

Computational Analysis and Efficient Algorithms for Micro and Macro OFDMA Scheduling

Reuven Cohen Liran Katzir
Department of Computer Science
Technion–Israel Institute of Technology
Haifa 32000, Israel

Abstract—OFDMA is one of the most important modulation and access methods for the future mobile networks. Before transmitting a frame on the downlink, an OFDMA base station has to invoke an algorithm that determines which of the pending packets will be transmitted, what modulation should be used for each of them, and how to construct the complex OFDMA frame matrix as a collection of rectangles that fit into a single matrix with fixed dimensions. We propose efficient, and theoretically best possible, algorithms that solves this intricate OFDMA scheduling problem by breaking it down into two sub-problems, referred to as macro and micro scheduling. We analyze the computational complexity of these sub-problems and develop efficient algorithms for solving them.

I. INTRODUCTION

Orthogonal Frequency Division Multiple Access (OFDMA) is one of the most important modulation and multiple access methods for the future mobile networks. It is an extension of OFDM, which is today the modulation of choice for non-mobile wireless access systems such as IEEE 802.11 (WiFi) and IEEE 802.16 (WiMax). OFDM divides a single frequency band into dozens of sub-carriers for parallel transmissions by the same user. This division increases the tolerance to noise and multi-path, while enabling more efficient use of bandwidth allocation. OFDMA extends OFDM by dividing the original band into many subchannels, each comprising a group of orthogonal carriers. Various modulations and FEC (Forward Error Correction) techniques are used for each subchannel, in order to provide improved coverage and throughput.

Unlike OFDM, OFDMA has many intricate constraints on its frame structure. The structure of an OFDMA downlink frame is depicted in Figure 1. The frame can be viewed as a 2-dimensional matrix, with the y-axis indicating the number of subchannels, each consisting of several orthogonal and not necessarily adjacent frequencies, and the x-axis indicating the time. The frame starts with a Frame Control Header (FCH), which contains information about the code rate, modulation level, and the length of the downlink (DL) and uplink (UL) maps. This part of the frame is always coded using a pre-determined Phy-profile. The data messages (PDUs) are transmitted in multiple bursts. There are 7 bursts shown in Figure 1. Each burst is transmitted by the base station using a specific Phy-profile. Due to Physical layer considerations, each data burst must be allocated a rectangular region within the frame. A burst may contain one or more PDUs destined for one or more receivers. To ensure correct decoding of the received signals, certain Phy-profile information about every burst is required. This information is included as overhead in the DL map burst.

This paper studies combinational aspects of scheduling on an OFDMA channel. Before transmitting a downstream frame,

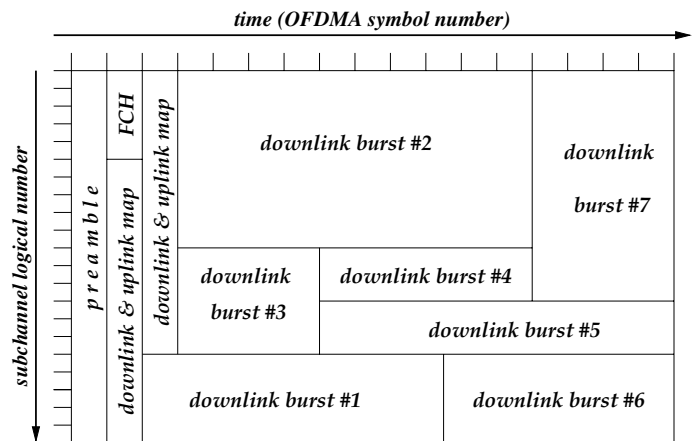


Fig. 1. The Structure of an OFDMA Downlink Frame

typically once every few ms., the base station has to invoke a scheduling algorithm that will generate the frame matrix as shown in Figure 1. To this end, the scheduler needs to address the following decision problems:

- To decide which Phy-profile will be used for each PDU. There are several dozen potential profiles, each with its own bandwidth cost and robustness against transmission errors. It is not possible to transmit all the profiles in the same frame. Moreover, each profile introduces a fixed significant overhead in the DL map field of the frame, and therefore, because of throughput considerations, the scheduler should try to minimize the number of profiles accommodated in every frame.
- To determine which PDU will be transmitted in the next frame. This decision has to take into account many factors, such as: (a) Quality of Service, since some of the PDUs have a guaranteed upper bound on their maximum delay; (b) total throughput maximization, since transmission to some of the hosts is difficult and requires more bandwidth for reliable delivery, and (c) fairness.
- To decide where exactly in the frame every burst will be located. Here there are also several constraints, some of which are: (a) power boosting, namely the ability of the base station to increase the transmission power used for some burst while decreasing the power used for other bursts transmitted at the same time on different subchannels; (b) efficiency, since the requirement that every Phy-profile will be represented as a rectangle in the frame matrix may leave some unused space in each rectangle.

There are two kinds of difficulties associated with addressing these problems. The first is that a solution for each problem depends on the solutions for the other two. This leads to a circular dependency, which has to be broken somehow. The second difficulty is that each of these decision problems is NP-complete, which means that even if the circular dependency is broken, finding an efficient scheduling algorithm is difficult.

In this paper we address the OFDMA scheduling problem in two approaches. In the first approach (Section IV and Section V) we assume that the base station determines in advance the Phy-profile to be used for each PDU, and the profit gained from transmitting this PDU using its selected Phy-profile. Then, there are two phases. In the first phase, referred to as *macro scheduling*, the base station determines which of the PDUs will actually be selected for transmission. In the second phase, referred to as *micro scheduling*, the scheduler determines how to build the OFDMA matrix from the selected PDUs. In the second approach we extend the macro scheduler such that it also determines the Phy-profile for each PDU. This is referred to as *extended macro scheduling*.

The assignment of profit to PDUs, which is based on various considerations such as QoS, throughput maximization, fairness, and channel state information (CSI), is beyond the scope of this paper. This issue is addressed in [4] in the context of uplink transmission, and similar ideas can be implemented on the downlink as well.

The main contributions of this paper are:

- Breaking the OFDMA scheduling problem into two more tractable problems, referred to as macro scheduling and micro scheduling.
- Analyzing the computational complexity of the macro and micro scheduling problems, and providing the best possible approximations.
- Developing efficient and practical algorithms for these NP-hard scheduling problems. Specifically, we provide an optimal macro scheduling problem, and an algorithm with only 2.5% overhead for the micro scheduling problem.
- Introducing a new concept in which the transmission profile of a PDU is not only determined by the channel condition of the intended receiver, but on the entire system conditions. To this end, we extend the macro scheduling problem to include the selection of transmission parameters for each PDU, and provide an efficient algorithm for this extended macro scheduling problem. This algorithm has the best possible approximation guarantee up to a small constant. It provides a significant improvement over the algorithm for the macro scheduling problem, at the cost of higher running time.

