation algorithms with guaranteed approximation ratios are necessary for practical LTE uplink systems. In Section V and VI respectively, we give two approximation algorithms computable in polynomial time and find their approximation ratios. Finally, we conclude the paper in Section VII.

II. Preliminaries

In this section, we provide a brief introduction to the theory of approximation and complexity we need in this work.

A. Approximation Ratio of an Approximation Algorithm

In practice, many optimization problems are NP-hard, which means that exact solutions of these problems are widely believed to be time consuming. This necessities efficient approximation algorithms which always compute solutions close to the optimum. However, different approximation algorithms can achieve different degrees of approximation for the same problem. So we introduce the definition of approximation ratio to qualify the degree of approximation achieved by an approximation algorithm.

Assume that G is an instance of a maximization problem[1]. We denote the size of its input by $|G|$ and its optimal value by $OPT(G)$. Let ALG be an approximation algorithm for the maximization problem. For instance G, we denote the value of ALG by $ALG(G)$. We say that ALG has an approximation ratio of $\rho(|G|)$ [13] if, for any instance G, $OPT(G)$ is within a factor of $\rho(|G|)$ of $ALG(G)$:

$$OPT(G) \leq \rho(|G|) \cdot ALG(G)$$

We also call ALG a $\rho(|G|)$-approximation algorithm. When the approximation ratio is independent of the input size $|G|$, we will use the terms approximation ratio of ρ and ρ-approximation algorithm, indicating no dependence on $|G|$.

Note that $\rho(|G|)$ (or ρ) always ≥ 1, and a smaller value of $\rho(|G|)$ (or ρ) indicates that the approximation algorithm has a better performance in a worst-case sense. In particular, when $\rho = 1$, the approximation algorithm ALG essentially finds the optimal solution for any instance G.

B. Polynomial-Time Approximation Scheme

A polynomial-time approximation scheme (PTAS) [13] for a maximization problem is an approximation algorithm that takes as an input not only an instance of the problem, but also a value $\epsilon > 0$ such that for any fixed ϵ, the scheme is a $(1+\epsilon)$-approximation algorithm which is computable in time polynomial in the size of the input instance.

In a technical sense, a PTAS is the best one can hope for an NP-hard optimization problem, assuming $P \neq NP$.

C. L-Reduction

Suppose that $\mathcal{A}$ and $\mathcal{B}$ are maximization problems,. An L-reduction [14] from $\mathcal{A}$ to $\mathcal{B}$ is a pair of functions R and S, both computable in polynomial time, with the following two additional properties:

First, if X is an instance of $\mathcal{A}$ with optimum $OPT(X)$, then $R(X)$ is an instance of $\mathcal{B}$ with optimum $OPT(R(X))$ that satisfies

$$OPT(R(X)) \leq \alpha \cdot OPT(X), \tag{1}$$

where α is a positive constant.

Second, if s is any feasible solution of $R(X)$, then $S(s)$ is a feasible solution of X such that

$$OPT(X) - VAL(S(s)) \leq \beta \cdot (OPT(R(X)) - VAL(s)), \tag{2}$$

where β is another positive constant particular to the reduction and VAL denotes the value of the feasible solution in both instances. (2) guarantees that S returns a feasible solution of X which is not much more suboptimal than the given solution of $R(X)$. In particular, if s is the optimal solution of $R(X)$, then $S(s)$ is the optimal solution of X.

L-reductions have the composition property [14]:

Lemma 1: If (R, S) is an L-reduction from problem $\mathcal{A}$ to problem $\mathcal{B}$, and (R', S') is an L-reduction from problem $\mathcal{B}$

[1]We focus on maximization problems in this paper since LTE UL FDPS is a maximization problem.

to problem $\mathcal{C}$, then their composition $(R \cdot R', S' \cdot S)$ is an L-reduction from $\mathcal{A}$ to $\mathcal{C}$.

D. MAX-SNP Hardness

In computational complexity theory, SNP (from Strict NP) is a complexity class containing a limited subset of NP based on its logical characterization in terms of graph-theoretical properties. The class MAX SNP is a subset of optimization problems derived from SNP. The formal definition of MAX SNP can be found in [14]. A problem is said to be MAX SNP-hard if all MAX SNP problems can be L-reduced to this problem. Note that the problem itself may not necessarily be MAX SNP.

MAX SNP-hard problems are hard to approximate. It is shown in [14] that

Lemma 2: Any MAX SNP-hard problem does not have a PTAS unless $P = NP$.

According to Lemma 1, to prove that a problem $\mathcal{B}$ is MAX SNP-hard, it suffices to present an L-reduction from a known MAX SNP-hard problem $\mathcal{A}$ to $\mathcal{B}$.

E. Job Interval Selection Problem with k Intervals per Job

The job interval selection problem with k intervals per job (JISPk) is stated as follows [15]:

Input: We are given n job, each of which is associated with k intervals on the real line. Thus we have $k \cdot n$ intervals. For each interval l a starting time s_l and a finishing time $f_l (> s_l)$ is known, $l = 1, \cdots, k \cdot n$. All starting and finishing times are integers. An interval is said to be active at time t if and only if $t \in [s_l, f_l)$. Two interval intersect if and only if there is a time t during which both intervals are active.

Goal: Select as many intervals as possible such that (i) for each job, at most one interval is selected from its associated k intervals, and (ii) no two selected intervals intersect.

Measure: The number of intervals selected.

[15] presents an L-reduction from MAX-3SAT-B to JISP2. Since MAX-3SAT-B is MAX SNP-hard [14], according to Lemma 1, [15] essentially proves that

Lemma 3: JISP2 is MAX SNP-hard, and hence it does not have a PTAS unless $P = NP$.

III. Problem Formulation

A. System Model

We consider a cellular network whose uplink system bandwidth is divided into m RBs. Besides, the network has a single base station and n active wireless users. We denote the set of all RBs by M ($M = \{1, 2, \cdots, m\}$) and the set of all users by N ($N = \{1, 2, \cdots, n\}$).

At each time slot, the base station can allocate m RBs to n users. Specifically, each user can be assigned to a set of contiguous RBs, and each RB is assigned to at most one user. Figure 1 is an example of feasible FDPS scheduling for the LTE uplink.

