# Joint Scheduling and Fast Cell Selection in OFDMA Wireless Networks

Reuven Cohen Guy Grebla
Department of Computer Science
Technion—Israel Institute of Technology
Haifa 32000, Israel

***Abstract*—In modern broadband cellular networks, the omni-directional antenna at each cell is replaced by 3 or 6 directional antennas, one in every sector. While every sector can run its own scheduling algorithm, bandwidth utilization can be significantly increased if a joint scheduler makes these decisions for all the sectors. This gives rise to a new problem, referred to as "joint scheduling," addressed in this paper for the first time. The problem is proven to be NP-hard, but we propose efficient algorithms with a worst-case performance guarantee for solving it. We then show that the proposed algorithms indeed substantially increase the network throughput.**

***Index Terms*—Cellular networks, 4G mobile communication, Optimal scheduling.**

## I. Introduction

A crucial step in the evolution of broadband cellular networks is reducing the size of the cells and increasing their number, in order to address the fast growing demand for bandwidth. The major expenditure in the deployment of a wireless network is installing BSs (Base Stations) and connecting them to the backbone. Thus, it is important to increase the number of cells without the concomitant cost associated with the deployment of many new BSs. This goal can be attained in one of the following two ways, or by a combination thereof.

(a) Using cell sectorization: the omni-directional antenna at each BS is replaced by 3 antennas of 120 degrees, or 6 antennas of 60 degrees, all operated by the same BS.

(b) Using relay nodes: such relay nodes are governed by low-cost BSs that have only wireless connectivity to the backbone through their "parent" (regular) BS.

In this paper we study the first approach. A cell is divided into multiple sectors, each is served by a directional antenna, and all the antennas are governed by the same BS (Fig. 1). The BS receives all downlink packets destined for users associated with any of the cell sectors. We define the new OFDMA (Orthogonal Frequency Division Multiple Access) scheduling problem encountered by a BS in the proposed architecture as "OFDMA joint scheduling," because a single entity (the BS) needs to make scheduling decisions for multiple transmitting sectors/antennas[1]. This is a new OFDMA scheduling problem, defined and solved for the first time in this paper.

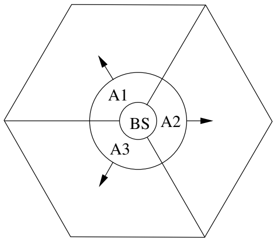


Fig. 1. A cell of a cellular network, divided into three sectors using antennas $A1$, $A2$, $A3$.

In addition to defining and solving the new OFDMA joint scheduling problem, we build a detailed simulation model, study the performance gain of joint scheduling, and compare between the various algorithms presented in the paper. We show that our new joint scheduling algorithms significantly increase the throughput of an OFDMA network.

In contrast to a "regular" scheduling algorithm, which only needs to decide which packet should be transmitted in the next OFDMA 1ms subframe, the joint scheduling algorithm also needs to determine which antenna is the best for serving each packet. This will not necessarily be the one with which the target user has the best SINR (Signal-to-Interference-plus-Noise Ratio). For example, if the user has the best SINR with antenna $A1$ but reasonable SINR with $A2$, and the sector of $A1$ is more heavily loaded than $A2$, then a global optimum is likely to be obtained by scheduling the transmission of this packet using the OFDMA resources of $A2$ rather than those of $A1$.

We show that the joint scheduling problem is equivalent to the known NP-hard problem called GAP (Gen-

[1] Throughout the paper we use the words *antenna* and *sector* interchangeably.

eralized Assignment Problem) if the scheduler does not have to choose an MCS (Modulation and Coding Scheme) for each packet. However, to improve the performance of joint scheduling, we generalize it to also select the most appropriate MCS for each packet. In this case **we get a new theoretical NP-hard problem**, which combines two known NP-hard problems: GAP [13] and MCKP (Multiple Choice Knapsack Problem) [23]. In addition to formulating this problem for the first time, we also develop an efficient approximation algorithm with a proven lower bound performance guarantee.

The fact that the scheduler determines the transmitting sector for each user can be viewed as an implementation of a concept sometimes known as "Fast Cell Selection." While this concept is currently not standardized by LTE (Long Term Evolution), we believe that the results of this paper can play an important role in the integration of joint scheduling and fast cell selection into LTE.

The rest of the paper is organized as follows. In Section II we discuss related work. In Section III we present our OFDMA joint scheduling network model. In Section IV we define the joint scheduling problem and show its equivalence to the NP-hard GAP problem. In Section V we extend the joint scheduling problem to allow dynamic MCS selection of each packet. This results in a new NP-hard problem to which we present a new approximation. Section VI presents an extensive simulation study and Section VII concludes the paper.

## II. Related Work

To the best of our knowledge, this paper is the first to define a packet-level joint scheduling scheme for an OFDMA wireless network. Related work can be divided into: (a) papers on the problem of deciding which BS should transmit to which user; (b) papers on the relationship between wireless scheduling and GAP; and (c) other relevant papers.

Papers from the first group include [3], [9], [14], [17], [27], [30], [31], [34]. They all address the problem of deciding which BS should transmit to which user. We refer to this problem as user-level fast cell selection, which is different from our packet-level fast sector selection, where two packets destined for the same user can be transmitted using different sectors.

In [3], the authors formalize the cell selection problem as an optimization problem and show that the problem is NP-hard. They propose approximation algorithms for special cases of this problem and compare them to a greedy algorithm that selects for every user device the BS with which it has the highest SINR. There are several important differences between [3] and our work. First, the algorithms in [3] are for the user-level and are therefore more appropriate for admission control. In contrast, our algorithms are for packet-level, and are therefore appropriate for a real-time scheduler that needs to make packet-level decisions once every 1ms subframe. Second, we allow different MCSs to be used for every packet, while in [3] only one MCS is considered. Finally, in [3] the profit associated with a [user, BS] pair is fixed, while in our paper it is dynamically determined (in Section IV we show concrete examples for dynamically determined profit values).

In [9], two basic cell selection schemes are considered and a new handover decision algorithm for improving cell edge throughput is proposed. In contrast to our scheme, the algorithm in [9] aims at improving cell edge throughput, while we optimize overall network performance.

In [14], the advantage of fast cell selection in HSDPA (High Speed Downlink Packet Access) networks is investigated. Such a selection scheme is proposed and evaluated. Like the scheme in [3], the proposed scheme is a user-level admission control scheme and not a packet-level scheduling scheme.

In [17], a joint scheduling scheme for joint processing fast cell selection is proposed and evaluated. The scheme applies muting to the strongest neighbor cell for decreasing interference to cell edge users. The scheme improves cell edge user throughput and cell average user throughput, but overall optimization is not considered. In addition, this is not a packet-level scheduling scheme, because it allocates the scheduling blocks on a per user basis.

In [34], an adaptive resource allocation scheme is proposed for OFDM networks. The proposed scheme involves cell selection and adaptive modulation. Unlike in our work, cell selection decisions and adaptive modulation decisions are made separately. In addition, packet-level optimization is not performed. The target of the scheme in [34] is to maximize the overall throughput. Thus, different QoS for different users is not supported.

Papers from the second group, which deal with the relationship between wireless scheduling and GAP, are [4], [20], [29]. In [4], the scheduling problem in MIMO wireless networks is formulated as a GAP problem, and a general solution that uses adaptive proportional fair scheduling is proposed. In [20], the multi-carrier proportional fair scheduling problem is shown to be equivalent to GAP when each user always has data to transmit. In [29], the authors address the problem of providing minimum rate guarantees to different service classes in an OFDMA network.

Papers from the third group are [5], [12], [16], [18], [22], [30], [32]. In both [5] and [32], the access point association problem is addressed. Fast cell selection is not used and a user receives all of its packets from a single access point with which it is associated.

In [16], the cell selection problem for femtocell networks is studied. A learning algorithm is presented, which solves the problem while taking into account the condition of the channel. However, packet scheduling is not addressed. The work in [15] proposes an algorithm for user-level cell selection. The proposed algorithm is based on the received power of the reference signal.

In [18], several downlink scheduling schemes combined with fast cell selection are proposed for WCDMA. Our work differs from [18] mainly in that our algorithms are for OFDMA networks. Other differences are that in [18] (a) cell selection and MCS selection are performed separately; (b) the scheduling is user- and not packet-level; (c) a fast Rayleigh fading channel is assumed; (d) different QoS for different users is not considered; and (e) throughput and fairness are improved but not overall network performance.

In [30], a new cell selection strategy is proposed. In this scheme a node is more likely to select a low power relay node as its serving station in order to reduce the interference caused by this transmission. The proposed scheme is suitable for networks with low power nodes. This scheme does not schedule the transmissions and its main goal is to improve spectral efficiency.