The rest of this paper is organized as follows. In Section II we discuss related work. In Section III we divide the OFDMA scheduling problem into macro and micro scheduling sub-problems, and discuss their computational complexity. In Section IV we present efficient algorithms for the macro scheduling sub-problem and in Section V we present such algorithms for the micro scheduling sub-problem. Section VI presents an extension to the macro scheduling problem, which allows the base station to select a Phy-profile for every PDU as a part of the scheduling process. This extension is shown to significantly improve the performance of the scheduler. Section VII presents a simulation study and Section VIII concludes the paper.

II. RELATED WORK

While much work has been done on scheduling in wireless networks, only a few papers address resource allocation in OFDMA channels. In [2], some variations of MiSP are addressed. In these variations, the input consists of bursts and their sizes. The bursts have priorities and the objective is to find an efficient allocation with accordance to these priorities. The main differences between [2] and our paper are that: (a) we analyze the computational complexity of MiSP and propose also approximation algorithms while [2] presents only heuristics; (b) we address not only MiSP but also MaSP and the combination between MiSP and MaSP; and (c) we consider a more general case, where profit is assigned to each PDU and not to each burst (a collection of PDUs).

In [8], a variation of MiSP is addressed. In this variation, the bursts have a specific order in which they can be scheduled. That is, in order to schedule burst i , all bursts $1 \dots i - 1$ must also be scheduled. The authors provide both theoretical hardness results, and an approximation algorithm. The main differences between [8] and our paper are that: (a) we address not only MiSP but also MaSP and the combination between MiSP and MaSP; and (b) in our paper the bursts are not scheduled based a predetermined order, but according to the profit of their PDUs.

In [14], it is shown that when an earliest-deadline-first greedy algorithm is implemented over good channels of a centralized wireless network, like the one we consider, the number of packets lost due to deadline expiration is minimized. In [12], the authors provide a fluid fair queueing scheduler for a noisy channel. In [15], the authors present an efficient fair queueing approximation of a noisy channel using deficit round robin, which takes less time to process. In [1], a scheduling algorithm that uses an N-state Markov model to characterize the channel is presented. This algorithm supports adaptive modulation and coding (AMC), which are used to adjust the modulation and FEC to the forecasted channel state. The idea of assigning higher data rates to hosts with a better channel – to maximize throughput while ensuring acceptable bit-error rate (BER) – is used by [5], [13].

In the uplink channel of an OFDM network, multiple hosts can transmit simultaneously over different sub-carriers. Since the channel characteristics for different users may be independent, dynamic assignment of sub-carriers to hosts can significantly improve the throughput [17], [18]. However, this “water filling” approach for maximizing instantaneous throughput does not take into account the QoS requirements of these packets, and it is therefore unsuitable for synchronous traffic. The authors of [11] address this problem in the context of OFDMA, by allocating sub-carriers to hosts in a way that satisfies the rate requirements of each host, while using minimum power. In [16], utility-based cross-layer optimization problems are defined. In [19], the authors provide an algorithm for OFDMA power assignment.

III. PROBLEM FORMULATION AND COMPLEXITY CLASSIFICATION

Consider a set of PDUs awaiting transmission. Each PDU can be transmitted using several different Phy-profiles, where each such a profile includes a modulation technique, code rate, and FEC. There are two possible timings for selecting a Phy-profile for each PDU. The first is to determine the Phy-profile for each PDU in advance (a priori), and only then to choose the PDUs for transmission. The second is to determine

first the profit of scheduling each PDU using each of the possible Phy-profiles, and only then to let the scheduler pick, on the fly, at most one instance for each PDU. In other words, with the a priori approach the selection of Phy-profiles is not performed by the macro scheduler, but by another algorithm that is executed before the macro scheduler has to make its decisions. We assume that even if the selection of a Phy-profile is performed a priori, the base station has the same information about the physical layer that it has when this selection is performed on the fly. Hence, the advantage, to be shown later, of using the on the fly approach, stems from the coupling between the selection of a Phy-profile for each PDU and the other tasks of the macro-scheduler. We address the a priori Phy-profile selection in Section IV, and the on the fly selection in Section VI.

The scheduling space in every frame is a two dimensional matrix, of T time slots and C OFDMA logical channels. Due to physical layer considerations, the scheduler must divide this matrix into rectangles, where each rectangle is used for the transmission of one or more PDUs from the same Phy-profile [7]. These rectangles are referred to as a *burst*. In the beginning of the frame, each rectangle has an overhead for describing its exact location within the frame and its Phy-profile attributes. The length H of this overhead is constant for each Phy-profile. Due to its critical role, this overhead is transmitted using the most robust and bandwidth-consuming modulation technique, and it consumes a significant portion of the frame bandwidth. Optimizing the usage of this header is therefore critical for the efficiency of the scheduler.

The goal of the scheduler is to maximize the profit gained by the transmitted PDUs in each frame. This goal requires the scheduler to make two types of decisions for every downstream frame:

MaSP: A Macro Scheduling Decision: Deciding which Phy-profiles will be accommodated in this frame, and which PDUs will be transmitted for every selected Phy-profile, assuming that the association between PDUs and their Phy-profiles has already been determined.

MiSP: A Micro Scheduling Decision: Deciding how many rectangles (bursts) will be used for each Phy-profile, and where to locate each rectangle within the frame.

We later show that the macro scheduling problem (MaSP) is equivalent to the well-known NP-hard Multiple-Choice Knapsack Problem [9]. The micro scheduling problem (MiSP) is associated with a tradeoff that has a critical impact on the performance of the downstream channel. On one hand, it is important to minimize the number of bursts for each Phy-profile, because each such burst has a significant overhead header. On the other hand, with smaller rectangles it is easier to minimize the bandwidth that is allocated to a rectangle but not fully used.

The table in Figure 2 classifies the two problems with respect to their computational complexity. Both problems are NP-hard. However, we prove that they can be solved in polynomial time within $(1 + \epsilon)$ from their optimum, for any $\epsilon > 0$. The selection of ϵ has, of course, a critical effect on the running time of the scheduler. The last row in the table is related to the extended MaSP problem, to be discussed in Section VI.

IV. THE MACRO SCHEDULING PROBLEM (MASP)

We start with a formal definition of MaSP.

Problem 1 (MaSP):

Instance: The frame's size $L = T \cdot C$, the overhead H for using a Phy-profile, and a set $P = \{p_1, p_2, \dots, p_n\}$ of PDUs awaiting transmission. Each PDU p_i is associated with a Phy-profile $Phy(p_i)$, a non-negative profit $p(p_i)$, and a non-negative size $s(p_i)$.

Objective: Find a feasible subset $P' \subseteq P$ of PDUs with a maximum profit. The size of a subset P' is the cumulative size of its PDUs plus a penalty of H for every used Phy-profiles. A feasible subset is a subset whose size is $\leq L$. The profit of a subset P' is the cumulative profit gained by its PDUs.

