sGSA: A SDMA-OFDMA Greedy Scheduling Algorithm for WiMAX Networks

Anatolij Zubow[a], Johannes Marotzke[a], Daniel Camps-Mur[b], Xavier Perez Costa[b],

[a] *Humboldt Universität zu Berlin, Unter den Linden 6, Berlin, Germany*
[b] *NEC Laboratories Europe, Network Research Division, Kurfürsten-Anlage 36, Heidelberg, Germany*

Abstract

Future wireless networks need to address the predicted growth in mobile traffic volume, expected to have an explosive growth in the next five years mainly driven by video and web applications. Transmission schemes based on Orthogonal Frequency Division Multiple Access (OFDMA) combined with Space Division Multiple Access (SDMA) techniques are key promising technologies to increase current spectral efficiencies. A Joint SDMA-OFDMA system has to allocate resources in time, frequency and space dimensions to different mobile stations, resulting in a highly complex resource allocation problem. In contrast to related work approaches, in this paper we take a comprehensive view at the complete SDMA-OFDMA scheduling challenge and propose a SDMA-OFDMA Greedy Scheduling Algorithm (sGSA) for WiMAX systems. The proposed solution considers feasibility constraints in order to allocate resources for multiple mobile stations on a per packet basis by using i) a low complexity SINR prediction algorithm, ii) a cluster-based SDMA grouping algorithm and iii) a computationally efficient frame layout scheme which allocates multiple SDMA groups per frame according to their packet QoS utility. A performance evaluation of the proposed sGSA solution as compared to state of the art solutions is provided, based on a comprehensive WiMAX simulation tool.

Key words: SDMA, OFDMA, 802.16, QoS, Scheduler

1. Introduction

Mobile wireless networks are faced with an ever increasing demand for higher data rates, expected to grow exponentially in the next years due to the smartphone market success and its corresponding set of bandwidth hungry applications, specially video and web. Transmission schemes based on Orthogonal Frequency Division Multiple Access (OFDMA) and Multi-User Multiple Input Multiple Output (MU-MIMO) techniques are promising technologies to increase current spectral efficiencies. The performance of MU-MIMO relies on the precoding capability of the transmitter and, in order to take full advantage of such a technique, a Base Station (BS) needs full Channel State Information (CSI) on the properties of the communication link to each mobile station (MS). A well-known MU-MIMO mode is Space-Division Multiple Access (SDMA) which can be used in the downlink direction to allow a group of spatially separable MSs to share the same time/frequency resources.

SDMA-OFDMA systems have to allocate resources in time, frequency and space dimensions to different MSs resulting in a highly complex resource allocation problem. Therefore, in order to reduce the complexity of such a system the overall problem can be divided into several subproblems, each conquering the different dimensions separately [1]. However, these subproblems still hold a high level of complexity and thus, suboptimal strategies are often proposed in the literature [2].

In contrast to related work approaches, in this paper we take a comprehensive view at the complete SDMA-OFDMA scheduling challenge and propose a SDMA-OFDMA Greedy Scheduling Algorithm (sGSA) for WiMAX systems. Specifically, we focus on the WiMAX IEEE 802.16-2009 [3] standard, which provides scalable OFDMA capabilities as well as additional MIMO PHY features. According to our proposed *sGSA* approach the SDMA-OFDMA scheduling solution is divided into three main steps:

1. SINR Prediction - Estimation of the SINR experienced by each MS in an SDMA group
2. SDMA grouping - partitioning of the MSs into a set of SDMA groups, where spatially compatible MSs are arranged in the same SDMA group
3. Frame partitioning - partitioning of an OFDMA frame into multiple containers each representing a single SDMA group

This paper is an extension of the work from the authors in [4] which is currently under submission. The paper at hands extends [4] by i) designing a low complexity SINR prediction algorithm to reduce the computational load of the overall SDMA-OFDMA scheduling solution, ii) evaluating the computational load reduction and corresponding SINR prediction error impact, iii) recalculating all results in the performance evaluation section accordingly and iv) considering additional factors in the system performance evaluation as number of operations, signaling overhead and buffer size limitations.