While much work has been done on scheduling in wireless networks, only a few papers address resource allocation in OFDMA networks [12], [22]. In contrast to our paper, these papers do not consider joint scheduling. In [12], the authors formulate the OFDMA scheduling problem in the context of WiMax, and propose efficient algorithms for solving it. The BS determines which packets will be transmitted in each OFDMA frame, using which MCS, and how the OFDMA frame matrix will be constructed. This paper is probably the first to propose to model the MCS selection as an instance of MCKP. When the BS needs to make scheduling decisions for multiple consecutive frames rather than for each frame separately, the packet selection problem is also shown to be similar to GAP[2].

## III. Frequency Reuse Model

In general, algorithms for joint scheduling depend to a large extent on the network model, and in particular on the frequency reuse model employed by the network. In order to make our contribution more concrete, we present our algorithms in the context of the FFR (Fractional Frequency Reuse) model [25], [28], which is the most common frequency reuse model in wireless networks. However, the algorithms are applicable even if another frequency reuse model is used, including SFR (Soft Frequency Reuse) [25] and reuse-1.

Throughout the paper we consider a cell with 3 sectors. However, all our results are applicable to cells with 6 or any other number of sectors. In Fig. 1 we showed a division of an OFDMA cell into 3 sectors. Fig. 2 shows a schematic description of such a division, but this time with the implementation of FFR. Bandwidth is partitioned into $N+1$ subbands: F0, F1, F2 and F3 ($N=3$ in the figure). Subband F0 is used by all three sectors at the same time (reuse-1) and is intended for users who can get a relatively good SINR from this band despite interference from neighboring sectors. Subband F1 is used only by sector 1. Therefore, users receiving their transmission in this subband will not suffer from interference due to neighboring sectors. Similarly, subband F2 is used only by sector 2 and subband F3 only by sector 3. Thus, the reuse factor of F1, F2 and F3 is $1/3$.

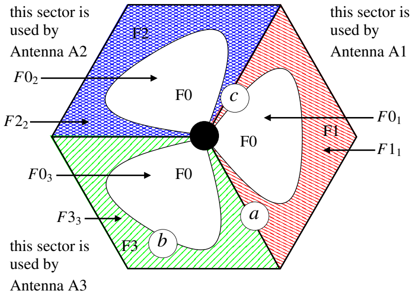

this sector is used by Antenna A2
F2
$F0_2$
F0
c
$F2_2$
this sector is used by Antenna A1
$F0_1$
F0
F1
$F1_1$
$F0_3$
F0
a
$F3_3$
F3
b
this sector is used by Antenna A3

Fig. 2. A cell with 3 sectors and 3 users

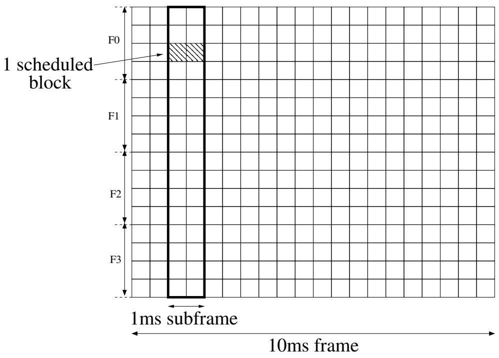


Fig. 3. An abstract structure of the LTE frame and subframe

As an example for a typical frame in an OFDMA network, Fig. 3 shows a schematic structure of an LTE 10 ms frame[3]. The frame is divided into 10 1ms subframes, and the scheduler needs to make a scheduling decision for each. The frame can be logically viewed as divided into the 4 subbands mentioned above. Each subband consists of several scheduled blocks[4]. The total number

[2] This is shown in Lemma 2.

[3] We are trying to abstract the problem in the most generic way. Therefore, we skip some of the LTE physical layer details that are not directly relevant to the description of the problem and algorithms.

[4] A scheduled block is the minimum allocation unit. Its size is equal to $12 \cdot 14 = 168$ OFDMA symbols. The bit capacity of a symbol depends on the MCS of the packet; e.g., with a modulation of 16-QAM and a coding rate of $3/4$, each symbol accommodates $4 \cdot 3/4 = 3$ bits.

of scheduled blocks in a subframe depends on the system capacity; it is 100 in a 20MHz system, for example.

Throughout the paper, each reuse-1/3 or reuse-1 area that corresponds to a sector within a cell, is referred to as **a scheduling area.** In our case, there are 6 such areas: F$0_1$, F$0_2$, F$0_3$, F$1_1$, F$2_2$, and F$3_3$ (see Fig. 2), where F$i_j$ indicates that this scheduling area is in the F$i$ bands and the transmitting antenna is A$j$. Before the transmission of every subframe, the joint scheduler needs to decide how to fill up the 6 scheduling areas. Its output is 3 subframes: one for transmission by antenna A1 in sector 1, for which F$0_1$ and F$1_1$ are used; one for transmission by antenna A2 in sector 2, for which F$0_2$ and F$2_2$ are used; and one for transmission by antenna A3 in sector 3, for which F$0_3$ and F$3_3$ are used.

As in standard wireless networks, we assume that the BS receives periodic CSIs (Channel State Indicators) [10] from the users. Using these reports, the BS is able to predict the SINR for the transmission to the user in each scheduling area. The difference in the SINR of different scheduled blocks in the same scheduling area is negligible compared to the difference in the SINR of different scheduled blocks in different scheduling areas. Thus, it is usually ignored by the BS. This allows the BS to get a single CSI value from each user for each scheduling area. Nevertheless, all the algorithms proposed in this paper can also be used when more CSI values are reported, by considering each set of subbands for which a CSI value is reported as a different scheduling area.

The joint scheduler not only needs to determine which packet will be sent by which antenna and in what scheduling area, but also what MCS should be used for each packet. The number of scheduled blocks required for such a transmission is calculated from the selected MCS, the length of the packet, and the length (number of OFDMA symbols) of a scheduled block [2]. By selecting the appropriate MCS for every packet, the scheduler can significantly increase bandwidth utilization. For example, suppose that the transmission of a certain packet requires 1.3 scheduled blocks using the default MCS. In such a case, the scheduler must allocate 2 scheduled blocks because only integral numbers of blocks can be allocated to each packet. Now, suppose that the scheduler is given the option to use other MCSs for this packet. Specifically, it can choose a more efficient but less robust MCS, which requires only 0.9 scheduled blocks and reduces the probability for successful transmission from 0.97 to 0.9. By choosing this MCS, the scheduler reduces the transmission cost of this packet by 50%, because only 1 scheduled block is needed rather than 2. This is accomplished with a success probability reduction of only 7.22% (from 0.97 to 0.9).

## IV. The OFDMA Joint Scheduling Problem

In this section we define and study the basic problem of OFDMA joint scheduling, where we assume that each packet can be transmitted in every scheduling area using at most one MCS. This default MCS is chosen in the following way:

- If the SINR enables the user to receive a packet with a probability not smaller than $1-\epsilon$, then the MCS that consumes minimum bandwidth and guarantees this probability is chosen.
- Else, the most robust MCS (which guarantees the highest success probability) is chosen.

The value of $\epsilon$ may vary from one packet to another depending on the application QoS requirements. In our scheduling model, we assume that the transmission of a packet in each scheduling area is associated with a profit that depends on the following parameters (see [11] for more details): (a) the importance of this packet for the sending application; (b) the importance of transmitting the packet in this subframe, rather than in a future one; and (c) the probability that this packet will be successfully received by the user.

We now give examples of concrete profit values whose aim is to optimize either the throughput, energy, delay, or fairness.

$\boldsymbol{p}_{\textbf{packets}}$ - This profit value is defined as the packet transmission success probability. As a result, the sum of all profit values equals the expected number of successfully received packets, i.e., packet-level throughput.

$\boldsymbol{p}_{\textbf{throughput}}$ - This profit value is defined as $p_{\text{packets}}$ multiplied by the length of the packet. As a result, the sum of all profit values of all transmitted packets equals the expected number of successfully received bits, i.e., bit-level throughput.

$\boldsymbol{p}_{\textbf{energy}}$ - This profit value is defined as $p_{\text{throughput}}$ divided by the transmission energy cost. As a result, the sum of all profit values of all packets transmitted equals the expected number of bits transmitted per energy unit, namely, the transmission energy utilization.

$\boldsymbol{p}_{\textbf{delay}}$ - For each packet, if there is "enough time" (i.e., more than a given threshold $\Delta$) until the packet must be transmitted in order to meet its deadline, the profit value is defined as $p_{\text{throughput}}$. But if the packet must be transmitted soon, its profit is set to a large value, in order to increase the likelihood that it will be transmitted on time.

$\boldsymbol{p}_{\textbf{pf}}$ - For each user, the most urgent packet destined for this user is assigned a profit value of $\log(p_{\text{throughput}})$. The profit for all remaining packets is set to zero. It is shown in [24] that an allocation that maximizes $\sum \log R_u$, where $R_u$ is the rate of user $u$, is proportional fair. As a result, a proportional fair allocation is one that maximizes $\sum p_{\text{pf}}$.