We note that the well-known Knapsack problem is a special case of MaSP, with only one Phy-profile. The Knapsack problem is defined as follows:

Instance: : A set S of items s_1, s_2, \dots, s_m and a capacity c . Each item s_i has profit $p(s_i)$ and a weight $w(s_i)$.

Objective: Find a subset $S' \subseteq S$ of items such that this subset has a feasible packing, namely, $\sum_{s_j \in S'} w(s_j) \leq c$, and the aggregated profit $\sum_{s_j \in S'} p(s_j)$ is maximized.

We now show that MaSP is polynomially equivalent to the Multiple-Choice Knapsack Problem (MCKP). Therefore, an algorithm for MCKP can also solve MaSP with the same performance guarantee. MCKP is defined as follows:

Instance: : Disjoint classes N_1, \dots, N_m of items, and a capacity c . Each item $s \in N_j$ has a profit $p(s)$ and a weight $w(s)$.

Objective: : Find a subset S' of items with at most one item from each class that has a feasible packing, such that the aggregated profit $\sum_{s_j \in S'} p(s_j)$ is maximized.

Definitions:

- 1) An r -approximate solution for a maximization problem is a feasible solution whose profit is at least $\frac{1}{r}$ the profit of the optimal solution.
- 2) An r -approximation algorithm is an algorithm that always finds an r -approximate solution.
- 3) An approximation scheme is an algorithm that receives a parameter $\epsilon > 0$ along with the problem instance and produces a $(1 + \epsilon)$ -approximate solution for the specified instance.
- 4) A Polynomial Time Approximation Scheme (PTAS) is an approximation scheme whose running time is polynomial in the size of the input.
- 5) A Fully Polynomial Time Approximation Scheme (FPTAS) is a PTAS whose running time is also polynomial in $\frac{1}{\epsilon}$.
- 6) A pseudo-polynomial time algorithm is an algorithm whose running time is polynomial in the numeric value of the input. In contrast, a polynomial time algorithm is polynomial in the size of the input (the number of digits).

Both Knapsack and MCKP have several known approximation algorithms [9], and both have an FPTAS [9]. In addition, both problems have pseudo-polynomial dynamic programming algorithms that find the optimal solution [9]. In particular, there exists a pseudo-polynomial algorithm whose running time is $O(n \cdot c)$, where n is the number of items and c is the knapsack capacity. Such an algorithm is practical only when c is small.

Theorem 1: MaSP can be solved optimally in pseudo-polynomial time by a transformation to MCKP.

Proof: We transform an instance of MaSP to an instance of MCKP such that a feasible solution for MCKP with profit p is also a feasible solution to MaSP with the same profit.

| Problem | NP-hard | Approximation guarantee | Running time |
|---------------------------------|---------|---|--------------------------------|
| MaSP (Macro Scheduling Problem) | Yes | $(1 + \epsilon)$ -approximation for every fixed ϵ | polynomial in $1/\epsilon$ |
| MiSP (Micro Scheduling Problem) | Yes | $(1 + \epsilon)$ -approximation for every fixed ϵ | not polynomial in $1/\epsilon$ |
| E-MaSP (Extended MaSP) | Yes | $(\frac{2e-1}{e-1} + \epsilon)$ -approximation for every fixed ϵ | polynomial in $1/\epsilon$ |

Fig. 2. Computational Complexity and Running Time for the Various Scheduling Problems

Therefore, an optimal solution for MCKP is also an optimal solution for MaSP. The knapsack size is set to $T \cdot C$. For each Phy-profile phy , we define a class of items N_{phy} . There exists an item $s_A \in N_{phy}$ for every subset A of Phy-profile phy , where $w(s_A) = H + \sum_{a \in A} w(a)$ and $p(s_A) = \sum_{a \in A} p(a)$. It is easy to see that a feasible solution to MCKP with profit p is also a feasible solution to MaSP with profit p .

At first glance it seems that the time required for this reduction is not polynomial, because the number of subsets is exponential. However, if there are two subsets A and A' such that $w(s_A) = w(s_{A'})$, then only the one with the greatest profit will be selected. There are at most $T \cdot C$ different item sizes, and for each size the item's profit can be found by a pseudo-polynomial algorithm for the single knapsack problem. Therefore, this reduction takes pseudo-polynomial time. Furthermore, a careful implementation will find the profit for all possible sizes at once, thereby reducing the total running time to $O(T \cdot C \cdot n)$ for each profile. ■

Typical dimensions of an OFDMA frame are $T \approx 20$ and $C \approx 60$. For these values it is reasonable to use a pseudo-polynomial time algorithm. However, it is important to note that the same reduction as in Theorem 1 can also be performed implicitly, in polynomial time. In the following we show how to use this rationale for obtaining an FPTAS for MaSP. However, for the sake of completeness we first describe a simple FPTAS for MCKP. Although this is not the algorithm with the best running time, it is probably the simplest one.

Algorithm 1 (An FPTAS for MCKP): Let $M[i, j]$ be a table of size $m \times P$, where m is the number of classes of items, and P is the sum of the profits of all items. Let $M[i, j]$ be the minimum size knapsack with profit gain j from the N_1, N_2, \dots, N_i classes, and let n be the number of items. Using dynamic programming, M can be filled in $O(n \cdot P)$ time. In order to solve MCKP in polynomial time, we scale all profits by a factor of $\frac{n}{\epsilon \cdot P}$, and then calculate M using the new scaled profits. The returned solution must be within a $(1 + \epsilon)$ factor from the optimum [9].

Theorem 2: MaSP has an FPTAS.

Proof: First, observe that in Algorithm 1, if there are two items with the same profit then the one with the lowest size is selected. Algorithm 1 uses profit scaling and therefore is concerned only with a polynomial number of different profits. Hence, the reduction can be performed implicitly, in the following way: whenever the algorithm seeks an item with a specific scaled profit, it chooses the minimum weight item with this profit gain. Therefore, an approximation that uses profit scaling can be implemented in polynomial time. Since this is basically the same reduction, a feasible solution for MCKP with profit p is also a feasible solution to MaSP with the same profit. Thus, an α -approximation for MCKP is also an α -approximation for MaSP. ■

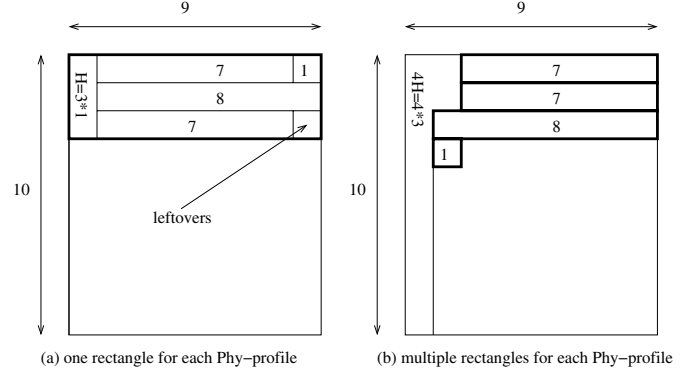


Fig. 3. The Two Approaches for Micro Scheduling

V. THE MICRO SCHEDULING PROBLEM (MiSP)

A. Computational analysis of MiSP

After the macro scheduler selects the set of PDUs to be transmitted, the micro scheduler has to build the transmission matrix from rectangles. The scheduler represents each Phy-profile as a sequence of one or more contiguous rectangles, and it faces the following trade-off. It can use many rectangles for each Phy-profile, thereby minimizing the leftovers at the end of each rectangle, it can use the opposite approach of a single rectangle for each Phy-profile, thereby minimizing the header overhead. Figure 3 depicts both approaches for 4 PDUs from the same Phy-profile set, whose sizes are 7, 7, 8, and 1. The first method consumes more space due to the burst overhead $H = 3$, while the second one consumes more space due to leftovers. A good scheduler should find the golden mean between these two opposite approaches.