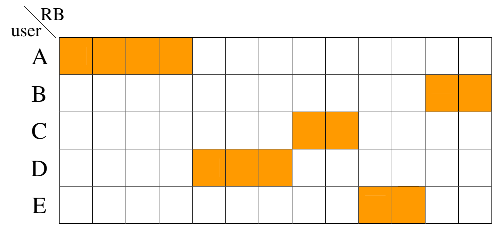

Fig. 1. A feasible FDPS scheduling for the LTE uplink. $m = 13, n = 5$. The orange colored (user, RB) pairs denote the RB-to-user assignment of the feasible schedule. Note that the RBs assigned to each user must be contiguous in frequency domain.

We denote by A the collection of all sets of contiguous RBs; in other words, $A \subseteq \mathcal{P}(M)$ ($\mathcal{P}(M)$ denotes the power set of M, i.e. the collection of all subsets of M), and $\forall a \in A$, $a = \{i, i+1, \cdots, i+l\}$, $1 \leq i \leq i+l \leq m$. For $a, b \in A$, we call a intersects b if $a \cap b \neq \emptyset$. $\forall a \in A$, we use $head(a)$ to denote the smallest RB of a, and $tail(a)$ the largest RB of a. We use the boolean variable x_i^a to indicate whether or not a set of contiguous RBs a is assigned to user i. User i gets $a \in A$ if and only if $x_i^a = 1$.

We define the profit function $p : A \times N \rightarrow \mathbb{R}^{\geq 0}$. $p(a, i)$ indicates the profit gained by assigning $a \in A$ to user i in the schedule. The profit $p(a, i)$ is a general term and might be different for different scheduling policies. We only require that $p(a, i) \geq 0$ for all $a \in A, i \in N$.

We can use $p(a, i)$ to represent various scheduling policies. For example, $p(a, i) = \sum_{c \in a} \lambda_i^c$ is the proportional fair scheduling studied in [7], where λ_i^c is the PF metric value that user i has on RB c. Moreover, we can express the three objective functions of [10] as $p(a, i) = Q_i^s \sum_{c \in a} r(i, c)$, $p(a, i) = Q_i^s \min\{Q_i^s, \sum_{c \in a} r(i, c)\}$ and $p(a, i) = (Q_i^s)^2 - (\max\{0, Q_i^s - \sum_{c \in a} r(i, c)\})^2$ respectively, where Q_i^s is the queue size for user i before scheduling, and $r(i, c)$ is the data rate for user i, RB c at the time slot. These three objective functions combine throughput maximization and queue stability together.

B. LTE UL FDPS

We consider a general FDPS problem for the LTE uplink. We are given an uplink system with m RBs and n users. In one time slot, for each set of contiguous RBs $a \in A$ and each user i, we have a profit $p(a, i)$. Our goal is to schedule the system for this time slot. In other words, we intend to find the most advantageous way to assign an $a \in A$ to user i so that the total profit is maximized. The LTE UL FDPS problem is formalized as the following combinatorial optimization problem.

$$\max \sum_{(a,i) \in A \times N} p(a, i) \cdot x_i^a$$

$$\begin{aligned}
&\text{subject to:} \\
&\text{for each RB } c \in M: && \sum_{i \in N, a : c \in a} x_i^a \leq 1 \\
&\text{for each user } i \in N: && \sum_{a \in A} x_i^a \leq 1 \\
&\text{for } i \in N, a \in A: && x_i^a \in \{0, 1\}
\end{aligned} \tag{3}$$

The first constraint shows that every RB is assigned to at most one user, and the second constraint ensures that each user can get no more than one set of contiguous RBs. In fact, LTE UL FDPS aims to find a subset of $A \times N$ which maximizes its total profit in a time slot, according to the scheduling policy specified in the profit function. Problem (3) is a binary integer programming and it is not hard to find that the PF-FDPS problem studied in [7] is a special case of (3).

IV. Hardness Results

A. Hardness of (3)

It is not difficult to show that LTE UL FDPS is NP-hard. [7] has shown that the LTE UL PF-FDPS problem, a special case of LTE UL FDPS, is NP-hard. Furthermore, it is straightforward to reduce the LTE UL PF-FDPS problem to LTE UL FDPS (by simply setting $p(a,i) = \sum_{c \in a} \lambda_i^c$). Thus, LTE UL FDPS is NP-hard.

Furthermore, we have a stronger result here.

Theorem 1: LTE UL FDPS is MAX SNP-hard, and hence it does not have a PTAS assuming $P \neq NP$.

Proof: We prove this theorem by presenting an L-reduction from JISP2 to LTE UL FDPS.

Assume that X is an instance of JISP2. X has n jobs, and we denote them by $J_1, J_2, \cdots, J_n$. Job J_i has two intervals, namely $[s_i^{(1)}, f_i^{(1)})$ and $[s_i^{(2)}, f_i^{(2)})$. $s_i^{(j)}$ and $f_i^{(j)}$ are integers, and $f_i^{(j)} > s_i^{(j)}$, $j = 1, 2$, $i = 1, 2, \cdots, n$.

Now we construct function R. $R(X)$ is defined as follows. Let $m = \max\limits_{i=1,2,j=1,\cdots n} \{f_j^{(i)}\} - 1$. $R(X)$ is an LTE uplink system which has n users and m RBs. $N = \{1, 2, \cdots n\}$ is the set of active users. A is the set of all contiguous RBs such that $\forall a \in A$, $a = \{i, i+1, \cdots, i+l\}, 1 \leq i \leq i+l \leq m$. User i corresponds to job J_i, $i = 1, 2, \cdots, n$, and the profit function $p : A \times N \rightarrow \mathbb{R}^{\geq 0}$ is defined as follows.

$$p(a,i) = \begin{cases} 1 & \text{if } a = \{s_i^{(j)}, s_i^{(j)}+1, \cdots, f_i^{(j)}-1\}, \\ & j = 1, 2, \\ 0 & \text{otherwise.} \end{cases}$$

Since $s_i^{(j)}$ and $f_i^{(j)}$ are integers, and $f_i^{(j)} > s_i^{(j)}$, $j = 1, 2$, $i = 1, 2, \cdots, n$, the profit function p is well defined.