The success probability for transmitting a given packet varies from one scheduling area to another. Thus, the profit of a packet might also dynamically change.

As an example, consider Fig. 2 with three users: $a$, $b$ and $c$. Suppose that:

• packet$_1$ of user $a$ can be transmitted either in the reuse-1/3 area of sector 1 (F$1_1$) or in the reuse-1/3 area of sector 3 (F$3_3$). Suppose that in the former case, the default MCS that guarantees the $1-\epsilon$ success probability is 16-QAM with a coding rate of $1/2$, which is translated into 0.9 scheduled blocks; i.e., 0.9 scheduled blocks are required in order to transmit all the bits of this packet using 16-QAM with a coding rate of $1/2$. Since allocation is possible using only integral numbers of scheduled blocks, 1 scheduled block is actually needed. Suppose that in the case where the packet is transmitted by sector 3 in F$3_3$, the default MCS that guarantees the $1-\epsilon$ success probability is QPSK with a coding rate of $2/3$, which is translated into $\lceil 1.35 \rceil = 2$ scheduled blocks.

• packet$_2$ of user $b$ can be transmitted either in the reuse-1/3 area of sector 3 (F$3_3$) using [64-QAM, $5/6$], or in the reuse-1 area of sector 3 (F$0_3$) using [16-QAM, $3/4$].

• packet$_3$ of user $c$ can be transmitted in the reuse-1/3 area of sector 1 (F$1_1$) using [64-QAM, $5/6$], or in the reuse-1/3 area of sector 2 (F$2_2$) using [16-QAM, $2/3$], or in the reuse-1 area of sector 1 (F$0_1$) using [16-QAM, $3/4$].

Based on the input above, and on the input regarding other waiting packets of all users, the scheduler should determine which packet will be transmitted in each scheduling area (F$0_1$, F$0_2$, F$0_3$, F$1_1$, F$2_2$ or F$3_3$) during the next OFDMA subframe. The decision regarding the MCS to be used for every packet is a consequence of the selected scheduling area. To this end, the scheduler needs to solve the following problem :

**Problem 1 (OFDMA Joint Scheduling)**

**Instance:** The scheduler is given a set of scheduling areas for OFDMA joint scheduling, and the number of scheduled blocks to be allocated in each. The scheduler is also given a set of packets that are awaiting transmission in the next subframe. Each packet can have an arbitrary length. For each packet$_i$, the scheduler is given a set of feasible scheduling areas for which the packet's receiver has sufficiently good SINR (see Definition 1 below). From this information, the scheduler determines the default MCS and the success probability for transmitting the packet in each scheduling area. Then, the scheduler determines the number of scheduled blocks required for transmitting the packet in each scheduling area, i.e., the transmission cost, and the profit for each transmission. All this information is considered as input for the OFDMA joint scheduling problem.

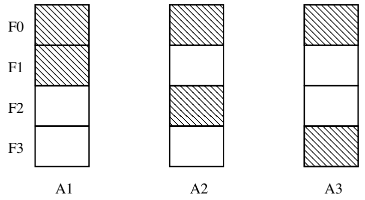


Fig. 4. The OFDMA subframes of a cell transmitted in the 3 sectors by antenna A1, A2 and A3

**Objective:** Find a feasible schedule that maximizes the total profit for the next subframe. A feasible schedule is a mapping between waiting packets and scheduling areas such that: (a) at most one scheduling area is chosen for each packet; (b) the number of scheduled blocks available in every scheduling area is not exceeded; and (c) for each user no two packets are scheduled to be transmitted by different antennas in the same subbands at the same time.

In Lemma 2, this problem is shown to be equivalent to GAP, for which a formal (mathematical) formulation is given. ■

*Definition 1:* An SINR value is said to be *"sufficiently good"* if it is greater than 1.

The value of 1 is chosen because the transmission success probability for SINR $\leq$ 1 is very small [6]. Therefore, the scheduler is configured to use only transmissions whose SINR is not too small.

To understand restriction (c) of Problem 1, consider Fig. 4. This figure shows the OFDMA subframe transmitted by each antenna when a cell is divided into 3 sectors. Recall that subband $F0$ is the one used for the reuse-1 scheduling area of each sector. Therefore, it is occupied by all 3 sectors. If the scheduler decides that A1 and A2 have to transmit to the same user using subband $F0$, i.e., one packet is scheduled in $F0_1$ (the reuse-1 scheduling area of sector A1) and another in $F0_2$ (the reuse-1 scheduling area of sector A2), the user will be able to decode at most one of these packets. We avoid such a collision using restriction (c). Note, however, that restriction (c) does not apply for two packets destined for *different* users. For example, antennas A1 and A2 can be used for transmitting packets to *different* users in subband $F0$ at the same time. This makes sense if the first user is in the middle of sector A1, and the second user is in the middle of sector A2.

*Lemma 1:* The set of feasible scheduling areas for each packet contains at most one reuse-1 scheduling area.

*Proof:* Let $p(Ai)$ be the power received by a user from the transmission of antenna $Ai$ in reuse-1 area $F0_i$ for $i \in \{1,2,3\}$. The SINR of a user for a transmission of antenna $Ai$ in $F0_i$ is $\frac{p(Ai)}{p_I(Ai)+n_0 w}$, where $p_I(Ai)$ is the interference due to transmissions in $F0_j$ for $j \neq i$ and

$n_0$ is the thermal noise over the bandwidth $w$. Suppose that the set of feasible scheduling areas for a given user contains the reuse-1 area $F0_1$. Thus, the SINR for the transmission of $A1$ is bigger than 1 and therefore $p(A1) > p_I(A1) \geq p(A2) + p(A3)$. This implies that the SINR for the transmission of $A2$ and $A3$ is not "sufficiently good" (Definition 1), and therefore it is not possible for the transmission of $A2$ or $A3$ to have a good SINR. ■

*Corollary 1:* Restriction (c) is always met.

*Lemma 2:* Under the considered FFR model, Problem 1 is equivalent to GAP. Thus, (a) the problem is NP-hard; (b) any $\alpha$-approximation algorithm for the Knapsack problem can be transformed into a $(1+\alpha)$-approximation[5] algorithm for Problem 1.

*Proof:* GAP is defined as follows [13]. The instance is a pair $[B, I]$ and a 2D profit matrix $P$, where $B$ is a set of bins (knapsacks), $I$ is a set of items, and $P$ is a $|I| \times |B|$ matrix that indicates the profit and size for each item in each bin. The objective is to find a subset $U \subseteq S$ of items that has a feasible packing in $B$, such that the profit is maximized. A feasible packing is a mapping of each item to at most one bin such that the capacity of each bin is not exceeded.

Mathematically, GAP can be formulated as:

$$\text{maximize: } \sum_{i=1}^{|I|} \sum_{j=1}^{|B|} p_{ij} x_{ij}$$

subject to:

$$\sum_{i=1}^{|I|} s_{ij} x_{ij} \leq B_j \text{ for } 1 \leq j \leq |B|, \tag{1}$$

$$\sum_{j=1}^{|B|} x_{ij} \leq 1 \text{ for } 1 \leq i \leq |I|, \tag{2}$$

$$\text{and } x_{ij} \in \{0, 1\} \text{ for } 1 \leq i \leq |I|, 1 \leq j \leq |B|. \tag{3}$$

In the above formulation, $p_{ij}$ is the profit obtained from packing item $i$ in bin $j$, $s_{ij}$ is the size of item $i$ for bin $j$, $B_j$ is the capacity of bin $j$, and $x_{ij}$ is a binary variable that indicates whether or not item $i$ is chosen for bin $j$. Eq. (1) ensures that the capacity is not exceeded in each bin $j$. Eq. (2) ensures that each item is packed in at most one bin. Eq. (3) prevents the solution from packing fractions of items.

We first show how to transform an instance of GAP into an instance of Problem 1 in polynomial time. Without loss of generality, we assume that the bin sizes are of the same size $S$. Every bin is transformed into a reuse-$(1/|B|)$ scheduling area with $S$ scheduled blocks. Every GAP item is transformed into a waiting packet whose size and profit for each scheduling area are equal to the size and profit of the GAP's item in the corresponding bin. Note that condition (c) of Problem 1 holds for the constructed instance.

Next, we present a polynomial time transformation of a Problem 1 instance into a GAP instance. Every scheduling area is considered as a GAP bin whose size is equal to the number of scheduled blocks in that area. Every packet is transformed into a GAP item. For a given scheduling area, the size and profit are determined according to the default MCS and the target success probability of the packet.

In [13] it is shown that GAP is NP-hard and that any $\alpha$-approximation algorithm for the Knapsack problem can be transformed into a $(1+\alpha)$-approximation algorithm for GAP. ■

Knapsack is one of the most studied problems in combinatorial optimization [23]. Although it is NP-hard, it has many efficient algorithms. From Lemma 2 it follows that the well-known polynomial time greedy 2-approximation for Knapsack can be transformed into a 3-approximation algorithm for Problem 1. The algorithm for Knapsack described in [26] will be transformed into a $(2+\epsilon)$-approximation algorithm that runs in poly$(n, 1/\epsilon)$ time where $n$ is the total input length.