Theorem 3: MiSP is NP-hard. Moreover, it does not have an FPTAS, unless $P=NP$.

The proof is omitted due to length constraints. It is presented in the full version of the paper [3].

B. A Linear Time Dual Approximation for MiSP

If the scheduling space of the OFDMA matrix had been continuous, i.e., the slot length had been as small as one would like (even a fraction of a bit), the MaSP output could have easily been scheduled as rectangles by setting the width and the length of each rectangle of size S to be C and $\frac{S}{C}$ respectively. Applying this intuition in practice would have resulted in a lot of wasted space, since a small profile would have consumed at least C slots. In the following we present an algorithm that captures this rationale yet deals with small profiles without wasting a lot of space. The algorithm is a dual $(1 + \epsilon)$ -approximation, which means that it can schedule all profiles using a space of $(1 + \epsilon)OPT$, where OPT is the space required for optimal allocation. The main idea is to partition the Phy-profiles into sets in which all Phy-profiles are roughly the same size, and to schedule each set separately.

We later present several heuristics that work in a similar way, but perform better in practice.

Note: In order for a dual approximation of MiSP to integrate with MaSP, the matrix size given to MaSP should be smaller by a factor of ϵ than the actual size.

Algorithm 2 (A dual approximation for MiSP): An algorithm for MiSP with a $(1 + \frac{1}{poly(T)})$ -approximation guarantee. The value of $poly(T)$ is $T^{\frac{1}{5}}$ in the worst case. The exact value is discussed later.

- 1) Let k and m be constants, whose values are discussed later.
- 2) Partition the Phy-profiles into $m+1$ sets in the following way. For $i = 1 \dots m-1$, set P_i contains all Phy-profiles whose size falls into the range $(k^{2-2^{-(i-1)}}, k^{2-2^{-i}}]$. Set P_0 contains all Phy-profiles whose size is smaller than k , and set P_m contains all Phy-profiles whose size is strictly larger than $k^{2-2^{-(m-1)}}$. For example, for $m = 3$, the range for P_0 is $[1, k]$, for P_1 it is $(k, k^{\frac{3}{2}}]$, for P_2 it is $(k^{\frac{3}{2}}, k^{\frac{7}{4}}]$, and for P_3 it is $(k^{\frac{7}{4}}, \infty)$.
- 3) For $i = 0 \dots m-1$ do
 - a) Set $r_i \leftarrow k^{1-2^{-i}}$ (that is, $1, k^{\frac{1}{2}}, k^{\frac{3}{4}}, \dots$).
 - b) Round up the size of each Phy-profile in P_i to an exact multiple of r_i .
 - c) Schedule P_i on strips of r_i consecutive rows, using a next-fit bin-packing algorithm [6].
Each Phy-profile is spread out across the entire height of the bin.
- 4) Round all Phy-profiles in P_m to an exact multiple of T , and schedule the items in P_m in some arbitrary order spread out across the entire T dimension.

Theorem 4: Algorithm 2 is a $(1 + \frac{1}{poly(T)})$ -approximation. More specifically, for $m = 2$ and $k = T^{\frac{1}{5}}$, it has a $(1 + T^{-\frac{1}{5}})$ -approximation guarantee.

The proof is omitted due to length constraints. It is presented in the full version of the paper [3].

Theorem 5: The running time of Algorithm 2 is linear in the number of Phy-profiles.

Proof: Steps 2 and 4 can be performed in a constant time per item. The linear running time of step 3(c) follows from [6]. ■

Algorithm 2 can be easily turned into a $(1 + \epsilon)$ -approximation for any ϵ by applying optimal dynamic programming to “large” areas when $\frac{1}{\epsilon} < poly(T)$. The details are quite technical, and are therefore omitted.

The next-fit bin-packing algorithm was good enough to achieve the approximation ratio. However, in practice, algorithms that outperform next-fit can improve the performance. One of these algorithms is the best-fit decreasing algorithm [6], which runs in $O(n \log n)$.

C. Heuristics for MiSP

Having presented an algorithm with worst-case performance guarantee, we now present several heuristics for MiSP. While these algorithms might not perform very well in the worst case scenario, their performance is very good in practice, as shown in Section VII.

The first algorithm is called Increasing Size. The rationale behind this algorithm is to schedule items of roughly the same size next to each other. A variation of this idea was used in Algorithm 2. However, while in Algorithm 2 the boundary between two sets is fixed, the Increasing Size algorithm tries to set it dynamically.

Algorithm 3 (An increasing size algorithm for MiSP):

- 1) Maintain the Phy-profiles as a list L sorted by their size in increasing order.
- 2) While there are Phy-profiles left in L do
 - a) For every k and t , check the relative space wasted when scheduling the first t Phy-profiles that fit on bins of k consecutive rows. This is the classic bin-packing problem of the first t Phy-profiles where each bin contains k consecutive rows and each item is spread out across the entire height of the bin. The space used, S , is $k \cdot T$ times the number of bins used, so the relative space wasted is S over the sum of sizes of the first t Phy-profiles.
 - b) Select the Phy-profiles that produce the minimum relative wasted space.
 - c) Remove the selected items from L , i.e., the first t items for the choice of k and t that minimized the space wasted.

Claim 1: The maximum value of k in Algorithm 3 is bounded by $2C$.

Proof: From Algorithm 2 and Theorem 4 we know that there exists a schedule such that the number of rows is at most $C \cdot (1 + T^{-\frac{1}{5}})$. We expect Algorithm 3 to outperform Algorithm 2, but one can see that every choice of $k > C \cdot (1 + T^{-\frac{1}{5}})$ clearly performs worse. Therefore, the maximum value of k in which this heuristic outperforms Algorithm 2 is bounded by $2C$. ■

Let n be the number of Phy-profiles. It is easy to see that $t \leq n$, and that the total number of choices of k and t for a single iteration is bounded by $2C \cdot n$. The running time for applying the best-fit decreasing algorithm for the bin-packing problem is $O(n \log n)$. Therefore, the total running time for a single iteration of the algorithm is $2C \cdot n \cdot n \log n$, and the total running time $O(2C \cdot n^3 \log n)$. Since a typical value of n is smaller than 10, and a typical value of C is smaller than 100, this running time can be considered reasonable.