The rest of the paper is structured as follows. Sections 2 and 3 provide the IEEE 802.16-2009 basic elements consid-

Email addresses: zubow@informatik.hu-berlin.de (Anatolij Zubow), marotzke@informatik.hu-berlin.de (Johannes Marotzke), Daniel.CampsMur@nw.neclab.eu (Daniel Camps-Mur), Xavier.Perez-Costa@nw.neclab.eu (Xavier Perez Costa)

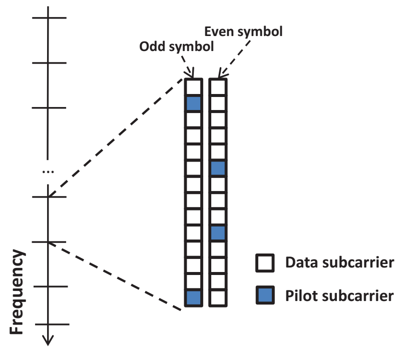


Figure 1: DL PUSC cluster structure

ered in our solution and summarize the most relevant work in the area. Next, Section 4 formalizes the problem description and formulates the SDMA-OFDMA scheduling as an optimization problem. Section 5 describes our proposed sGSA solution along with the designed algorithms required. In Section 6 a performance evaluation is conducted through simulations. Finally, Section 7 concludes the paper.

2. SDMA-OFDMA support in IEEE 802.16e

The focus of this paper is an IEEE 802.16e TDD OFDMA system where Base Stations (BSs) access the channel by means of Downlink (DL) and Uplink (UL) subframes. Each subframe is split into a set of OFDMA symbols in time and logical subchannels in frequency, where the mapping between physical subcarriers and logical subchannels is determined by a *permutation scheme*. 802.16e supports different permutation schemes being *Partial Usage of SubCarriers (PUSC)* the most widely used. PUSC is a distributed permutation scheme that achieves a frequency diversity gain by mapping a set of subcarriers randomly spreaded across the available bandwidth into each logical subchannel. In the rest of the paper we will consider PUSC as the permutation scheme used by our system.

PUSC works in the following way. First, the available physical subcarriers are divided into *clusters* of 14 contiguous subcarriers, containing each cluster 2 pilot subcarriers and 12 data subcarriers (as depicted in Figure 1). The pilot subcarriers in each cluster are used by a receiver to estimate the channel between transmitter and receiver and thus demodulate the received data subcarriers. Then, in order to achieve the intended diversity gain, physical clusters are randomly renumbered, using a random seed local to each BS, and sequentially mapped to each logical subchannel. Each logical subchannel is hence composed of two PUSC clusters.

Leveraging PUSC, a BS can implement beamforming or SDMA in a transparent way for the Mobile Stations (MSs) by simply using *dedicated pilots*. When beamforming or SDMA is utilized a BS uses a set of beamforming weights or a pre-coding matrix, W, in order to shape its antenna pattern towards its intended receivers. Thus, if the same beamforming weights are used for the pilot subcarriers and the data subcarriers within each cluster, a receiver will simply perceive the beamformed transmission as a modified channel, and will transparently demodulate the beamformed data. In order to do so, a BS signals the use of dedicated pilots to a MS, such that the MS only uses the pilot subcarriers belonging to its own data allocation to demodulate the received data. Given the use of dedicated pilots and the cluster structure in PUSC, 802.16e BSs must use the same beamforming weights or pre-coding matrix for each PUSC cluster (i.e. 14 contiguous subcarriers). However, BSs are allowed to use different beamforming weights for different PUSC clusters.

Figure 2 illustrates how SDMA can be implemented over an 802.16e TDD OFDMA DL subframe. Note that the initial MAP messages used to signal the data allocations to each MS in the DL subframe are transmitted omnidirectionally. However, successive time frequency allocations within the OFDMA frame can be shared among different MSs, if these are spatially separable. For instance, we can see in Figure 2 how the BS chooses to allocate the same time frequency resources for MS1 and MS2 because they can be separated in the space domain. The lower part of Figure 2 illustrates how the individual PUSC clusters randomly map to each logical subchannel, and how the BS, by applying a different pre-coding matrix to each cluster, is able to concurrently steer different spatial beams in different time frequency regions within the OFDMA frame. Notice that in order to allocate the same time frequency resource to different MSs, the BS simply adds for each MS an entry in the MAP pointing to the intended region.

In order to derive the pre-coding matrix, $W(f)$, to be applied to each cluster the BS needs Channel State Information (CSI) from each MS. Indeed, in a wideband channel like the one considered for IEEE 802.16e systems the spatial channel is typically dependent on frequency, i.e. $H(u, f)$ where u indicates the MS and f the subcarrier. Several options are available in 802.16e for the MSs to provide the required feedback to the BS. In this paper we will assume *UL sounding* as the employed feedback mechanism and we describe it next.