## V. OFDMA Joint Scheduling With Dynamic MCS Selection

In the previous section we assumed that a packet is transmitted using a default MCS based on the target success probability. The performance of the joint scheduler can be improved if it is permitted to choose the MCS for every packet in every scheduling area instead. When the scheduler chooses a more efficient but less robust MCS for a packet, it reduces the cost of the assignment but also reduces the profit, because the profit is proportional to the transmission success probability.

As an example, suppose that there are 2 scheduling areas: SA$_1$, which contains 3 scheduled blocks, and SA$_2$, which contains 1 scheduled block. Suppose there are two waiting packets whose scheduling parameters are identical in both scheduling areas, and are shown in Table I. If every packet can only be transmitted using its default MCS [QPSK, 1/2], then only one packet can be accommodated in the next subframe. The extension proposed in this section allows the joint scheduler to choose [16-QAM, 3/4] for packet$_1$ and to schedule both packets: one in SA$_1$ and one in SA$_2$. The new problem is called "OFDMA Joint Scheduling with Dynamic MCS Selection," and is formally defined as follows.

*Definition 2:* A transmission instance is a combination of a scheduling area and an MCS as determined by the scheduler for a given waiting packet. ■

[5] Let $p_{\text{opt}}$ be the total profit of the optimal solution and $\alpha \geq 1$. An $\alpha$-approximation returns a solution whose profit is at least $\frac{p_{\text{opt}}}{\alpha}$.

|  | QPSK 1/2 |  | 16-QAM 3/4 |  |
|---|---|---|---|---|
|  | Length | Success prob. | Length | Success prob. |
| packet$_1$ | 3 | $(1-\epsilon)$ | 1 | 0.5 |
| packet$_2$ | 3 | $(1-2\epsilon)$ | 1 | 0.4 |

TABLE I
AN EXAMPLE OF THE ADVANTAGE OF JOINT SCHEDULING AND MCS SELECTION

**Problem 2 (OFDMA Joint Scheduling with Dynamic MCS Selection)**

**Instance:** Same as Problem 1, except that for each packet$_i$, we are not given a set of feasible scheduling areas but a set of feasible transmission instances, packet$_i^1$, packet$_i^2$ $\cdots$ packet$_i^M$. Each such set may contain transmission instances from the same scheduling area but with different MCSs.

**Objective:** Find a feasible schedule that maximizes the total profit for the next subframe. A feasible schedule is a mapping between the waiting packets and their transmission instances, such that: (a) at most one scheduling area is chosen for each packet; (b) the number of scheduled blocks available in every scheduling area is not exceeded; and (c) for each user no two packets are scheduled to be transmitted by different antennas in the same subbands at the same time. ■

To solve Problem 2, we define a new general theoretical problem, which extends GAP to allow multiple choices from each item. The new problem is called MC-GAP (Multiple Choice GAP), and is defined as follows.

**Problem 3 (MC-GAP)**

**Instance:** A triplet $(B, I, C)$ and a 3D profit matrix $P$, where $B$ is a set of bins (knapsacks), $I$ is a set of items, $C$ is a set of configurations, and $P$ is a $|I| \times |C| \times |B|$ matrix that indicates the profit and size for each item in each bin using each configuration.

**Objective:** Find a subset $U \subseteq (I \times C)$ of [item, configuration] pairs that has a feasible packing in $B$, such that each item is packed at most once, using one of its configurations, and the profit is maximized. ■

Mathematically, MC-GAP can be formulated as:

$$\text{maximize: } \sum_{i=1}^{|I|}\sum_{c=1}^{|C|}\sum_{j=1}^{|B|} p_{icj}x_{icj}$$

subject to:

$$\sum_{i=1}^{|I|}\sum_{c=1}^{|C|} s_{icj}x_{icj} \le B_j \text{ for } 1 \le j \le |B|, \tag{4}$$

$$\sum_{c=1}^{|C|}\sum_{j=1}^{|B|} x_{icj} \le 1 \text{ for } 1 \le i \le |I|, \tag{5}$$

$$\text{and } x_{icj} \in \{0,1\} \text{ for } 1 \le i \le |I|, 1 \le c \le |C|, \quad 1 \le j \le |B|. \tag{6}$$

In the above formulation, $p_{icj}$ is the profit obtained from packing item $i$ using configuration $c$ in bin $j$, $s_{icj}$ is the size of item $i$ using configuration $c$ for bin $j$, $B_j$ is the capacity of bin $j$, and $x_{icj}$ is a binary variable that indicates whether or not item $i$ is chosen for bin $j$ using configuration $c$. Eq. (4) ensures that the capacity is not exceeded in each bin $j$. Eq. (5) ensures that each item is packed using at most one configuration and in at most one bin. Eq. (6) prevents the solution from packing fractions of items.

*Lemma 3:* Problem 2 can be transformed into an instance of MC-GAP in linear time.

*Proof:* Every scheduling area of Problem 2 can be considered as a bin whose size is equal to the number of scheduled blocks in that area. Every packet is mapped to an MC-GAP item. Each MCS is an MC-GAP configuration. If the packet has a transmission instance for a given scheduling area and a given MCS, the size and the profit are determined according to this instance. ■

MC-GAP is a combination of two known NP-hard problems: GAP and MCKP (Multiple Choice Knapsack Problem). In MCKP there is only one knapsack, i.e., only one scheduling area, whereas in GAP there is only one choice (one MCS) for selecting an item (a packet) into a knapsack (scheduling area). Although MCKP is NP-hard [23], it has efficient approximations [7] and an optimal pseudo-polynomial time algorithm [23].

We now present an algorithm for solving MC-GAP. The algorithm extends the one presented in [13] for solving GAP. Using the local-ratio technique [8], our algorithm transforms any $\alpha$-approximation algorithm for MCKP into a $(1+\alpha)$-approximation algorithm for MC-GAP.

The local-ratio argument is as follows. Let $F$ be a set of constraints and let $p(), p_1(), p_2()$ be profit functions such that $p() = p_1() + p_2()$. Then, if $x$ is an $r$-approximate solution with respect to $(F, p_1())$ and with respect to $(F, p_2())$, it is also an $r$-approximate solution with respect to $(F, p())$. The proof is very simple [8]. Let $x^*$, $x_1^*$ and $x_2^*$ be optimal solutions for $(F, p())$, $(F, p_1())$, and $(F, p_2())$ respectively. Then $p(x) = p_1(x) + p_2(x) \ge r \cdot p_1(x_1^*) + r \cdot p_2(x_2^*) \ge r \cdot (p_1(x^*) + p_2(x^*)) = r \cdot p(x^*)$.

To apply the local-ratio argument, our algorithm splits the profit matrix $p$ into two profit matrices, $p_1$ and $p_2$, whose sum equals $p$. We start by describing the *profit-split* procedure, as demonstrated in Fig. 5 for item $i$. The input for the procedure is the profit matrix $p$ and a set $\overline{S}$ of [item, configuration] pairs. In $p_1$, the profit of $i$ in the first bin is not changed. For any other bin, if $i$ does not appear in $\overline{S}$, its profit in $p_1$ is set to 0 (Fig. 5(b)); otherwise, there is some configuration $\overline{c}$ for which $(i, \overline{c}) \in \overline{S}$, and the profit of $i$ in $p_1$ is set to

| | bin 1 | bin 2 | … | bin $\lvert B\rvert$ |
|---|---|---|---|---|
| configuration 1 | $p(i,1,1)$ | $p(i,1,2)$ | … | $p(i,1,\lvert B\rvert)$ |
| configuration 2 | $p(i,2,1)$ | $p(i,2,2)$ | … | $p(i,2,\lvert B\rvert)$ |
| ⋮ | ⋮ | ⋮ | … | ⋮ |
| configuration $\lvert C\rvert$ | $p(i,\lvert C\rvert,1)$ | $p(i,\lvert C\rvert,2)$ | … | $p(i,\lvert C\rvert,\lvert B\rvert)$ |

(a) Entries of item $i$ in the original profit matrix $p$

| | bin 1 | bin 2 | … | bin $\lvert B\rvert$ |
|---|---|---|---|---|
| configuration 1 | $p(i,1,1)$ | 0 | … | 0 |
| configuration 2 | $p(i,2,1)$ | 0 | … | 0 |
| ⋮ | ⋮ | ⋮ | … | ⋮ |
| configuration $\lvert C\rvert$ | $p(i,\lvert C\rvert,1)$ | 0 | … | 0 |