The next algorithm is called Decreasing Size. The rationale behind this algorithm is the same as for Increasing Size. However, Phy-profiles are considered in reverse order, so that the asymptotic running time will decrease when the best-fit decreasing bin-packing algorithm is used.

Algorithm 4 (A decreasing size algorithm for MiSP):

Same as Algorithm 3, except that in step 1 the items are sorted in decreasing order.

The running time of Algorithm 4 is similar to that of Algorithm 3. However, as t grows, we can use the result of the previous iteration because the new Phy-profile has a smaller size. This reduces the total running time to $O(2C \cdot n^2 \log n)$.

The last algorithm we present is called Best Fit. The intuition behind this algorithm is to dynamically choose items that locally minimize the wasted space. In the previous algorithms this was achieved by selecting items that are roughly the same size. Here we try to achieve this goal by optimizing the schedule on k consecutive rows, as in Algorithm 2. But unlike Algorithm 2, this algorithm tries all possible combinations of Phy-profiles, and picks the best one.

Algorithm 5 (A best fit algorithm for MiSP):

- 1) Add all Phy-profiles to a list L .
- 2) While there are Phy-profiles left in L do
 - a) For every k , check the relative space wasted when scheduling any Phy-profiles with a fixed height of k . This is a knapsack problem, where items are

spread out across the entire height of the knapsack. The relative space wasted is $k \cdot T$ over the sum of sizes of the scheduled items.

- b) Select the Phy-profiles that produce the minimum relative wasted space.
- c) Remove the selected items from L .

The total running time of Algorithm 5 is $O(2C \cdot T \cdot n^2)$, where n is the number of Phy-profiles, because Claim 1 applies here as well. The running time of single iteration is, therefore, $2C \cdot T \cdot n$, and the total running time is $O(2C \cdot T \cdot n^2)$. Since a typical value of T is ≤ 40 , this running time can be considered reasonable.

VI. THE EXTENDED MACRO SCHEDULING PROBLEM (E-MASP)

As discussed in Section III, another approach for MaSP is not to decide in advance (off-line) what the Phy-profile of every PDU should be, but rather to make this decision on-line, as part of MaSP. The resulting optimization problem, referred to as E-MASP (Extended MaSP), is defined as follows:

E-MaSP: An Extended Macro Scheduling Decision: Deciding which Phy-profiles will be used for every PDU, which Phy-profiles will be accommodated in the next frame, and which PDUs will be transmitted for every selected Phy-profile.

An E-MaSP-based scheduler has advantages over a MaSP-based scheduler especially when the channel is not heavily loaded. As an example, consider a single PDU waiting for transmission. The MaSP-based scheduler will choose to schedule this PDU using a pre-determined Phy-profile. In contrast, due to the availability of bandwidth, the E-MaSP-based scheduler will transmit this PDU using the most robust Phy-profile. Hence, the profit gained by the E-MaSP scheduler in this case is likely to be higher than the profit gained by the MaSP scheduler, and this extra profit is gained at no additional cost.

We now formally define E-MaSP:

Problem 2 (E-MaSP):

Instance: : The frame's size $L = T \cdot C$, the overhead H for using a Phy-profile, a set $P = \{p_1, p_2, \dots, p_n\}$ of PDUs awaiting transmission, and a set $Phy = \{Phy_1, Phy_2, \dots, Phy_m\}$ of Phy-profiles. Each pair of PDU p_i and Phy-profile Phy_j is associated with a non-negative profit $p(Phy_j(p_i))$ and non-negative size $s(Phy_j(p_i))$. These profit and size are relevant when the considered PDU is transmitted using this specific Phy-profile. In addition, each Phy-profile has a fixed overhead H .

Objective: : Find a feasible schedule S with maximum profit. A schedule is a subset P' of PDUs and an assignment function f from P' to Phy . The size of a schedule S is equal to H times the number of Phy-profiles used by S , plus the cumulative size of the accommodated PDUs (recall that the size for each PDU depend upon the Phy-profile selected for it), namely, $H \cdot |\cup_{p_i \in P'} f(p_i)| + \sum_{p_i \in P'} s(f(p_i))$. A feasible schedule is a schedule whose size is $\leq L$. The profit of a subset S is the cumulative profit gained by the scheduled PDUs with respect to the Phy-profile selected for each of them, namely, $\sum_{p_i \in P'} p(f(p_i))$.

Note that in some cases a PDU is destined to a user that does not support certain Phy-profiles. This case can be captured by

E-MaSP by assigning a 0 profit to the combination of this PDU and each such Phy-profile.

In Sections III and IV we showed that both MaSP and MiSP are NP-hard but can be approximated with arbitrarily close precision in polynomial time (see Figure II). In the following we show that E-MaSP is not only NP-hard, but also cannot be approximated with a factor smaller than $\frac{e}{e-1}$. This implies that E-MaSP is computationally harder than MaSP and MiSP.

We start with the observation that the well-known NP-hard Maximum Coverage problem [6] is reducible to a special case of E-MaSP. This problem is defined as follows:

Instance: : A number k and a collection of sets $S = S_1, S_2, \dots, S_m$, where $S_i \subseteq \{x_1, x_2, \dots, x_n\}$.

Objective: : Find a subset $S' \subseteq S$ of sets, such that $|S'| \leq k$ and the number of covered elements $|\cup_{S_i \in S'} S_i|$ is maximized.

Theorem 6: E-MaSP is NP-hard. Moreover, it cannot be approximated within a factor small than $\frac{e}{e-1}$.

Proof: We show a reduction from Maximum Coverage to E-MaSP. Each item x_i in the Maximum Coverage instance is transformed into a PDU. Each set S_j is transformed into a Phy-profile. For an item x_i and a set S_j , if $x_i \in S_j$ then the profit and size of the assignment of the corresponding PDU to the corresponding Phy-profile are 1 and 0 respectively. If $x_i \notin S_j$, the profit and size are 0 and ∞ respectively. In addition, the overhead H for every Phy-profile is set to 1, and the frame size is set to k .

It is clear that at most k Phy-profiles can be scheduled in the next frame. For the scheduled Phy-profiles, it is clear that the total profit is not higher than the number of PDUs that can be assigned to the selected Phy-profiles. Therefore, a schedule that gains profit of p is translated into a maximum coverage solution whose size is $\leq k$ that covers at least p items. This proves that an optimal schedule gains no more than optimal maximum coverage.

On the other hand, a maximum coverage solution of size k that covers m elements can be transformed into a schedule whose size is k and that schedules m PDUs. This is done by choosing the corresponding Phy-profiles and assigning a (covered) PDU to one of them. Therefore, an optimal schedule gains as much as the optimal maximum coverage.