In an 802.16e TDD system, BSs and MSs can run a calibration protocol that ensures that the downlink and uplink channels are symmetric. Therefore, the BS can estimate the downlink channel towards each MS by observing the distortion experienced by sounding signals sent by the MSs in the uplink. For this purpose, the BS allocates an UL sounding region in the uplink subframe, as depicted in Figure 2, and indicates in the UL MAP which MS should transmit sounding signals on which frequency band within this region. If a MS moves, the BS can command a periodic transmission of sounding signals to be able to track the varying channel. Finally, in order to reduce signaling overhead, several MSs can be multiplexed within the same UL Sounding region, e.g., making use of *decimation* whereby each MS only sends a sounding signal every Dth subcarrier.

3. Related Work

In this section we discuss related work relevant to the contributions presented in this paper. This section is divided in

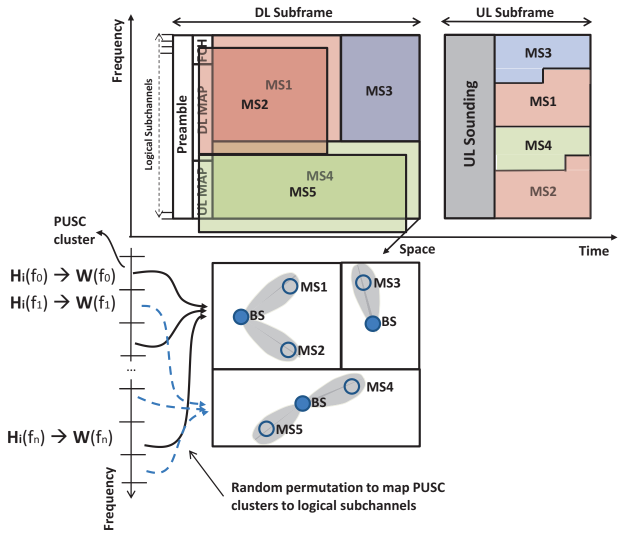


Figure 2: SDMA over 802.16e using PUSC

four subsections: i) SDMA Transmitter techniques, ii) SDMA Grouping Algorithms, iii) SDMA Grouping Metrics, and iv) Full SDMA-OFDMA Solutions.

3.1. SDMA Transmitter Techniques

In SDMA multiple antennas are used to send signals to several spatially separated MSs using the same time-frequency resources. For this purpose knowledge of the spatial signatures of each MS is required, that can be utilized in different ways. Next, we describe two major SDMA transmitter techniques: i) *Dirty Paper Coding* (DPC), and ii) *Beamforming*.

First, DPC is a technique that using knowledge about the interference at the transmitter, pre-codes the transmitted signal in order to cancel the effect of such interference. Instances of DPC techniques include Costa precoding [5], Tomlinson-Harashima precoding [6] and the vector perturbation technique [7].

Second, beamforming allows the separation of multiple MSs in space by means of allocating a beamforming direction to a given MS while treating other MSs as noise. Common beamforming techniques are the Zero-Forcing technique [1] that introduces nulls in the directions of the interferers, and techniques that maximize the receiver SIR like Minimum Variance Distortionless Response, Maximum SIR or Minimum Mean Square Error [8].

Compared to DPC, beamforming techniques are suboptimal [9]. However, beamforming has a lower computational cost than DPC, and therefore is a valuable alternative in practical systems. In addition, under some circumstances beamforming can achieve the same capacity as DPC [10].

3.2. SDMA Grouping Algorithm

Randomly selecting the MSs that are multiplexed in space using SDMA may result in a high interference between MSs. Therefore, in practice a *grouping algorithm* is used that groups MSs into suitable subsets that can be effectively separated in space.

The task of a grouping algorithm is thus to determine an efficient SDMA group with a complexity as low as possible. Next, we describe some of the SDMA grouping algorithms commonly used in the state of the art. The *Best Fit Algorithm* (BFA) [2] is a greedy algorithm that constructs an SDMA group starting with the MS that experiences the lowest SNR/SINR, and sequentially extending this group by admitting the MS providing the highest increase of a given *grouping metric* (refer to Section 3.3). Once the group reaches its target size, or no more MSs can increase the grouping metric, the SDMA group is considered complete [1]. Another closely related algorithm with even lower complexity is the *First Fit Algorithm* (FFA). Its only difference compared to the BFA, is that instead of admitting the MS providing the highest increase of a given grouping metric, FFA adds to the SDMA group the first MS that provides any increase in the grouping metric [2].