(b) Entries of item $i$ in $p_1$ for the case where $i$ is not in $\overline{S}$

| | bin 1 | bin 2 | … | bin $\lvert B\rvert$ |
|---|---|---|---|---|
| configuration 1 | $p(i,1,1)$ | $p(i,\overline{c},1)$ | … | $p(i,\overline{c},1)$ |
| configuration 2 | $p(i,2,1)$ | $p(i,\overline{c},1)$ | … | $p(i,\overline{c},1)$ |
| ⋮ | ⋮ | ⋮ | … | ⋮ |
| configuration $\overline{c}$ | $p(i,\overline{c},1)$ | $p(i,\overline{c},1)$ | $p(i,\overline{c},1)$ | $p(i,\overline{c},1)$ |
| ⋮ | ⋮ | ⋮ | … | ⋮ |
| configuration $\lvert C\rvert$ | $p(i,\lvert C\rvert,1)$ | $p(i,\overline{c},1)$ | $p(i,\overline{c},1)$ | $p(i,\overline{c},1)$ |

(c) Entries of item $i$ in $p_1$ for the case where $(i,\overline{c}) \in \overline{S}$

Fig. 5. Entries of item $i$ in the profit matrices used by our *profit-split* procedure

$p(i,\overline{c},1)$ (Fig. 5(c)). Matrix $p_2$ is defined as $p_2 = p - p_1$. The formal description of the procedure is as follows:

*Procedure profit-split(p, $\overline{S}$)*
Compute $p_1$ and $p_2$ using the following equations:

$$p_1[i,c,k] = \begin{cases} p[i,\overline{c},1] & \text{if } k \neq 1 \text{ and } \exists \overline{c} \text{ such} \\ & \text{that } (i,\overline{c}) \in \overline{S} \\ p[i,c,k] & \text{if } k = 1 \\ 0 & \text{Otherwise} \end{cases}$$

$$p_2 = p - p_1$$

An informal description of the algorithm is as follows. Let $\text{ALG}_{\text{MCKP}}$ be an $\alpha$-approximation algorithm for MCKP. Our algorithm first invokes $\text{ALG}_{\text{MCKP}}$ with respect to the first bin of MC-GAP. Let $\overline{S}$ be the output of $\text{ALG}_{\text{MCKP}}$. If there is only one bin, $\overline{S}$ is the final output of the algorithm. Otherwise, the algorithm invokes *profit-split(p, $\overline{S}$)* to obtain $p_1$ and $p_2$. The algorithm then ignores the first bin and continues recursively with $p_2$ as the new profit matrix. Let $S$ be the solution returned by the recursive call. For every $(i,c) \in \overline{S}$, if $i$ is not already in $S$, it is added. Finally, the algorithm returns $S$.

For a single bin, the returned solution of $\text{ALG}_{\text{MCKP}}$ is clearly a $(1+\alpha)$-approximation. If there are more bins, each time the algorithm returns from the recursive call and considers another bin, the obtained profit increases by some amount $X$, while the profit of the optimal solution increases by at most $(1+\alpha) \cdot X$. Therefore, the updated solution is also a $(1+\alpha)$-approximation.

We now give a formal description of the algorithm.

**Algorithm $\text{ALG}_{\text{MC-GAP}}$:**
Recall that B is the set of bins, $I$ is the set of items, $C$ is the set of configurations, and $p$ is a $|I| \times |C| \times |B|$ profit matrix. The value of $p[i,c,j]$ indicates the profit of item $i$ in bin $j$ using configuration $c$. We now construct from $\text{ALG}_{\text{MCKP}}$ a recursive algorithm for MC-GAP. Since our algorithm dynamically updates the profit function, we use $p_j$ to indicate the profit matrix at the beginning of the $j$th recursive call. Initially we set $p_1 \leftarrow p$, and we invoke the following Next-Bin procedure with $j = 1$:

Procedure Next-Bin(j)

1) Run $\text{ALG}_{\text{MCKP}}$ on bin $j$ using $p_j$ as the profit function. Let $\overline{S_j}$ be the set of selected [item, configuration] pairs returned by $\text{ALG}_{\text{MCKP}}$.
2) Decompose the profit function $p_j$ into two profit functions $p_j^1$ and $p_j^2$ by invoking profit-split($p_j$, $\overline{S_j}$).
3) If $j < |B|$ then
   - Set $p_{j+1} \leftarrow p_j^2$, and remove the column of bin $j$ from $p_{j+1}$.
   - Invoke Next-Bin($j+1$). Let $S_{j+1}$ be the returned assignment list.
   - Let $S_j$ be the same as $S_{j+1}$ except that for each item $i$, if $i$ is assigned in $\overline{S_j}$ for some $c$, $(i,c) \in \overline{S_j}$, and it is not assigned in $\cup_{k=j+1}^{|B|} S_k$, then the assignment of $(i,c)$ to bin $j$ is added to $S_j$.
   - Return $S_j$.

   Else, return $S_j = \overline{S_j}$. ■

*Theorem 1:* If $\text{ALG}_{\text{MCKP}}$ is an $\alpha$-approximation for MCKP, then $\text{ALG}_{\text{MC-GAP}}$ is a $(1+\alpha)$ approximation for MC-GAP.

*Proof:* We use the notation $p(S)$ to indicate the profit gained by assignment $S$. The proof is by induction on the number of bins available when the algorithm is invoked. For a single bin, $\overline{S_{|B|}}$ is an $\alpha$-approximation solution due to ALG$_{\text{MCKP}}$, and therefore it is a $(1+\alpha)$-approximation with respect to $p_{|B|}$. For the inductive step, assume that $S_{j+1}$ is a $(1+\alpha)$-approximation with respect to $p_{j+1}$. Matrix $p_j^2$ is identical to $p_{j+1}$ except that it contains a column with profit 0. Thus, $S_{j+1}$ is also an $(1+\alpha)$-approximation with respect to $p_j^2$. Since $S_j$ contains the items assigned by $S_{j+1}$, it is also a $(1+\alpha)$-approximation with respect to $p_j^2$.

Profit matrix $p_j^1$ has three components: (1) items in bin $j$, whose profit is the same as in $p_j$; (2) items not in bin $j$, which belong to $\overline{S_j}$; their profit in any configuration is identical to their profit in $\overline{S_j}$ (using the configuration specified in $\overline{S_j}$); (3) the remaining entries are all 0. Only components (1) and (2) of $p_j^1$ can contribute profit to an assignment. By the validity of ALG$_{\text{MCKP}}$, $\overline{S_j}$ is an $\alpha$-approximation with respect to component (1). Therefore, the best solution with respect to component (1) will gain a profit of at most $\alpha \cdot p_j^1(\overline{S_j})$. Moreover, the best solution with respect to component (2) will gain a profit of at most $p_j^1(\overline{S_j})$, since the profit of these items is the same regardless of where they are assigned and which configuration they use. This implies that $\overline{S_j}$ is a $(1+\alpha)$-approximation with respect to $p_j^1$. According to the last step of the algorithm, $p_j^1(S_j) = p_j^1(\overline{S_j})$ and $S_j$ is a $(1+\alpha)$-approximation with respect to both $p_j^1$ and $p_j^2$. Since $p_j = p_j^1 + p_j^2$, by the local-ratio argument, $S_j$ is also a $(1+\alpha)$-approximation with respect to $p_j$. ■

The lower bound proven in Theorem 1 is tight. Namely, there are instances of MC-GAP such that the profit returned by ALG$_{\text{MC-GAP}}$ equals $1/(1+\alpha)$ of the maximum profit. This is because instances of MC-GAP for which $|C| = 1$, i.e., there is only one configuration per item, are identical to instances of GAP. Furthermore, ALG$_{\text{MC-GAP}}$ on such instances is identical to the algorithm for GAP presented in [13]. Since in [13] it is shown that the approximation ratio of the algorithm for GAP is tight, the approximation ratio of ALG$_{\text{MC-GAP}}$ is also tight.

ALG$_{\text{MC-GAP}}$ can be implemented by an iterative algorithm whose running time is $O(|B| \cdot f(|I|, |C|) + |B| \cdot |I| \cdot |C|)$, where $f(|I|, |C|)$ is the running time of ALG$_{\text{MCKP}}$.

From Theorem 1 it follows that the performance of ALG$_{\text{MC-GAP}}$ depends on the performance of ALG$_{\text{MCKP}}$. The most efficient ALG$_{\text{MCKP}}$ is the algorithm described in [7]. This algorithm finds a $(1+\epsilon)$-approximate solution in $O(|I|^2 \cdot |C|/\epsilon)$ time. Thus, it can be transformed into a $(2+\epsilon)$-approximation algorithm for MC-GAP and Problem 2 whose running time is $O(|B| \cdot (|I|^2 \cdot |C|/\epsilon) + |B| \cdot |I| \cdot |C|)$. In [19], a $(5/4)$-approximation algorithm for MCKP whose running time is $O(|I| \cdot |C| \cdot \log |I|)$ is proposed. This algorithm can be transformed into a $(9/4)$-approximation algorithm for MC-GAP whose running time is $O(|B| \cdot (|I| \cdot |C| \log |I|) + |B| \cdot |I| \cdot |C|)$.