Since an optimal solution for maximum coverage can be translated (in polynomial time) into an optimal solution for the specific E-MaSP instance and vice versa then: (a) the fact that Maximum Coverage is NP-hard implies that E-MaSP is also NP-hard; (b) the fact that Maximum Coverage cannot be approximated within a factor smaller than $\frac{e}{e-1}$, under standard assumptions¹ [10], implies that E-MaSP also cannot be approximated within the same factor. ■

A simple observation from the reduction in Theorem 6 is that E-MaSP is a generalization of the Maximum Coverage Problem. It is easy to see that when the overhead H is 0, then E-MaSP is equivalent to the Multiple Choice Knapsack Problem (MCKP), defined in Section III. Thus, E-MaSP can be viewed as a generalization of both MCKP and the Maximum Coverage Problem.

In the following we present an approximation algorithm for E-MaSP, which is a generalization of the greedy algorithm proposed in [10] for Budgeted Maximum Coverage. The latter

¹ $NP \not\subseteq DTIME(n^{O(\log \log n)})$, namely, that not all problems in NP can be solved in $O(n^{c \log \log n})$ for every constant c .

is a generalization of the Maximum Coverage Problem, in which the S_i subsets are not necessarily of the same size.

A key difference between Budgeted Maximum Coverage and E-MaSP is that even if we know which Phy-profiles should be used in the optimal schedule, we still have to find a valid assignment of PDUs to these Phy-profiles, since the PDUs have a non-negative size and non-negative profit for each Phy-profile. Therefore, an algorithm must also find an assignment of PDUs to the selected Phy-profiles. In addition, it must find a way to treat PDUs that were selected to Phy-profiles with low profit and low size. Such PDUs should be reconsidered for an “upgrade” to a Phy-profile with a higher profit and size.

The main idea behind the new algorithm is to define the residual profit of PDUs that have already been selected by the scheduler as the original profit minus the profit gained when the PDU was previously selected. The rationale behind this is that by selecting a PDU for one Phy-profile instead of the other we gain only the profit difference and pay only for the size difference. The algorithm iteratively picks up the best combination of a Phy-profile and a set of PDUs assigned to it, until the frame’s capacity $L = T \cdot C$ is reached. When this greedy selection ends, its output is compared to the most profitable schedule of a single Phy-profile. From these two schedules, the one with the highest profit is chosen.

Algorithm 6 (A greedy algorithm for E-MaSP):

- 1) Let S be an empty schedule.
- 2) While changes are made to S do:
 - a) If there is a Phy-profile ϕ that has already been selected to S and a PDU m such that (a) the residual size of assigning m to ϕ is not positive, and (b) the residual profit of assigning m to ϕ is positive, then add m to S and assign it to ϕ .
 - b) Otherwise, if there is a Phy-profile ϕ and a set of PDUs M such that (1) the residual size of assigning any PDU from M to ϕ is non-positive, (2) the residual profit of assigning any PDU from M to ϕ is positive, and (3) the total size of adding ϕ and M to S while assigning all the PDUs from M to ϕ is non-positive, then add ϕ and M to S and assign all the PDUs in M to S .
 - c) Otherwise, select a Phy-profit ϕ and a set of PDUs M , such that by adding ϕ to S and assigning all the PDUs in M to ϕ we achieve the highest residual density (the residual density is the residual profit divided by the residual size). Add ϕ to S and assign all the PDUs from M to S .
- 3) Find a maximum profit schedule with exactly one Phy-profile. If the profit of this schedule is greater than the profit of S , as found in the previous steps, return this schedule. Else, return S .

During each iteration of the algorithm at least one PDU is chosen. Therefore, the combination of this PDU and the Phy-profile to which it is chosen will not be considered again. Thus, Algorithm 6 has at most $n \cdot m$ iterations, where n is the number of PDUs and m is the number of Phy-profiles. Although finding a set with maximum density, as required in step 2(c), is NP-hard, such a set can be found by a pseudo polynomial algorithm. Since L is in the order of 1200, using a pseudo polynomial algorithm is reasonable. If L is too large, an $(1 + \epsilon)$ -approximation can be applied and it will only affect the approximation analysis by $(1 + \epsilon)$.

Theorem 7: Algorithm 6 is a $(\frac{2e-1}{e-1})$ -approximation² for E-MaSP. Moreover, this algorithm can be implemented in $O(m^2 \cdot n^2 \cdot L)$ where n is the number of PDUs, m is the number of Phy-profiles, and L is the size of the frame. The proof is very long, and is therefore out of the scope of this paper.

VII. SIMULATION STUDY

The purpose of this section is two-fold. First, to compare the performance of the MiSP algorithms presented in Section V. Second, to compare between the performance of the MaSP and the E-MaSP algorithms. In the following graphs, the performance is measured against the $T \cdot C$ lower bound, namely, the actual size of the Phy-profiles. We measure the wasted space used by every algorithm: we view the extra space as the difference between the actual size of Phy-profiles and $T \cdot C$ and the ratio between the extra space and $T \cdot C$ as the overhead (space wasted). During the simulation study, we consider a Gaussian distribution for the Phy-profile sizes, with a mean of $\frac{T \cdot C}{n}$. With this distribution, all Phy-profiles are roughly the same size. Simulations under different distributions, such as uniform, or under different means, show similar results. Note that typical values of C , T , and n are $35 \leq C \leq 70$, $13 \leq T \leq 26$, and $3 \leq n \leq 8$.

Figure 4(a) shows the effect of C on the performance, assuming $T = 20$ and $n = 6$. Figure 4(b) shows the effect of n on the performance, assuming $T = 20$ and $C = 60$. Figure 4(c) shows the effect of T on the performance, assuming $C = 60$ and $n = 6$, and Figure 4(d) shows the effect of $\frac{C}{T}$ on the performance, assuming $C = 2.5T$ and $n = 6$.

From Figure 4(a) we can see that when C increases, the performance of all the algorithms improve. This is because when C and the Phy-profiles are bigger, the matrix can be viewed as continuous, in which case the produced schedules are closer to optimal.

In Figure 4(b), all the algorithms have similar performance for $n = 2$. This is because the ways in which such a small number of profiles can be scheduled are severely limited. As n grows, the Best Fit algorithm has more combinations of Phy-profiles to choose from (e.g. $2^n - 1$ in the first iteration), while Increasing Size and Decreasing Size have only n in their first iteration, and Dual has only 1. The difference in the number of combinations between Best Fit and the rest of the algorithms stems from the loose structure of the scheduled items, i.e., the largest and smallest items can be scheduled next to each other. Increasing Size and Decreasing Size perform roughly the same, since they have the same number of combinations to choose from. Dual has only 1, and its performance decreases linearly with n . More specifically, when T and $\frac{TC}{n}$ are relatively large, the expected wasted space per item is $\frac{T}{2}$, and the total expected wasted space is $\frac{nT}{2TC} = \frac{n}{2C}$. In Figure 4(b), where $T = 20$ and $C = 60$, we get $\frac{n}{2 \cdot 60} = \frac{n}{120}$.