Next we construct function S. Assume that s is a feasible solution of $R(X)$. s can be written as $\{(a_i, u_i) \mid i = 1, 2, \cdots, k, a_i \in A, u_i \in N\}$. We define $p_{supp} = \{(a,i) \mid p(a,i) = 1, (a,i) \in A \times N\}$. Let $s' = s \cap p_{supp}$. Obviously s' is also a feasible solution of $R(X)$ such that each element of s' has a positive profit. That is, in s', each user at most selects one set of contiguous RBs and no two a_is appearing in s' intersect.

According to the definition of the profit function p, $\forall (a_i, u_i) \in s'$, job J_{u_i} is associated with an interval $[head(a_i), tail(a_i)+1)$ in X. Therefore $S(s)$ is defined as

$$S(s) = \{([head(a_i), tail(a_i)+1), J_{u_i}) \mid (a_i, u_i) \in s'\}$$

We use the pair $([head(a_i), tail(a_i)+1), J_{u_i})$ to denote that in $S(s)$, job J_{u_i} selects the interval $[head(a_i), tail(a_i)+1)$. Since s' is a feasible solution of $R(X)$, correspondingly in $S(s)$, each job at most selects one interval, and no two selected intervals intersect. So $S(s)$ is a feasible solution of X. Thus function S is well defined.

Now we check (1). Assume that s_0 is the optimal solution of $R(X)$, and define $s_0' = s_0 \cap s_{supp}$. Apparently, we have

$$VAL(s_0') = VAL(s) = OPT(R(X))$$

We know that $S(s_0)$ is a feasible solution of X, and we have

$$VAL(s_0') = VAL(S(s_0)) \leq OPT(X)$$

Thus, we have $OPT(R(X)) \leq OPT(X)$. So (1) holds ($\alpha = 1$).

Then we check (2). Let s be any feasible solution of $R(X)$. Obviously we have

$$VAL(s) = VAL(s') = VAL(S(s)) \tag{4}$$

where $s' = s \cap s_{supp}$. Assume that r is the optimal solution of X, and $r = \{([s_{u_i}, f_{u_i}), J_{u_i}), i = 1, 2, \cdots k\}$. In $R(X)$, we can correspondingly construct $s = \{(\{s_{u_i}, s_{u_i}+1, \cdots, f_{u_i}-1\}, u_i), i = 1, 2, \cdots, k\}$. In s, each user is assigned to at most one set of contiguous RBs, and any two sets of contiguous RBs do not intersect. So s is a feasible solution of $R(X)$. Moreover, since for $R(X)$,

$$p(\{s_{u_i}, s_{u_i}+1, \cdots, f_{u_i}-1\}, u_i) = 1, i = 1, 2 \cdots, k,$$

we have

$$OPT(R(X)) \geq VAL(s) = VAL(r) = OPT(X) \tag{5}$$

.

Combining (4) and (5), we have $OPT(X) - VAL(S(s)) \leq OPT(R(X)) - VAL(s)$. So (2) holds ($\beta = 1$).

Thus, (R, S) is an L-reduction from JISP2 to LTE UL FDPS. Since JISP2 is MAX SNP-hard, LTE UL FDPS is also MAX SNP-hard and it does not have a PTAS unless $P = NP$. ■

Theorem 1 is somewhat devastating, since the nonexistence of PTAS implies that for some constant $\delta > 0$, there are no polynomial time $(1+\delta)$-approximation algorithms for LTE UL FDPS unless $P = NP$. That is to say, we could at most hope for approximation algorithms which have constant approximation ratios[2].

B. The Size of Search Space of (3)

Despite the fact that the LTE UL FDPS problem is MAX SNP-hard, one may still be tempted to find the optimal solution by an exhaustive search, since this approach does not consume much computation power when the search space is small, and enumerating all feasible schedules is sufficient to find the optimal schedule.

However, a further analysis shows that an exhaustive search for the optimal schedule is not practical for real systems. In the following, we calculate the number of feasible schedules

[2]For PTAS, we can have a $(1+\epsilon)$-approximation algorithm computable in polynomial time for any $\epsilon > 0$.

and estimate the running time of the exhaustive search for an uplink system which has m RBs and n active users.

Assume that in a feasible schedule, k out of n users are assigned to contiguous RBs. These k sets of contiguous RBs are denoted as $a_1, \cdots, a_k$ such that $1 \le head(a_1) \le tail(a_1) < head(a_2) \le tail(a_2) \cdots < head(a_k) \le tail(a_k) \le m$. So $1 \le head(a_1) < tail(a_1)+1 < head(a_2)+1 < tail(a_2)+2 \cdots < head(a_k)+k-1 < tail(a_k)+k \le m+k$. Thus, there is a 1-1 correspondence between the number of choices of k sets of contiguous RBs and the number of $2k$ integers $\{b_i, i = 1, 2, \cdots, 2k\}$ such that $1 \le b_1 < \cdots < b_{2k} \le m+k$. So the number of choices of k sets of contiguous RBs is $\binom{m+k}{2k}$. After assigning k users to k sets of contiguous RBs, we have $\binom{m+k}{2k} \cdot \frac{n!}{(n-k)!}$ feasible schedules in which k active users are assigned to contiguous RBs. So the total number of feasible schedules is

$$\sum_{k=0}^{n} \binom{m+k}{2k} \cdot \frac{n!}{(n-k)!} > \binom{m+n}{2n} \cdot n! \tag{6}$$

In practical systems, 3GPP LTE Release 8 specifies that $6 \le m \le 110$ [16]. The set of allowed values for m is given as $\{6, 15, 25, 50, 75, 100\}$ [17]. On the other hand, we may have at least several tens of active users in a cell. If $m = 25, n = 15$, then the right side of (6) $\approx 1.1 \times 10^{21}$. Assume that checking one feasible schedule takes 1×10^{-9}s. The running time of an exhaustive search ($> 1 \times 10^{12}$s$\gg$TTI=1×10^{-3}s) is entirely unacceptable.

Thus, the hardness result and the giant search space of (3) imply that approximation algorithms computable in polynomial time with guaranteed performance is indispensable for the LTE UL FDPS problem.

In the two subsequent sections, we present two polynomial time approximation algorithms respectively. The first algorithm is intuitive and easy to follow, but its performance declines slightly as the number of active users grows. The second one is relatively delicate, but achieves a better performance. The second approximation algorithm is derived from the local ratio technique [12].