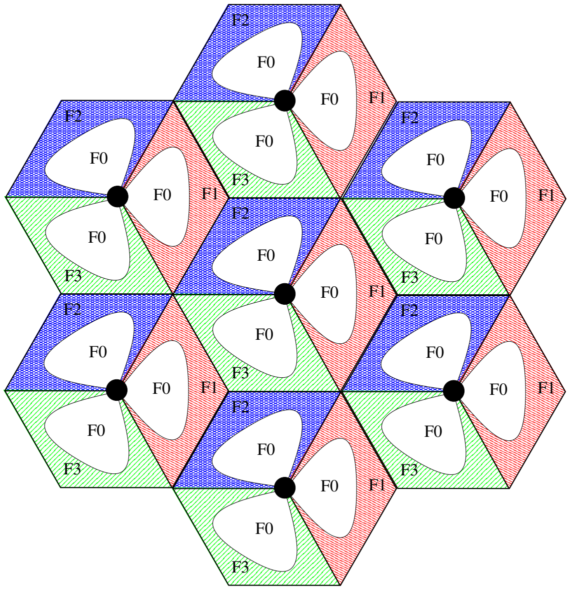


Fig. 6. Simulation network model

## VI. Simulation Study

In this section we present Monte-Carlo simulation results for the algorithms proposed in the paper. The purpose of this section is three-fold. First, we evaluate our approximation for the new MC-GAP problem by comparing its performance to that of an exponential-time optimal algorithm. Since the problem is NP-hard, this part of the study is conducted for small instances only. Second, we use the results of a water-filling algorithm, which fills the scheduling areas in each sector, as a benchmark to which we compare the performance of the algorithms proposed in Section IV and Section V under various network parameters. Third, we evaluate the performance gain from considering both joint scheduling and dynamic MCS selection (MC-GAP) compared to using only joint scheduling (GAP).

### A. Network Model

Fig. 6 shows the LTE network considered in the simulation study. Scheduling is performed for the cell in the center of the network, while the surrounding cells are considered for the calculations of the SINR experienced by each receiver. Our interference model and parameters are based on the 3GPP specifications [1] and on the work presented in [33]. These parameters are summarized in Table II. The number of reuse-1 blocks in a 1-ms subframe is 40 in each sector and the number of reuse-(1/3) blocks is 20.

As proposed in [33], each antenna is 20 meters high, and has a vertical tilt of 16°. The distance between two antennas in neighboring cells is 1700 meters.

The average size of each packet is 3.5 scheduled blocks if it is transmitted using [QPSK, 1/2], which is the most robust MCS out of 7 possible MCSs. The success probability for every [scheduling area, user, MCS] triplet is determined from the corresponding SINR value using data taken from [6]. The profit from transmitting a packet to a user using a particular MCS is taken as the corresponding success probability. Thus, our utility function in this section aims at maximizing the expected number of successfully delivered packets. The cost of transmitting a packet is equal to the discrete number of scheduled blocks used for the transmission, which depends on the length of the packet and the chosen MCS. The interference model of the network is described in the Appendix.

### *B. The Simulated Joint Scheduling Algorithms*

We compare the performance of our algorithms to a standard water-filling algorithm, which works as follows. Each user device is associated with the sector whose antenna yields the best SINR. When a new packet is introduced, the algorithm first tries to schedule the packet in the reuse-1 area of this sector using the default MCS. If there are not enough scheduled blocks available in the reuse-1 area, the algorithm tries to schedule the packet using the default MCS in the reuse-1/3 area of the same BS.

The benefit of our joint scheduling algorithms compared to this water-filling algorithm can be divided into two parts. First, for each sector we solve the problem for both 1/3- and 1-reuse areas together, which can be viewed as intra-sector joint scheduling. Second, we solve the problem for all the sectors in the cell together, which can be viewed as inter-sector joint scheduling. To distinguish between the benefit from each part, we implement two versions of each algorithm: one that uses only intra-sector joint scheduling and one that uses both inter- and intra-sector joint scheduling. Thus, for the rest of this section we refer to the following 4 algorithms:

- Alg-1: a GAP algorithm, used for inter-sector joint scheduling using only a default MCS for each packet.
- Alg-2: $\text{ALG}_{\text{MC-GAP}}$, used for inter-sector joint scheduling with dynamic MCS selection.
- Alg-3: a GAP algorithm, used for intra-sector joint scheduling using only a default MCS for each packet.
- Alg-4: $\text{ALG}_{\text{MC-GAP}}$, used for intra-sector joint scheduling and dynamic MCS selection.

For the simulations, we implemented modified versions of the approximation algorithm for GAP (from [13]) and for MC-GAP (from Sections IV and V). The purpose of these modifications is to improve the average-case performance of these algorithms without affecting their lower bounds. Instead of considering the bins (scheduling areas) in some arbitrary order, we consider 4 specific orderings, and choose the one that yields the maximum profit. The considered orderings are as follows:

(a) an ordering where a reuse-1 bin is chosen before a reuse-(1/3) bin of the same antenna.
(b) an ordering where a reuse-(1/3) bin is chosen before a reuse-1 bin of the same antenna.
(c) an ordering where all reuse-1 bins are chosen before all reuse-(1/3) bins.
(d) an ordering where all reuse-(1/3) bins are chosen before all reuse-1 bins.

For the GAP and MC-GAP algorithms invoked for solving Problem 1 and Problem 2 respectively, we use as a procedure the optimal pseudopolynomial time algorithm for MCKP [23]. Thus, both GAP and MC-GAP algorithms are 2-approximation.

The simulations are conducted using a Linux virtual machine with 1GB memory and 1 core. The running time of our most intensive algorithm (Alg-2) is always less than 1ms and can be improved by using a virtual machine with more resources. Because this running time is measured for a very large number of users (more than 200), the running time of the scheduler is expected, in practice, to be much shorter.

### *C. Simulation Results*

Throughout this section, to draw one point on a graph, 100 random instances are generated and the results are averaged. First, we want to compare the performance of $\text{ALG}_{\text{MC-GAP}}$ to the optimal solution. Since MC-GAP is NP-hard, we use an exponential time brute-force algorithm for finding the optimal solution for small instances (15 packets) and compare this solution to the one found by $\text{ALG}_{\text{MC-GAP}}$. We test different network parameters and the results show that the actual profit obtained by $\text{ALG}_{\text{MC-GAP}}$ is only 4-6% lower than that of the optimal solution. This suggests that the new algorithm performs very well.

We now compare the performance of our algorithms to the standard water-filling algorithm described in Section VI-B. We use 2 different running sets, which differ in how user devices are distributed across a scheduling cluster. In Fig. 7(a) the user devices are uniformly distributed, while in Fig. 7(b) the probability of a user device to be in sector 1 is 20 times greater than its probability to be in sector 2 or sector 3. Both figures show the ratio between the profit of each of the four algorithms described in Section VI-B and the profit of the water-filling algorithm, as a function of the normalized load. The load is defined as the number of waiting packets divided by the total number of scheduled blocks in the cell. The number of users is identical to the number of waiting packets because we assign to each

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| **network layout** | 7 BSs | **TX power** | 39dBm |
| **system bandwidth** | 20MHz | **inter-site distance** | 1700m |
| **BS antenna height** | 20m | **user height** | 1.5m |
| **propagation loss model** | Hata model | **system frequency** | 1,500 MHz |
| **TX antenna gain** | 18.9dBi | **vertical tilt** | $-16°$ |
| **vertical half power beam width ($\theta_{\mathbf{3dB}}$)** | $+10°$ | **horizontal half power beam width ($\varphi_{\mathbf{3dB}}$)** | $+70°$ |
| **side lobe attenuation (SLA$_v$)** | 20dB | **front-to-back attenuation ($A_m$)** | 25dB |