From Figure 4(c) we can see that when T increases, the performance of all the algorithms except Dual improves. The Dual algorithm’s performance declines when T grows because, for a high value of T , all items are relatively large, and are rounded to an exact multiple of T . When T is small, there is a greater chance for a Phy-profile size to be an exact multiple of T . When T increases, the expected space wasted due to

$$\frac{2 \cdot 2e-1}{e-1} \approx 2.58$$

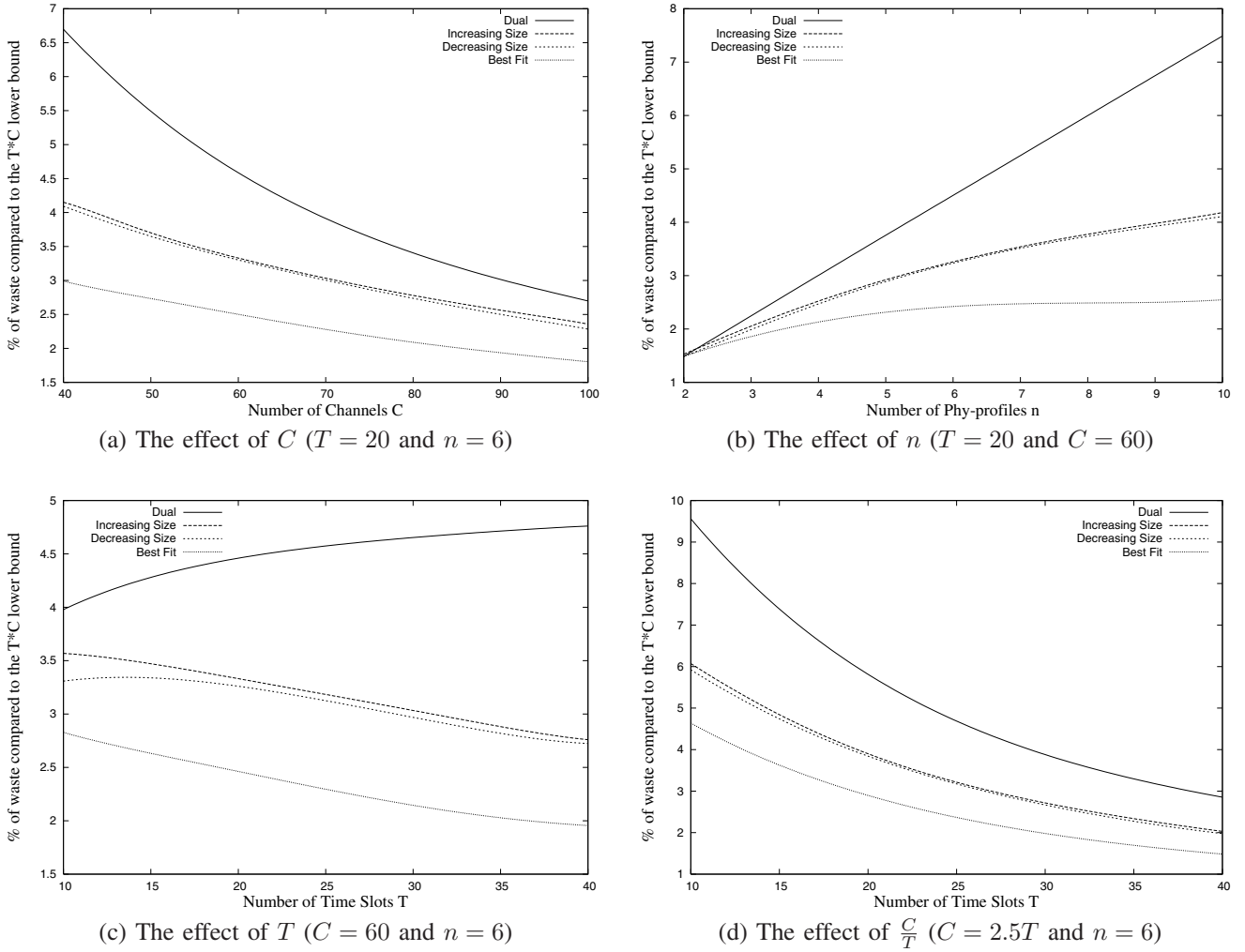


Fig. 4. The Performance of the Algorithms as a Function of Various Parameters.

this rounding is $\frac{T}{2}$ per item. Therefore, the total wasted space when T is relatively large converges to $\frac{nT}{2CT} = \frac{n}{2C}$. For the parameters in Figure 4(c), i.e., $C = 60$ and $n = 6$, we get $\frac{6}{2 \cdot 60} = 5\%$. For all the other algorithms, the improvement is due to the greater chance that two or more items can be fit into a single bin. Again, the number of combinations of Phy-profiles to choose from plays a critical role.

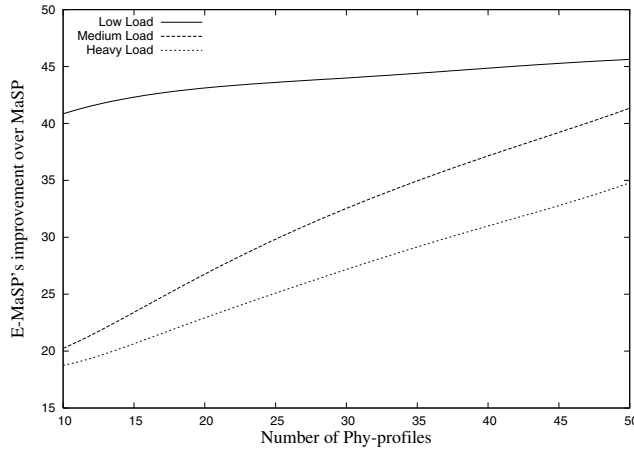
Finally, Figure 4(d) shows that the algorithms are highly scalable, because their performance improves when T and C increase, which indicates that MiSP becomes easier to solve. For the Increasing Size, Decreasing Size, and Best Fit algorithms, this improvement is attributed to both the increase in T (see Figure 4(c)) and the increase in C (see Figure 4(a)). For the Dual algorithm, the improvement stems from our earlier observation on Figure 4(c), where the performance converges to $\frac{n}{2C}$ when T increases.