V. A Greedy Strategy Based Algorithm

Here we present our first approximation algorithm. The main idea of the heuristic algorithm is to divide the LTE UL FDPS problem into several subproblems according to the profit, and then apply a greedy method to each subproblem. We prove that this algorithm has an approximation ratio of $O(\ln n)$, where n is the number of active users in a cell.

A. A Special Scheduling Problem

We start from considering a special scheduling problem.

$$\begin{aligned}
&\max \sum_{(a,i)\in\mathcal{C}\subseteq A\times N} x_i^a \\
&\text{subject to:} \\
&\text{for each RB } c \in M: \quad \sum_{(a,i)|(a,i)\in\mathcal{C}, c\in a} x_i^a \le 1 \\
&\text{for each user } i \in N: \quad \sum_{(a,i)\in\mathcal{C}} x_i^a \le 1 \\
&\text{for each } (a,i) \in \mathcal{A}: \quad x_i^a \in \{0,1\}
\end{aligned} \tag{7}$$

In this problem, a user i may not choose a set of contiguous RBs arbitrarily, but from $\mathcal{C}$, a collection of legitimate sets of (a,i) pairs. All pairs of one user and one of his legitimate sets of contiguous RBs constitute the set $\mathcal{C}$, which is a subset of $A \times N$. In addition, $p(a,i) = 1, \forall (a,i) \in \mathcal{C}$. That is, the goal of this problem is to schedule as many users as possible in a time slot. Problem (7) also complies with the constraints on users and RBs of the LTE UL FDPS problem.

JISPk problem studied in [15] is very similar to (7). Based on [15], we provide a greedy algorithm for (7). $head(a)$ and $tail(a)$ are the smallest and the largest RBs in $a \in A$ respectively.

Algorithm 1 GREEDY

1: **input** $\mathcal{C}$
2: $G \leftarrow \emptyset$, $C \leftarrow \mathcal{C}$
3: **while** $C \ne \emptyset$ **do**
4: $\quad (a^*, i^*) \leftarrow \arg\min_{(a,i)\in C}(tail(a))$ // break ties arbitrary
5: $\quad C \leftarrow C \backslash \{(a,i) \mid i = i^*, \text{ or } head(a) \le tail(a^*)\}$
6: $\quad G \leftarrow G \cup \{(a^*, i^*)\}$
7: **end while**
8: **return** G

The main idea of Algorithm 1 is straightforward. It takes as an input $\mathcal{C}$, a collection of admissible pairs of one user and one set of contiguous RBs. The algorithm iteratively selects an (a,i) which has the smallest $tail(a)$ until C is empty. Algorithm 1 outputs G, a feasible schedule of (7). The value of Algorithm 1 is $|G|$. Using the technique of [15], we can prove the following theorem.

Theorem 2: Algorithm 1 is a 2-approximation algorithm for (7). Moreover, it can be shown that the approximation ratio of 2 is tight.

B. The Weighted Version of (7)

Then we consider the weighted version of (7). We associate each pair $(a,i) \in \mathcal{C}$ with profit $p(a,i)$, and change the objective function to

$$\max \sum_{(a,i)\in\mathcal{C}} p(a,i) \cdot x_i^a \tag{8}$$

The constraints of (8) are the same as those of (7). This problem is analogous to LTE UL FDPS except that a feasible schedule is selected from a subset $\mathcal{C} \subseteq A\times N$, not from $A\times N$ itself.

We consider the following procedure which provides a feasible solution for (8). For a subset $\mathcal{C} \subseteq A \times N$, we ignore the profit function and run Algorithm 1, and then get a feasible schedule G. Then we compute $WG = \sum_{(a,i)\in G} p(a,i)$. We denote the optimal value of (8) by $W\text{-}OPT(\mathcal{C})$. The following lemma establishes the relation between WG and $W\text{-}OPT(\mathcal{C})$.

Lemma 4: If $\forall(a,i) \in \mathcal{C}$, $0 < m_p \leq p(a,i) \leq M_p$, then $W\text{-}OPT(\mathcal{C}) \leq \frac{2M_p}{m_p} \cdot WG$.

Proof: If $\mathcal{C} = \emptyset$, then the lemma holds vacuously. Otherwise, we denote the optimal value of (7) by $S\text{-}OPT(\mathcal{C})$, then

$$W\text{-}OPT(\mathcal{C}) \leq M_p \cdot S\text{-}OPT(\mathcal{C}).$$

According to Theorem 2, we have $S\text{-}OPT(\mathcal{C}) \leq 2 \cdot |G|$, so

$$W\text{-}OPT(\mathcal{C}) \leq 2M_p \cdot |G|. \tag{9}$$

According to the proposed procedure, we have

$$WG \geq m_p \cdot |G|. \tag{10}$$

Combining (9) and (10), we finally get

$$W\text{-}OPT(\mathcal{C}) \leq \frac{2M_p}{m_p} \cdot WG.$$

■

C. The Approximation Algorithm

Taking Algorithm 1 as the subroutine, our first approximation algorithm for the LTE UL FDPS problem is stated as follows (Algorithm 2).

Algorithm 2 takes m, n, p as the input, where m is the number of RBs, n is the number of active users, and $p : A \times N \rightarrow \mathbb{R}^{\geq 0}$. It first partitions all (a,i) pairs into $k+1$ subsets $\{S_j, j = 0, 1, \cdots, k\}$ according to their profits. For $S_k, k \geq 1$, Algorithm 2 invokes Algorithm 1 to obtain a feasible schedule. Thus Algorithm 2 obtains k feasible schedules, then it chooses the schedule which has the largest total profit as the output.

Algorithm 2 GREEDY-BASED (G-B for short)

1: **input** m, n, p
2: $p_{max} \leftarrow \max\limits_{(a,i)\in A\times N} p(a,i)$
3: partition $A \times N$ into $k+1$ subsets: $S_0, S_1, \cdots, S_k$ such that $S_0 = \{(a,i) \mid p(a,i) \leq \frac{p_{max}}{n}\}$ and $S_j = \{(a,i) \mid \frac{\alpha^{j-1}\cdot p_{max}}{n} < p(a,i) \leq \frac{\alpha^j \cdot p_{max}}{n}\}$ for $j \geq 1$
4: // $\alpha > 1$ is a constant and $k = \lceil \frac{\ln n}{\ln \alpha} \rceil$. The specification of α will be discussed in the proof of Theorem 3
5: **for** $j = 1$ to k **do**
6: $\quad G_j \leftarrow GREEDY(S_j)$
7: $\quad WG_j \leftarrow \sum\limits_{(a,i)\in G_j} p(a,i)$
8: **end for**
9: $j^* \leftarrow \arg\max\limits_{1\leq j\leq k}(WG_j)$
10: **return** G_{j^*} and WG_{j^*}

Theorem 3: Algorithm 2 is an $O(\ln n)$-approximation algorithm for LTE UL FDPS, where n is the number of active users in a cell.