TABLE II
SIMULATION NETWORK PARAMETERS

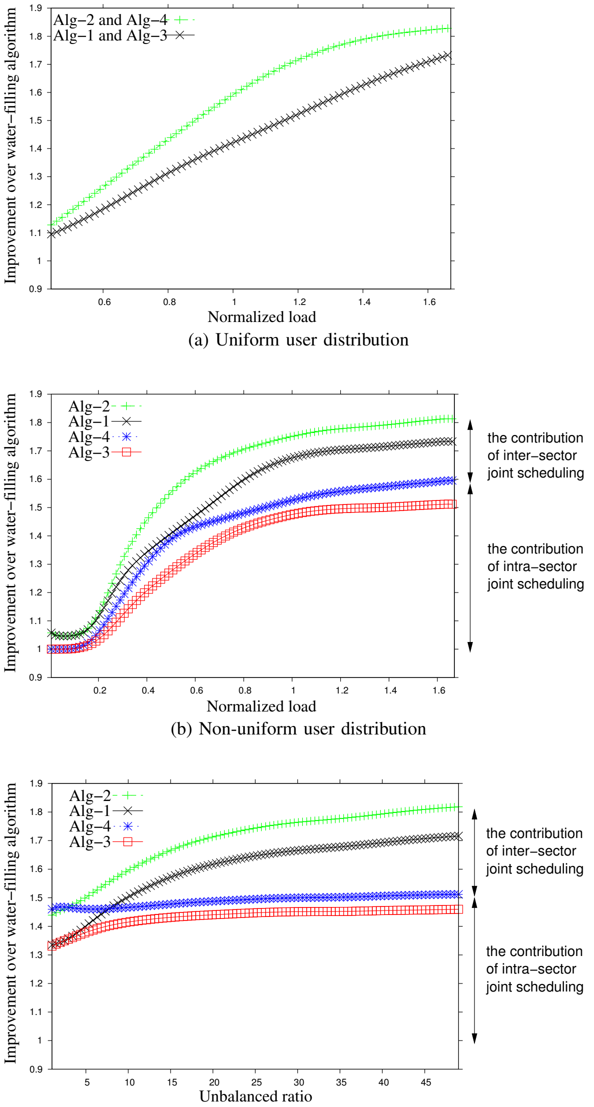
user one packet on average. In general, we see that all 4 algorithms perform much better than the water-filling algorithm, and that the performance gain increases when the load increases.

In Fig. 7(a) we see that the performance of Alg-1 is equal to that of Alg-3 (a single curve is shown for both algorithms), and the performance of Alg-2 is equal to that of Alg-4 (a single curve is shown for both algorithms). This implies that in this setting, all the benefit compared to the water-filling algorithm is attributed to intra-sector joint scheduling. The reason is that when the users are uniformly distributed, there is no advantage from scheduling a user using the resources of a remote sector. This is in contrast to Fig. 7(b), where user distribution is not uniform; thus, Alg-1 is significantly better than Alg-3 and Alg-2 is significantly better than Alg-4.

In the next set of simulations we investigate how user distribution affects the benefit obtained by the various algorithms. The x-axis in Fig. 7(c) shows the ratio between the probability of a user to be in sector 1 and the probability of a user to be in sector 2 or sector 3. As before, all 4 algorithms perform better than the water-filling algorithm, and the gain increases when the unbalanced ratio increases. As expected, we can see that the contribution of inter-sector joint scheduling is significantly greater than the contribution of intra-sector joint scheduling for higher values of unbalance ratio.

Finally, we show how different profit functions affect the performance. To this end, we compare $p_{\text{throughput}}$ and $p_{\text{delay}}$, where in $p_{\text{delay}}$ the profit is set to 3, which is larger than the maximum value of $p_{\text{throughput}}$, when the packet must to be scheduled within 3 subframes or less in order to meet its deadline. According to the definition of $p_{\text{delay}}$, before this time the profit of the packet is $p_{\text{throughput}}$. After the deadline, the profit drops to zero.

To generate a single point in the following graphs, we consider 500 consecutive OFDMA subframes. At the beginning of each subframe, a fixed number of new packets are introduced. Each new packet is uniformly associated with a number between 5 and 15, which determines the number of subframes (i.e., the time) before its deadline. The load is defined as the expected total number of blocks required to transmit the new packets using the most efficient modulation, divided by the total number of blocks available in the cell. Then, we invoke each algorithm for every subframe, and remove the scheduled packets. Not yet scheduled packets are considered by the scheduler in the next subframe, after their profit value increases, if needed. We use $\text{ALG}_{\text{MC-GAP}}(p')$ to denote an execution of $\text{ALG}_{\text{MC-GAP}}$ when a profit function $p'$ is used.



(a) Uniform user distribution

(b) Non-uniform user distribution

(c) As a function of the unbalanced ratio

Fig. 7. Total profit improvement ratio over water-filling algorithm for the 4 algorithms

Fig. 8 shows the fraction of packets transmitted on time as a function of the load. For every load value, $\text{ALG}_{\text{MC-GAP}}(p_{\text{delay}})$ always schedules on time at least as

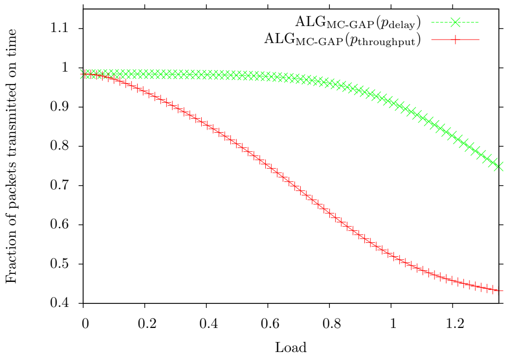


Fig. 8. Fraction of packets transmitted on time as a function of the load, for $\text{ALG}_{\text{MC-GAP}}(p_{\text{throughput}})$ and $\text{ALG}_{\text{MC-GAP}}(p_{\text{delay}})$

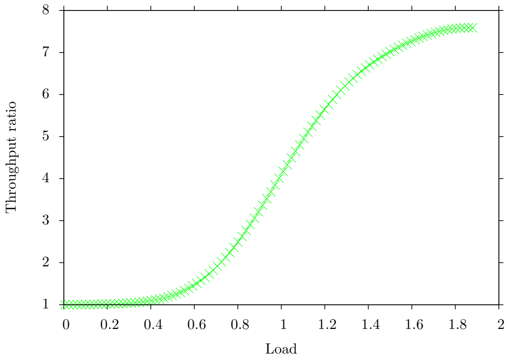


Fig. 9. Throughput ratio between $\text{ALG}_{\text{MC-GAP}}(p_{\text{throughput}})$ and $\text{ALG}_{\text{MC-GAP}}(p_{\text{delay}})$ as a function of the load

many packets as $\text{ALG}_{\text{MC-GAP}}(p_{\text{throughput}})$. For very small loads, all new packets can be scheduled as soon as they are introduced. Therefore, both $\text{ALG}_{\text{MC-GAP}}(p_{\text{delay}})$ and $\text{ALG}_{\text{MC-GAP}}(p_{\text{throughput}})$ transmit all packets on time. When the load increases, $\text{ALG}_{\text{MC-GAP}}(p_{\text{delay}})$ performs better than $\text{ALG}_{\text{MC-GAP}}(p_{\text{throughput}})$.

Fig. 9 shows the throughput ratio between $\text{ALG}_{\text{MC-GAP}}(p_{\text{throughput}})$ and $\text{ALG}_{\text{MC-GAP}}(p_{\text{delay}})$. As expected, for small loads, the throughput ratio is 1 since both algorithms schedule all packets. When the load increases, $\text{ALG}_{\text{MC-GAP}}(p_{\text{delay}})$ transmits more packets which are about to expire using inefficient MCSs. This comes at the expense of scheduling less packets using efficient MCSs and therefore the throughput ratio increases.

## VII. Conclusions

We addressed the new OFDMA joint scheduling problem encountered by a BS that controls multiple sectors, we showed that it is equivalent to the well-known NP-hard GAP problem. In order to further improve the joint scheduler's performance, we extended its role to also determine the MCS to be used for each packet. This resulted in a new NP-hard problem, which we called MC-GAP, and for which we proposed an efficient and practical approximation scheme. We conducted an extensive system level simulation study of the various algorithms, under various network parameters and for different optimization criteria, and showed that the performance of the new MC-GAP algorithm is very close to optimal and that our proposed joint scheduling algorithms significantly increase the throughput of an OFDMA network.

## References

[1] 3GPP. E-UTRA; Further Advancements for E-UTRA Physical Layer Aspects,TR 36.814 .

[2] 3GPP. Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 11), 3GPP TS 36.213, Oct. 2012.

[3] D. Amzallag, R. Bar-Yehuda, D. Raz, and G. Scalosub. Cell selection in 4G cellular networks. *IEEE INFOCOM*, 2008.

[4] G. Aniba and S. Aïssa. Adaptive scheduling for MIMO wireless networks: cross-layer approach and application to HSDPA. *IEEE Transactions on Wireless Communications*, 6(1):259–268, 2007.

[5] A. Argento, M. Cesana, and I. Malanchini. On access point association in wireless mesh networks. *WoWMoM*, pages 1–6, 2010.

[6] K. Balachandran et al. Design and analysis of an IEEE 802.16e-based OFDMA communication system. *BLTJ*, 11(4), 2007.

[7] M. Bansal and V. Venkaiah. Improved fully polynomial time approximation scheme for the 0-1 multiple-choice knapsack problem. *SIAM Conference on Discrete Mathematics*, 2004.

[8] R. Bar-Yehuda and S. Even. A local-ratio theorem for approximating the weighted vertex cover problem. *Annals of Discrete Mathematics*, 25:27–45, 1985.