3.3. SDMA Grouping Metric

As mentioned before, grouping algorithms require a *grouping metric* in order to compare candidate SDMA groups with each other. In general, a grouping metric makes use of *Channel State Information* (CSI) in order to map the characteristics of the spatial channels of the MSs to a scalar value that quantifies how efficiently these MSs can be separated in space [1]. The most commonly used group metrics are *group capacity* and *group minimum SINR*. The former considers the Shannon-Hartley capacity of an SDMA group as the spatial compatibility metric, and the latter returns the lowest SINR of an MS in a given SDMA group. In order to compute the previous metrics though, the actual beamforming weights or the power allocation have to be considered which involve complex vector/matrix operations. Therefore, the performance achieved by such metrics comes at the expense of an increased complexity [11]. In order to decrease complexity, grouping metrics that are based only on the spatial correlation and on the channel gains of the MSs, involving much simpler vector/matrix operations, have been proposed [12, 2, 1].

Finally, the previous grouping metrics usually consider a narrow-band channel, however in a wideband or frequency selective channel like 802.16e the characteristics of the spatial channel cannot longer be described by a single channel transfer matrix. Due to frequency selectivity the channel transfer matrix is different for different subcarriers; hence the spatial compatibility among MSs in an SDMA group depends on the used subcarriers. In order to address this fact, in this paper we consider PHY abstraction models like the Exponential Effective SIR Mapping (EESM[1]) that allow to collapse a set of per-subcarrier SINR measurements into an effective SINR measurement, that can then be used to calculate a grouping metric like group capacity. The idea of using EESM as a multi-carrier

[1]IEEE EESM Reference, `www.ieee802.org/16/tge/contrib/C80216e-05_141r3.pdf`

grouping metric is not far-fetched, as the authors of [13] concurrently to the work presented in this paper chose to do the same thing.

3.4. Full SDMA-OFDMA Solutions

To the best of the authors' knowledge only a few proposals in the state of the art evaluate the performance of SDMA in OFDMA-based systems like 802.16e. Next, we discuss these proposals.

Nascimento et al. [13] proposed a joint utility packet scheduler and SDMA-based resource allocation scheme for 802.16e. Similar to the work presented in this paper, [13] uses the EESM compression method to capture frequency selective channels and build SDMA groups consisting of spatially uncorrelated MSs. The proposed scheduling solution therein assigns MSs to beams, using an FFA-like grouping algorithm and spatial correlation (computed based on the EESM SINR) as a grouping metric. In addition, QoS is achieved through a prioritized assignment based on a utility framework. In order to reduce complexity, the authors in [13] reduce the SDMA capabilities to one third of the OFDMA frame leaving the rest for non-SDMA transmissions. Performance is evaluated under the assumptions of a full queue traffic model.

Similar to the previous work, Yao et al. [14] evaluated the MAC performance of an OFDMA system based on the superseded 802.16a-2003 standard, which suggests a frame layout separated in Adaptive Antenna System (AAS) and non-AAS capable zones. The scheduling solution presented therein prioritizes MSs based on their channel conditions and packet delays. Specifically, MSs are grouped according to their channel conditions, whereas, within a group, MSs are prioritized according to packet deadlines. Depending on the intra-beam interference MSs are assigned to a beam or moved to the regular non-AAS zone. The improvement achievable when using AAS is evaluated for FTP and VoIP services.

The main difference between these approaches and the work presented in this paper is the assumption of an idealized LOS channel allowing beamsteering based on the estimation of the angle of direction of an MS. Instead, the work presented in this paper explicitly models the wireless channel coefficients, taking advantage of the LOS component as well as possible scattering and multi-path effects.

4. Problem Statement

This section describes the physical and QoS models used throughout this paper and formulates our suggested SDMA-OFDMA scheduling solution as an optimization problem.

4.1. PHY Model

We consider the DL of a single BS with a maximum transmit power that is being equally distributed among the served MSs. The BS is equipped with an Uniform Linear Array (ULA) and there are K single-antenna MSs associated with the BS (Figure 3). We consider OFDMA where the channel bandwidth is divided into S orthogonal subcarriers. In addition, physical subcarriers are mapped to logical subchannels using PUSC. Regarding CSI knowledge in the BS, we assume that for each MS u the BS knows the channel transfer matrix ($H_{u,s}$) of every Dth subcarrier, hence resulting in $\frac{S}{D}$ $H_{u,s}$ coefficients per MS[2]. In addition, we assume that the BS has full CSI on each $H_{u,s}$, i.e. there are no estimation errors. The required channel coefficients are generated with the WIM simulator [15], where only low to medium-speed mobility is considered.