Proof: For any instance $\mathcal{S}$ of (3), we denote the optimal value by $OPT(\mathcal{S})$ and the return value of Algorithm 2 by WG_{j^*}. First we show that $\frac{OPT(\mathcal{S})}{WG_{j^*}} \leq \alpha + \frac{2\alpha}{\ln \alpha} \ln n$.

In Algorithm 2, all (a,i) pairs are partitioned into $k+1$ subsets $\{S_j, j = 0, 1, \cdots, k\}$. Each S_j with corresponding profits can be regarded as an instance of (8), whose optimal value is written as $W\text{-}OPT(S_j)$. Then it is obvious that

$$OPT(\mathcal{S}) \leq \sum_{j=0}^{k} W\text{-}OPT(S_j). \tag{11}$$

For $j \geq 1$, $\frac{\alpha^{j-1}\cdot p_{max}}{n} < p(a,i) \leq \frac{\alpha^j\cdot p_{max}}{n}, \forall(a,i) \in S_j$. So Lemma 4 indicates that

$$W\text{-}OPT(S_j) \leq 2\alpha \cdot WG_j, j \geq 1.$$

For $j = 0$, since the number of users is n, the second constraint of (3) tells us that

$$W\text{-}OPT(S_0) \leq n \cdot \frac{p_{max}}{n} = p_{max}$$

Combining the above two equations, we turn (11) into

$$OPT(\mathcal{S}) \leq p_{max} + 2\alpha \sum_{j=1}^{k} WG_j$$

Note that $\{(a,i) \mid p(a,i) = p_{max}\} \subseteq S_k$, so $S_k \neq \emptyset$. Thus, $WG_k \geq \frac{\alpha^{k-1}\cdot p_{max}}{n}$. So we have

$$OPT(\mathcal{S}) \leq \frac{n}{\alpha^{k-1}} WG_k + 2\alpha \sum_{j=1}^{k} WG_j$$

Since $WG_{j^*} = \max_{1\leq j\leq k} WG_j$, we get

$$OPT(\mathcal{S}) \leq (\frac{n}{\alpha^{k-1}} + 2\alpha\cdot k)\cdot WG_{j^*} \leq (\alpha + \frac{2\alpha}{\ln \alpha}\ln n)\cdot WG_{j^*}$$

Thus,

$$\frac{OPT(\mathcal{S})}{WG_{j^*}} \leq \alpha + \frac{2\alpha}{\ln \alpha} \ln n \tag{12}$$

Furthermore, we can specify α in terms of n so that the right side of (12) is minimized. Let

$$\frac{\partial \left(\alpha + \frac{2\alpha}{\ln \alpha}\ln n\right)}{\partial \alpha} = 0$$

then we get $1 - \frac{2n}{(\ln \alpha)^2} + \frac{2n}{\ln \alpha} = 0$. Since $\alpha > 1$, we have $\alpha = \exp\left(\frac{2n}{n+\sqrt{2n+n^2}}\right)$. Substituting this expression of α into (12), we finally get

$$\frac{OPT(\mathcal{S})}{WG_{j^*}} \leq \exp\left(\frac{2\ln n}{\ln n + \sqrt{\ln n(2+\ln n)}}\right)\cdot \left(1 + \ln n + \sqrt{\ln n(2+\ln n)}\right) \leq 2e\cdot(\ln n + 1).$$

Since $\mathcal{S}$ is any instance of (3), we conclude that Algorithm 2 has an approximation ratio of $O(\ln n)$. ■

Figure 2 depicts the approximation ratio of Algorithm 2 for different numbers of active users. A cell typically has several tens of active users. In Figure 2, we can find that the approximation ratio of Algorithm 2 increases slowly as the number of active users n grows.

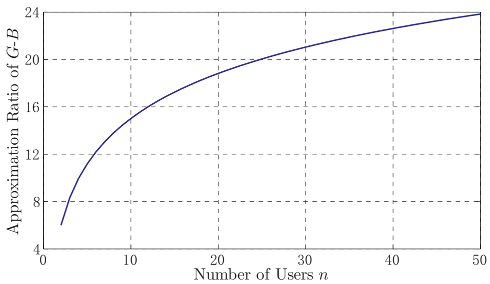

Fig. 2. Approximation ratio of Algorithm 2.

D. A Word on Complexity

Algorithm 2 is based on an intuitive idea which utilizes a greedy method. We have shown that the gap between the optimal value and the value returned by the algorithm widens gradually as the number of active users n increases. Here we show that Algorithm 2 has a small time complexity.

We denote by $\mathcal{N}$ the size of $A \times N$ and $\mathcal{N} = n \cdot (\binom{m}{2} + m)$. We use $\mathcal{N}_j$ to represent the size of S_j, $j \geq 1$. Note that $\sum_{j \geq 1}^{k} \mathcal{N}_j \leq \mathcal{N}$.

Algorithm 2 first finds p_{max} and partitions $A \times N$ into $S_0, S_1, \cdots, S_k$, which requires a traversal of all elements of $A \times N$. This takes $O(\mathcal{N})$ time.

For each $S_j, j \geq 1$, Algorithm 2 calls the subroutine Algorithm 1. Algorithm 1 iteratively seeks an available (a, i) with smallest $tail(a)$, which usually requires a sort for $(a, i) \in S_j$ based on $tail(a)$. Thus, it can be shown that for S_j, invoking Algorithm 1 takes $O(\mathcal{N}_j \ln \mathcal{N}_j)$ time. Moreover

$$\sum_{j=1}^{k} O(\mathcal{N}_j \ln \mathcal{N}_j) = \sum_{j=1}^{k} O(\mathcal{N}_j \ln \mathcal{N}) = O(\mathcal{N} \ln \mathcal{N}).$$

So the total running time of Algorithm 2 can be calculated as

$$\begin{aligned} T_{G\text{-}B} &= O(\mathcal{N}) + \sum_{j=1}^{k} O(\mathcal{N}_j \ln \mathcal{N}_j) = O(\mathcal{N} \ln \mathcal{N}) \\ &= O\left(n \cdot m^2 \ln(n \cdot m^2)\right) = O(n \cdot m^2 (\ln n + \ln m))\right. . \end{aligned} \tag{13}$$

VI. A Local Ratio Technique Based Algorithm

The basic idea of Algorithm 2 is intuitive, however the performance is not so satisfying when the number of active users grows (Figure 2). In this section, we introduce a more delicate approximation algorithm based on the local ratio technique [12], which achieves a constant approximation ratio of 2.