[9] H.-H. Choi, J. B. Lim, H. Hwang, and K. Jang. Optimal handover decision algorithm for throughput enhancement in cooperative cellular networks. *IEEE VTC Fall*, pages 1–5, 2010.

[10] R. Cohen and G. Grebla. Efficient allocation of CQI channels in broadband wireless networks. *IEEE INFOCOM*, pages 96 –100, April 2011.

[11] R. Cohen and L. Katzir. A generic quantitative approach to the scheduling of synchronous packets in a shared uplink wireless channel. *IEEE/ACM Trans. Netw.*, 15(4):932–943, Aug. 2007.

[12] R. Cohen and L. Katzir. Computational analysis and efficient algorithms for micro and macro OFDMA downlink scheduling. *IEEE/ACM Trans. Netw.*, 18(1):15–26, 2010.

[13] R. Cohen, L. Katzir, and D. Raz. An efficient approximation for the generalized assignment problem. *Information Processing Letters*, 100(4):162–166, Nov. 2006.

[14] A. Das, K. Balachandran, F. Khan, A. Sampath, and H. Su. Network controlled cell selection for the high speed downlink packet access in UMTS. *IEEE WCNC*, 4:1975–1979, Mar. 2004.

[15] A. De Domenico, E. Strinati, and A. Duda. An energy efficient cell selection scheme for open access femtocell networks. *IEEE PIMRC*, pages 436–441, 2012.

[16] C. Dhahri and T. Ohtsuki. Learning-based cell selection method for femtocell networks. *IEEE VTC*, pages 1–5, 2012.

[17] M. Feng, X. She, L. Chen, and Y. Kishiyama. Enhanced dynamic cell selection with muting scheme for DL CoMP in LTE-A. *IEEE VTC Spring*, pages 1–5, 2010.

[18] H. Fu and D. I. Kim. Downlink scheduling with AMC and FCS in WCDMA networks. *IEEE GLOBECOM*, 2007.

[19] G. Gens and E. Levner. An approximate binary search algorithm for the multiple-choice knapsack problem. *Information Processing Letters*, 67(5):261–265, 1998.

[20] A. Han and I.-T. Lu. Optimizing beyond the carrier by carrier proportional fair scheduler. *IEEE Sarnoff Symposium*, pages 1–5, may 2011.

[21] M. Hata. Empirical formula for propagation loss in land mobile radio services. *IEEE Transactions on Vehicular Technology*, 29(3):317–325, Aug. 1980.

[22] J. Huang, V. G. Subramanian, R. Agrawal, and R. A. Berry. Downlink scheduling and resource allocation for OFDM systems. *IEEE Trans. on Wireless Communic.*, 8(1):288–296, 2009.

[23] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

[24] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8(1):33–37, 1997.

[25] Y. M. Kwon, O. K. Lee, J. Y. Lee, and M. Y. Chung. Power control for soft fractional frequency reuse in OFDMA system. *ICCSA*, 6018:63–71, 2010.

[26] E. L. Lawler. Fast approximation algorithms for knapsack problems. *Math. Oper. Res*, 4(4):339–356, 1979.

[27] P. Mitran, C. Rosenberg, J. Sydor, J. Luo, and S. Shabdanov. On the capacity and scheduling of a multi-sector cell with co-channel interference knowledge. *Med-Hoc-Net*, pages 1–8, 2010.

[28] T. D. Novlan, J. G. Andrews, I. Sohn, R. K. Ganti, and A. Ghosh. Comparison of fractional frequency reuse approaches in the OFDMA cellular downlink. *IEEE GLOBECOM*, 2010.

[29] R. Pitic and A. Capone. An opportunistic scheduling scheme with minimum data-rate guarantees for OFDMA. *IEEE WCNC*, pages 1716–1721, 2008.

[30] T. Qu, D. Xiao, and D. Yang. A novel cell selection method in heterogeneous LTE-advanced systems. *IC-BNMT*, Oct. 2010.

[31] T. Qu, D. Xiao, D. Yang, W. Jin, and Y. He. Cell selection analysis in outdoor heterogeneous networks. *ICACTE*, 2010.

[32] T. Sun, Y. Zhang, and W. Trappe. Improving access point association protocols through channel utilization and adaptive switching. *IEEE MASS*, pages 155–157, 2011.

[33] N. Tabia, A. Gondran, O. Baala, and A. Caminada. Interference model and evaluation in LTE networks. *(WMNC)*, Oct. 2011.

[34] Y. J. Zhang and K. B. Letaief. Multiuser adaptive subcarrier-and-bit allocation with adaptive cell selection for OFDM systems. *IEEE Trans. Wireless Communic.*, 3(5):1566–1575, 2004.

## Appendix A
## Simulation Interference Model

We start by describing how the SINR of each user is calculated as a function of the end power it experiences. Recall that the bandwidth of each cell is partitioned into 4 subbands: F0, F1, F2 and F3 (Fig. 2). Let $S_i$ be the set of scheduling areas that use subband $Fi$, for $i \in \{0, 1, 2, 3\}$. For example, in the 7-cell network presented in Fig. 6, $|S_1| = |S_2| = |S_3| = 7$ and $|S_0| = 21$. Let $p_s(u)$ be the power received by user $u$ in scheduling area $s \in S_i$. The SINR experienced by $u$ is defined by:

$$\gamma_s(u) = \frac{p_s(u)}{\displaystyle\sum_{s' \neq s, s' \in S_i} p_{s'}(u) + n_0 w},$$

where $w$ is the total bandwidth used in the sector, $n_0$ is the thermal noise over the bandwidth $w$, and the end power $p_s(u)$ is given by the following equation [33]:

$$\begin{aligned} p_s(u) = & p_s - \text{PL}_s(u) + g_s - \\ & a_s^{\text{ver}}(\theta_s(u)) - a_s^{\text{hor}}(\varphi_s(u))(\text{dBm}), \end{aligned}$$

where $p_s$ is the power, in dBm, of the antenna transmitting to scheduling area $s$, and $g_s$ is the gain of this antenna. In addition, $a_s^{\text{ver}}$ and $a_s^{\text{hor}}$ are the vertical and horizontal radiation pattern due to the position of the user in relation to that of the transmitting antenna. Thus, they are a function of the vertical angle $\theta_s(u)$ and horizontal angle $\varphi_s(u)$ between the user and the antenna main beam. The path loss is estimated using the Hata propagation model for small to medium-sized cities and is denoted $\text{PL}_s(u)$.

The vertical and horizontal radiation patterns are calculated using the following equations [33]:

$$a_s^{\text{hor}}(\theta_s(u)) = -\min\left(12\left(\frac{\theta_s(u)}{\theta_{3\text{dB}}}\right), \text{SLA}_v\right)$$

$$a_s^{\text{ver}}(\varphi_s(u)) = -\min\left(12\left(\frac{\varphi_s(u)}{\varphi_{3\text{dB}}}\right), A_m\right),$$

where $\text{SLA}_v = 20\text{dB}$ is the side lobe attenuation, $A_m = 25\text{dB}$ is the front-to-back attenuation, and $\theta_{3\text{dB}}$, $\varphi_{3\text{dB}}$ are the half power beam width in vertical, horizontal plane respectively. The Hata propagation model for urban areas is calculated using the following equation [21]:

$$\begin{aligned} \text{PL}_s(u) &= 69.55 + 26.16\log_{10}(f_0) - 13.82\log_{10}(z_s) \\ &- a(z_u) + (44.9 - 6.55\log_{10}(z_u))\log_{10}(d_s(u)), \end{aligned}$$

where $f_0 = 1500MHz$ is the frequency of transmission, $z_s$ is the height (meters) of the antenna used for scheduling area $s$, $z_u$ is the height (meters) of user $u$, $d_s(u)$ is the distance (kilometers) between $u$ and the antenna of scheduling area $s$, and $a(z_u) = 0.8 + (1.1 \cdot \log_{10}(f_0) - 0.7) \cdot z_u - 1.56\log_{10}(f_0)$ for a small/medium sized city.

**Reuven Cohen** received the B.Sc., M.Sc. and Ph.D. degrees in Computer Science from the Technion - Israel Institute of Technology, completing his Ph.D. studies in 1991. From 1991 to 1993, he was with the IBM T.J. Watson Research Center, working on protocols for high speed networks. Since 1993, he has been a professor in the Department of Computer Science at the Technion. He has also been a consultant for numerous companies, mainly in the context of protocols and architectures for broadband access networks. Reuven Cohen has served as an editor of the IEEE/ACM Transactions on Networking and the ACM/Kluwer Journal on Wireless Networks (WINET). He was the co-chair of the technical program committee of Infocom 2010 and headed the Israeli chapter of the IEEE Communications Society from 2002 to 2010.

**Guy Grebla** received the B.A., M.A., and Ph.D. degrees in Computer Science from the Technion - Israel Institute of Technology, completing his Ph.D. studies in 2013. He is now a postdoctoral research scientist in Electrical Engineering department at Columbia University, New York, NY.