Next, we introduce the format used for the channel transfer matrix, the beamforming weights, and the SINR calculation. For an MS u ($u = 1, \ldots, K$) the channel coefficient $h_{i,u}$ denotes the sampled frequency response of the channel between the BS antenna element i and the receiver antenna of the u-th MS as depicted in Figure 3. Thus, the channel coefficients belonging to a given MS u can be grouped into a column vector $M \times 1$, i.e. $H_u = [h_{1,u}, h_{2,u}, ..., h_{M,u}]^{\mathrm{T}}$, where M is the number of available antennas at the BS and H_u holds all spatial correlations and multi-path effects of the MS u channel. The BS uses precoding weights to adjust the transmitted signal in order to mitigate the propagation effects of the channel and to control the interference among MSs in a SDMA group. Therefore, every channel coefficient $h_{i,j}$ is associated with a beamforming weight $w_{i,j}$, and $W_u = [w_{1,u}, w_{2,u}, ..., w_{M,u}]^{\mathrm{T}}$ is the normalized weight vector for a given MS u.

The effective channel gain at the MS can then be computed as $||W_u^{\mathrm{H}} \cdot H_u||_2^2$, where $(\cdot)^{\mathrm{H}}$ denotes the conjugate transpose of a vector/matrix. Notice, that multiple MSs being served during the same time-frequency resource cause interference upon each other (intra-cell interference). Therefore, given a certain frequency resource $s = 1, \ldots, S$, and the corresponding channel coefficients, the SINR of MS u can be computed as [1]:

$$\gamma_{u,s} = \frac{\overline{R}_{u,s} \cdot ||W_{u,s}^{\mathrm{H}} \cdot H_{u,s}||_2^2}{\sigma^2 + \sum_{v=1, u \neq v}^{M} \overline{R}_{u,s} \cdot ||W_{v,s}^{\mathrm{H}} \cdot H_{u,s}||_2^2} \tag{1}$$

where $\overline{R}_{u,s}$ represents the average received power[3] at MS u on subcarrier s. We assume that the BS allocates the same power to all subcarriers.

In this paper, we assume that the BS uses the *Minimum Mean Square Error* technique (MinMSE [8]) in order to compute beamforming weights. MinMSE works in the following way. For a given frequency resource unit s the matrix $\mathbf{H}_s$ contains in each u-th column the channel coefficients ($H_{u,s}$) of the u-th MS. Let $\mathbf{R_{ss}} = \mathbf{H}_s \cdot \mathbf{H}_s^H$ be the autocorrelation matrix of all MSs, and $\mathbf{R_{nn}}$ the noise correlation matrix. Then, the weight matrix $\mathbf{W_s}$ containing the beamforming weights for every u-th MS in an SDMA group, can be computed as:

$$\mathbf{W}_s = (\mathbf{R_{ss}} + \mathbf{R_{nn}})^{-1} \cdot \mathbf{H}_s, \tag{2}$$

where the u-th column in $\mathbf{W}_s$ contains the weights for the u-th MS for the given frequency resource unit s.

For comparison reasons, we will also consider in this paper the possibility of beamforming to a single MS instead of

[2] In simulations D was set to 24, i.e. the number of $H_{u,s}$ per MS was 30.

[3] Path loss and Shadowing are included in $\overline{R}_{u,s}$ and not in H_u.

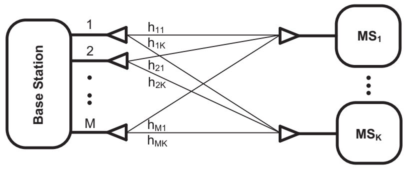


Figure 3: System model: DL of a single BS equipped with Antenna Array and multiple single-antenna MSs.

using SDMA. In this case, we assume that the BS uses *Maximum Ratio Combining* (MRC [16]) to calculate the required beamforming weights:

$$W_{u,s}^{\mathrm{H}} = \frac{H_{u,s}^{\mathrm{H}}}{||H_{u,s}||_2^2} \tag{3}$$

In case of MRC beamforming, the achieved SNR of a single MS u is also given by Equation 1, using the MRC beamforming weights and ignoring the intra-cell interference term. The denominator of Equation 3 normalizes the weights to unity so that the average total transmit energy remains the same. The same normalization is performed for the weights given by the MinMSE technique.

Finally, the set of per-subcarrier SINR values can be collapsed into a single equivalent SINR value for each MS making use of the EESM mapping technique.