We conclude this section by showing the advantage of E-MaSP over MaSP. Although the proposed MaSP algorithm was shown to be optimal, it can only assign each PDU to a single, pre-determined, Phy-profile. In contrast, E-MaSP can assign a PDU to every profile, but each assignment is associated with a different profit and cost. Therefore, the performance of E-MaSP is expected to be better than the performance of MaSP. For the following simulation study, we consider a Gaussian distribution of PDU sizes. In addition,

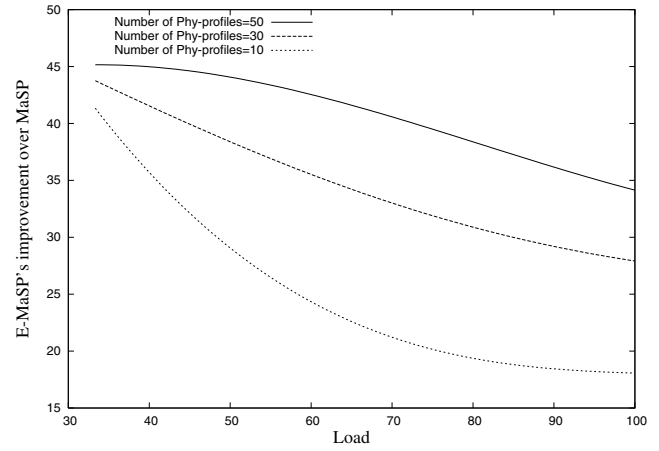
we set $C = 60$, $T = 20$, and a burst overhead of 5%. We assume that the MaSP scheduler chooses the most cost-effective Phy-profile for each PDU. That is, the Phy-profile for which the probability for successful transmission divided by the bandwidth cost is maximum.

Figure 5(a) shows the throughput improvement of E-MaSP over MaSP as a function of the number of Phy-profiles, for low (33%), medium (66%), and heavy (100%) loads. We can see that the improvement of E-MaSP increases when the number of Phy-profiles increases. The reason for this is that the MaSP algorithm might select two high profit PDUs for different Phy-profiles with close parameters. This increases the overhead penalty without increasing the likelihood for successful transmission. In contrast, E-MaSP tends to minimize the number of Phy-profiles when the effect on the loss probability is negligible. Therefore, MaSP is likely to accommodate more Phy-profiles in each frame, and, therefore, to increase the bandwidth overhead and the bandwidth loss due to fragmentation.

Figure 5(b) shows explicitly the correlation between the load and the E-MaSP's improvement. We can see that the improvement is much higher for low loads. This is because MaSP chooses the Phy-profiles without taking the load into account. In contrast, E-MaSP upgrades PDUs to more robust modulation schemes when there is enough available space.



(a) The effect of the number of Phy-profiles



(b) The effect of the load

Fig. 5. The Relative Performance Improvement of E-MaSP vs MaSP.

VIII. CONCLUSIONS

In this paper we defined and studied the intricate problem of downlink scheduling on an OFDMA channel. The main contributions of this paper are breaking the OFDMA scheduling problem into two more tractable problems, referred to as macro scheduling (MaSP) and micro scheduling (MiSP), analyzing the computational complexity of these two problems, and developing efficient algorithms for solving them.

We showed that MaSP can be solved optimally in pseudo-polynomial time by transforming it into the Multiple Choice Knapsack Problem, and that it can be approximated with an arbitrarily good precision in polynomial time. Then, we showed that MiSP is a more difficult problem than MaSP, in the sense that a $(1 + \epsilon)$ -approximation scheme for this problem cannot run in polynomial time in $\frac{1}{\epsilon}$. Nevertheless, we presented several efficient algorithms for MiSP.

We also presented an extended version of the macro scheduling problem, called E-MaSP. In this version, the association between a PDU and its Phy-profile is determined on-line by the scheduler, as part of the macro scheduling. We showed that E-MaSP has an advantage over MaSP especially when the channel is underloaded, because in such a case the probability for a successful transmission of a PDU using the Phy-profile determined by E-MaSP is greater than when the Phy-profile is determined in advance regardless of the load in the channel. However, the performance gains achieved by E-MaSP comes at the expense of higher running time complexity.

One of our most important results was that the Best Fit algorithm performs better than the other MiSP algorithms. With typical parameters, the overhead of Best Fit is only 2.5% of the total scheduling space.

REFERENCES

- [1] A. Azgin and M. Krunz. Scheduling in wireless cellular networks under probabilistic channel information. In *ICCCN*, pages 89–94, 2003.
- [2] Y. Ben-Shimol, I. Kitroser, and Y. Dinitz. Two-dimensional mapping for wireless ofdma systems. *IEEE Trans. Broadcasting*, 52(3):388–396, Sept 2006.
- [3] R. Cohen and L. Katzir. Computational analysis and efficient algorithms for micro and macro ofdma scheduling. Technical Report CS-2007-02, Technion-Israel Institute of Technology, April 2007.
- [4] 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, 2007.

- [5] A. J. Goldsmith and S. G. Chua. Variable-rate variable-power MQAM for fading channel. *IEEE Transactions on Communications*, 45(10), Oct. 1977.
- [6] D. S. Hochbaum, editor. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Co., Boston, MA, USA, 1997.
- [7] Institute of Electrical and Electronics Engineers Inc. IEEE standard for local and metropolitan area networks – part 16: Air interface for fixed broadband wireless access systems, 2004.
- [8] A. Israeli, D. Rawitz, and O. Sharon. On the complexity of sequential rectangle placement in ieee 802.16/wimax systems. In *ESA*, pages 570–581, 2007.
- [9] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.
- [10] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70(1):39–45, 1999.
- [11] D. Kivanc, G. Li, and H. Liu. Computationally efficient bandwidth allocation and power control for OFDMA. *IEEE Transactions on Wireless Communications*, 2(6), Nov. 2003.
- [12] S. Lu, V. Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on Networking*, 7(4):473–489, 1999.
- [13] S. Nanda, K. Balachandran, and S. Kumar. Adaptation techniques in wireless packet data services. *IEEE Communications Magazine*, 38(1), Jan. 2000.
- [14] S. Shakkottai and R. Srikant. Scheduling real-time traffic with deadlines over a wireless channel. *ACM/Baltzer Wireless Networks Journal*, 8(1):13–26, Jan. 2002.
- [15] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round-robin. *IEEE/ACM Transactions on Networking*, 4(3):375–385, 1996.
- [16] G. Song and Y. Li. Cross-layer optimization for OFDM wireless networks part I: Theoretical framework. *IEEE Transactions on Wireless Communications*, 4(2), Mar. 2005.
- [17] D. Tse and S. Hanly. Multi-access fading channels: Part I: Polymatroid structure, optimal resource allocation and throughput capacities. *IEEE Transactions on Information Theory*, 44(7):2796–2815, Nov. 1998.
- [18] P. Viswanath, D. Tse, and R. Laroia. Opportunistic beamforming using dumb antennas. *IEEE Transactions on Information Theory*, 48(6), June 2002.
- [19] I. Wong, Z. Shen, J. Andrews, and B. L. Evans. A low complexity algorithm for proportional resource allocation in OFDMA systems. *IEEE International Signal Processing Systems Workshop*, Oct 2004.