A. The Algorithm

The approximation algorithm is listed as Algorithm 3. It takes the number of RBs (m), the number of active users (n), and the profit function ($p : A \times N \rightarrow \mathbb{R}^{\geq 0}$) as the input. It outputs a feasible schedule S^* and its total profit W^*. Algorithm 3 first iterates from 1 through m to find candidate (a, i)s for S^*. In each loop, the algorithm tries to find the best candidate (a, i) (in the meaning of profit), and uses a stack S to store it. After the iteration, Algorithm 3 pops each (a, i) from S and adds (a, i) to S^* if it is valid. Finally, the algorithm generates a feasible schedule S^*.

Algorithm 3 A Local Ratio Technique Based Algorithm (L-R for short)

input m, n, p
$p' \leftarrow p$, $S \leftarrow \emptyset$ // S is a stack
for $j = 1$ to m **do**
  $(a^*, i^*) \leftarrow \underset{(a,i) \in \{(a,i) | tail(a) = j\}}{\arg\max} (p'(a, i))$ // break ties arbitrary
  if $p'(a^*, i^*) \leq 0$ **then**
    continue
  end if
  $S.push((a^*, i^*))$
  for each (a, i) such that $i = i^*$ or a intersects a^* **do**
    if $p'(a, i) > 0$ **then**
      $p'(a, i) \leftarrow p'(a, i) - p'(a^*, i^*)$
    end if
  end for
end for
$S' \leftarrow \emptyset$
while $S \neq \emptyset$ **do**
  $(a, i) \leftarrow S.pop()$
  if $S' \cup \{(a, i)\}$ is a valid schedule **then**
    // a valid schedule means that $S' \cup \{(a, i)\}$ should meet the constraints of (3)
    $S' \leftarrow S' \cup \{(a, i)\}$
  end if
end while
$S^* \leftarrow S'$, $W^* \leftarrow \sum_{(a,i) \in S^*} p(a, i)$
return S^* and W^*

B. The Approximation Ratio

In this subsection, we will prove that Algorithm 3 has an approximation ratio of 2, and that this approximation ratio is tight. To obtain this result, we first introduce Lemma 5, which is an instance of the Local Ratio Theorem [12]. Then we prove that Algorithm 3 is a 2-approximation algorithm for LTE UL FDPS. Finally, we use an example to show that the approximation ratio of 2 is tight.

Lemma 5: Consider an uplink system which has m RBs and n active users. Let p, p_1, p_2 be profit functions such that $p(a, i) = p_1(a, i) + p_2(a, i), \forall (a, i) \in A \times N$. Let S^*, S_1^* and S_2^* be optimal schedules for p, p_2 and p_3 respectively. Assume that S is a feasible schedule such that $r \cdot \sum_{(a,i)\in S} p_1(a,i) \geq \sum_{(a,i)\in S_1^*} p_1(a,i)$ and $r \cdot \sum_{(a,i)\in S} p_2(a,i) \geq \sum_{(a,i)\in S_2^*} p_2(a,i)$, where r is a constant. Then we have

$$r \cdot \sum_{(a,i)\in S} p(a,i) \geq \sum_{(a,i)\in S^*} p(a,i).$$

Proof: Note that

$$\begin{aligned}\sum_{(a,i)\in S^*} p(a,i) &= \sum_{(a,i)\in S^*} p_1(a,i) + \sum_{(a,i)\in S^*} p_2(a,i)\\ &\leq \sum_{(a,i)\in S_1^*} p_1(a,i) + \sum_{(a,i)\in S_2^*} p_2(a,i).\end{aligned}$$

On the other hand,

$$\begin{aligned}&\sum_{(a,i)\in S_1^*} p_1(a,i) + \sum_{(a,i)\in S_2^*} p_2(a,i)\\ \leq& r \cdot \sum_{(a,i)\in S} p_1(a,i) + r \cdot \sum_{(a,i)\in S} p_2(a,i) = r \cdot \sum_{(a,i)\in S} p(a,i).\end{aligned}$$

Thus the lemma is proved. ■

Note that in Lemma 5, the profit functions p, p_1 and p_2 need not be nonnegative functions. That is to say, the profit function can take a negative value for some $(a,i) \in A \times N$. In Theorem 4, we will prove that Algorithm 3 has a constant approximation ratio of 2. The proof of this result relies on the observation that Lemma 5 applies to Algorithm 3 and we will use Lemma 5 inductively in the proof.

Theorem 4: Algorithm 3 is a 2-approximation algorithm for LTE UL FDPS.

Proof: We first introduce some necessary notations and definitions.

After the outermost for-loop is finished, the stack S can be represented as $\cup_{j=1}^{m} S_j$ where each $|S_j| \leq 1$. If in the jth for-loop, the algorithm adds some (a_j^*, i_j^*) to S, then $S_j = \{(a_j^*, i_j^*)\}$. Otherwise, $S_j = \emptyset$.

Correspondingly, the outermost for-loop iteratively generates a series of profit functions $(p_1^{(j)}, p_2^{(j)}), j = 1, \cdots, m$. For $j \geq 1$, if $S_j = \{(a_j^*, i_j^*)\}$,

$$p_1^{(j)}(a,i) = \begin{cases} p_2^{(j-1)}(a_j^*, i_j^*) \cdot 1_{\mathbb{R}^{>0}}(p_2^{(j-1)}(a,i)) & i = i_j^* \text{ or } a \text{ intersects } a_j^*, \\ 0 & \text{otherwise}\end{cases}$$

and

$$p_2^{(j)}(a,i) = p_2^{(j-1)}(a,i) - p_1^{(j)}(a,i) \quad \forall (a,i) \in A \times N.$$

$1_{\mathbb{R}^{>0}}(x)$ is the characteristic function such that $1_{\mathbb{R}^{>0}}(x) = 1$ for $x > 0$, and $1_{\mathbb{R}^{>0}}(x) = 0$ for $x \leq 0$.

If $S_j = \emptyset$, we set $p_1^{(j)}(a,i) = 0$ and $p_2^{(j)}(a,i) = p_2^{(j-1)}(a,i)$.

We let $p_2^{(0)}(a,i) = p(a,i)$ and $p_1^{(0)}(a,i) = 0, \forall (a,i) \in A \times N$. Thus, we have $p_1^{(j)}(a,i) + p_2^{(j)}(a,i) = p_2^{(j-1)}(a,i), j \geq 1, (a,i) \in A \times N$. According to the algorithm, it can be shown that for $j \geq 1$,

$$p_2^{(j)}(a,i) \leq 0, \forall tail(a) \leq j,$$

and that

$$\forall (a,i) \in A \times N, p_2^{(j)}(a,i) \geq p_2^{(k)}(a,i), j \leq k.$$

In addition, the while-loop equivalently generates a series of $S_j^*, j = 0, \cdots, m$, where $S_0^* = S^*$, $S_m^* = \emptyset$ and $S_j^* \subseteq \cup_{k=m}^{j+1} S_k$. S_j^* is regarded as the value of variable S' after the algorithm tries to add $\cup_{k=m}^{j+1} S_k$ to S'. So it is obvious that

$$S_{j+1}^* \subseteq S_j^* \subseteq S_{j+1}^* \cup S_{j+1}, j = 0, \cdots, m-1.$$

We denote by $W_{opt}^{(j)}$ the total profit of the optimal schedule for $p_2^{(j)}$. If the optimal schedule is empty, we set $W_{opt}^{(j)} = 0$. In particular, $W_{opt}^{(0)}$ is the optimal value of (3). We define $W^{(j)} = \sum_{(a,i)\in S_j^*} p_2^{(j)}(a,i)$. If $S_j^* = \emptyset$, $W^{(j)} = 0$. In particular, $W^{(0)} = W^*$. So what we are going to prove is $W_{opt}^{(0)} \leq 2W^{(0)} = 2W^*$.

In the following, we will prove by induction that $W_{opt}^{(j)} \leq 2W^{(j)}$, $j = m, m-1, \cdots, 1, 0$. When $j = 0$, we obtain the desired result. The mathematical induction starts from m down to 0.

The Basis: When $j = m$, $p^{(m)}(a,i) \leq 0$, $tail(a) \leq m$. That is, $p^{(m)}(a,i) \leq 0, \forall (a,i) \in A \times N$. So $W_{opt}^{(m)} = 0$. Since $S_m^* = \emptyset$, $W_{opt}^{(m)} \leq 2W^{(m)}$ holds vacuously.

The Inductive Step: Assume that $W_{opt}^{(j)} \leq 2W^{(j)}$, $j \leq m$. we denote by $V_{opt}^{(j)}$ the total profit of the optimal schedule for $p_1^{(j)}$.

If $S_j = \emptyset$, then $p_2^{(j)}(a,i) = p_2^{(j-1)}(a,i)$, and $p_1^{(j)}(a,i) = 0$. So $W_{opt}^{(j-1)} = W_{opt}^{(j)}$. Since $S_j^* \subseteq S_{j-1}^* \subseteq S_j^* \cup S_j$, $S_{j-1}^* = S_j^*$. Thus, $W^{(j-1)} = \sum_{(a,i)\in S_{j-1}^*} p_2^{(j-1)}(a,i) = \sum_{(a,i)\in S_j^*} p_2^{(j)}(a,i) = W^{(j)}$. Then we get

$$W_{opt}^{(j-1)} = W_{opt}^{(j)} \leq 2W^{(j)} = 2W^{(j-1)}.$$

Otherwise, $S_j = \{(a_j^*, i_j^*)\}$. According to the algorithm, $p_2^{(j)}(a_j^*, i_j^*) = 0$. Because $S_j^* \subseteq S_{j-1}^* \subseteq S_j^* \cup S_j$,

$$\begin{aligned}2 \sum_{(a,i)\in S_{j-1}^*} p_2^{(j)}(a,i) &= 2 \sum_{(a,i)\in S_j^*} p_2^{(j)}(a,i)\\ &= 2W^{(j)} \geq W_{opt}^{(j)}.\end{aligned} \tag{14}$$

On the other hand, S_{j-1}^* contains at least one element (a', i') such that $i' = i_j^*$ or a' intersects a. According to the algorithm, $\{(a', i')\} = S_k, k \geq j$. Thus $p_2^{(j-1)}(a', i') \geq p_2^{(k-1)}(a', i') > 0$. So $p_1^{(j)}(a', i') = p_2^{(j-1)}(a_j^*, i_j^*)$. In this case,

$$\sum_{(a,i)\in S_{j-1}^*} p_1^{(j)}(a,i) \geq p_1^{(j)}(a', i') = p_2^{(j-1)}(a_j^*, i_j^*).$$

Then we derive an upper bound for $V_{opt}^{(j)}$. We define $Supp_{p_1} = \{(a,i) \mid p_1^{(j)}(a,i) > 0\}$. $Supp_{p_1}$ contains (a,i)s such that $i = i_j^*$ or a intersects a_j^*. Moreover, $p_2^{(j-1)}(a,i) \leq 0, tail(a) \leq j-1$, so for $(a,i) \in Supp_{p_1}$, $tail(a) \geq j$. Since $tail(a_j^*) = j$, $(a,i) \in Supp_{p_1}$ such that $i \neq i_j^*$ must intersects a_j^* at RB j. So the optimal solution for $p_1^{(j)}$ at most have two (a,i)s $\in Supp_{p_1}$: for one (a,i), $i = i_j^*$, and for another (a,i), a intersects a_j^*. That is, $V_{opt}^{(j)} \leq 2p_2^{(j-1)}(a_j^*, i_j^*)$. Thus,

$$2 \sum_{(a,i) \in S_{j-1}^*} p_1^{(j)}(a,i) \geq 2p_2^{(j-1)}(a_j^*, i_j^*) \geq V_{opt}^{(j)} \tag{15}$$

Since $p_2^{(j-1)}(a,i) = p_1^{(j)}(a,i) + p_2^{(j)}(a,i)$, according to (14) and (15), Lemma 5 indicates that $2W^{(j-1)} = 2\sum_{(a,i)\in S_{j-1}^*} p_2^{(j-1)}(a,i) \geq W_{opt}^{(j-1)}$.

Since both the basis and the inductive step have been proved, it has now been proved by mathematical induction that $W_{opt}^{(0)} \leq 2W^*$. So Algorithm 3 has an approximation ratio of 2. ■

Moreover, we find an LTE UL FDPS problem which shows that the approximation ratio of Algorithm 3 is tight.

Proposition 1: The approximation ratio of 2 is tight.

Proof: In the LTE UL FDPS problem of Figure 3, $m = n = 2$, and $0 < \epsilon < 1$. The optimal solution is $\{(\{2\},1), (\{1\},2)\}$, and the optimal value is $2-\epsilon$. However, for this instance of LTE UL FDPS, Algorithm 3 returns $S^* = \{(\{1\},1)\}$ and $W^* = 1$. So the approximation ratio $\geq \frac{2-\epsilon}{1}, \forall 0 < \epsilon < 1$. When $\epsilon \to 0$, $2-\epsilon \to 2$. Thus, the approximation ratio is tight.

(a,i)	$(\{1\},1)$	$(\{2\},1)$	$(\{1,2\},1)$
$p(a,i)$	1	1	1
(a,i)	$(\{1\},2)$	$(\{2\},2)$	$(\{1,2\},2)$
$p(a,i)$	$1-\epsilon$	0	1

Fig. 3. An LTE UL FDPS problem which approaches the approximation ratio.

■

C. A Word on Complexity

We have shown that Algorithm 3 has a constant approximation ratio, and here we analyze the complexity of the algorithm. In the jth for loop, the number of (a,i)s such that $tail(a) = j$ is $m \cdot j$. So to obtain (a_j^*, i_j^*), the algorithm at least accesses $m \cdot j$ (a,i)s. In addition, since in the jth for loop, $p'(a,i) \leq 0$, $tail(a) \leq j-1$, the number of (a,i)s such that $p'(a,i)$ is changed is at least $\binom{m}{2} + m - \frac{j(j-1)}{2} + (n-1)\cdot j \cdot (m-j+1)$. So the outermost for loop takes

$$\sum_{j=1}^{m} \left(m \cdot j + \binom{m}{2} + m + (n-1)\cdot j \cdot (m-j+1) \right.$$
$$\left. - \frac{j(j-1)}{2} \right) = \frac{1}{6}\left(-m + m^3 + 5mn + 6m^2 n + m^3 n\right)$$
$$= O(m^3 n)$$

running time. Since S contains at most m elements, the while loop takes $O(m)$ time. Thus, the running time of Algorithm 3 is

$$T_{L\text{-}R} = O(m^3 n). \tag{16}$$

VII. Conclusion

In this paper, we consider a general FDPS problem for the LTE uplink. Our formulation of this problem applies to many scheduling policies. We prove that the LTE UL FDPS problem is MAX SNP-hard, which implies that the problem is hard to approximate. We propose two approximation algorithms for this scheduling problem, both of which are computable in polynomial time. The first algorithm is based on a simple greedy method, and it has an approximation ratio of $O(\ln n)$, where n is the number of active users in the cell. The second algorithm is more delicate, and it achieves a constant approximation ratio of 2.

References

[1] "Physical layer aspects for evolved Universal Terrestrial Radio Access (UTRA) (Release 7)," 3GPP TR 25.814 V7.1.0, Tech. Rep., 2006.

[2] S.-B. Lee, S. Choudhury, A. Khoshnevis, S. Xu, and S. Lu, "Downlink MIMO with Frequency-Domain Packet Scheduling for 3GPP LTE," in *IEEE INFOCOM* , 2009.

[3] J. Zyren, "Overview of the 3GPP Long Term Evolution Physical Layer," Freescale Semiconductor, Tech. Rep., 2007.

[4] "LTE - an introduction," Ericsson, Tech. Rep., 2009.

[5] M. Rumney, "3GPP LTE: Introducing SIngle-Carrier FDMA," *Agilent Measurement Journal*, vol. 4, pp. 18–27, 2008.

[6] H. G. Myung, J. Lim, and D. J. Goodman, "Single carrier FDMA for uplink wireless transmission," *Vehicular Technology Magazine, IEEE*, vol. 1, no. 3, pp. 30–38, 2006.

[7] S.-B. Lee, I. Pefkianakis, A. Meyerson, S. Xu, and S. Lu, "Proportional Fair Frequency-Domain Packet Scheduling for 3GPP LTE Uplink," in *INFOCOM Mini-Conference*, 2009.

[8] J. Lim, H. Myung, K. Oh, and D. Goodman, "Proportional Fair Scheduling of Uplink Single-Carrier FDMA Systems," in *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, 2006, pp. 1–6.

[9] H. Myung, K. Oh, J. Lim, and D. Goodman, "Channel-Dependent Scheduling of an Uplink SC-FDMA System with Imperfect Channel Information," in *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, 31 2008-April 3 2008, pp. 1860–1864.

[10] M. Andrews and L. Zhang, "Scheduling algorithms for multi-carrier wireless data systems," in *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2007, pp. 3 – 14.

[11] M. Andrews, "Instability of the proportional fair scheduling algorithm for HDR," *Wireless Communications, IEEE Transactions on*, vol. 3, no. 5, pp. 1422–1426, 2004.

[12] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. S. Naor, and B. Schieber, "A unified approach to approximating resource allocation and scheduling," *J. ACM*, vol. 48, no. 5, pp. 1069 – 1090, 2001.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, 2001.

[14] C. H. Papadimitriou, *Computational Complexity*. Addison Wesley, 1993.

[15] F. C. R. Spieksma, "Approximating an Interval Scheduling Problem," in *APPROX '98: Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization*. London, UK: Springer-Verlag, 1998, pp. 169 – 180.

[16] "Physical Channels and Modulation (Release 8)," 3GPP TS 36.211 V8.7.0, Tech. Rep., 2009.

[17] "User Equipment (UE) radio transmission and reception (Release 9)," 3GPP TS 36.101 v9.0.0, Tech. Rep., 2009.