4.2. Base Station Architecture and QoS Model

We consider an *offline* BS architecture as formally defined in [17]. In an offline architecture there are two main building blocks that cooperate in order to maximize the utility of the scarce radio resources; these blocks are the *QoS Scheduler* and the *SDMA-OFDMA* scheduler. The task of the QoS Scheduler is to select from the higher layer packets available in each flow's buffer a *candidate* list of packets to be transmitted in the next DL subframe. In addition, the QoS scheduler tags each individual packet P_i with a *utility* value u_i. Typical QoS schedulers that can be accommodated in this architecture are Proportional Fair or Deficit Round Robin [18]. Thus, after receiving a set of candidate packets from the QoS Scheduler, the task of the SDMA-OFDMA scheduler is designing the SDMA-OFDMA frame layout, i.e. assign the time, frequency and space resources, in order to maximize the amount of utility carried in the SDMA-OFDMA frame. The considered offline BS architecture is illustrated in Figure 4.

4.3. Problem Formulation

We now introduce the DL SDMA-OFDMA scheduling problem considered in this paper. As previously stated, the main objective of a DL SDMA-OFDMA scheduler is to maximize the total utility carried in the DL subframe. However, optimally assigning time, frequency and space resources is a very complex task to be efficiently implemented in practice. Therefore, in order to reduce complexity we divide the overall optimization problem into two independent major tasks:

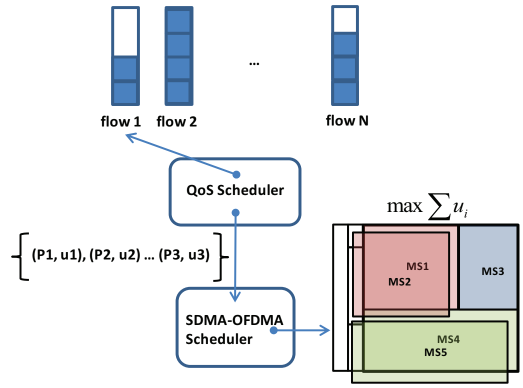


Figure 4: QoS architecture considered in the BS.

1. *SDMA group formation*: we look at the problem of creating spatially compatible SDMA groups, where each group consists of a set of stations which can be separated in the space domain.
2. *OFDMA frame construction*: given a set of SDMA groups, we consider the problem of scheduling the subset of these groups within the OFDMA DL subframe that maximizes the overall utility carried in the OFDMA frame.

SDMA group formation

The objective of this problem can be formalized in the following way:

Instance: A set U of MSs having buffered packets as delivered by the QoS scheduler (Figure 4).

Objective: Find the optimal partition of U into a set $\mathcal{G} = \{G_1, G_2, ..., G_m\}$ of non-overlapping and non-empty sets of MSs that cover all of U, i.e. $\bigcup_{i=1}^{m} G_i = U$ and $G_i \cap G_j = \emptyset, \forall i \neq j$, such that the average group capacity is maximized. Here $G_i = \{u_{i,1}, u_{i,2}, ...\}$ represents a single SDMA group where $u_{i,j}$ represents the j-th MS in group i. Moreover, γ_{u,G_i} is the SINR of MS u in group G_i.

This optimization problem can be formulated as follows:

$$\mathcal{G} = \arg\max_{\mathcal{G}} \left\{ \frac{1}{|\mathcal{G}|} \sum_{i=1}^{|\mathcal{G}|} \sum_{u \in G_i} \log(1 + \gamma_{u,G_i}) \right\} \tag{4}$$

subject to:

(I) A maximum group size limited to M, i.e. the number of available antennas at the BS:

$$|G_i| \leq M, \forall G_i \in \mathcal{G} \tag{5}$$

(II) A minimum SINR constraint, that makes sure that each MS in an SDMA group can be served at least on the lowest Modulation and Coding Scheme (MCS):

$$\gamma_{u,G_i} \geq \gamma_{\mathrm{MIN}}, \forall G_i \in \mathcal{G}, \forall u \in G_i \tag{6}$$

Notice in Equation 4 that the target metric is divided by the number of SDMA groups, $|\mathcal{G}|$, in order to favor solutions with a fewer number of groups. The reason is that solutions with fewer groups tend to be more efficient because different groups can only be scheduled in different time-frequency resources.

Complexity: The optimal solution for this group formation problem was shown to be NP-complete in [2].

OFDMA frame construction

Before formalizing this problem, we introduce an additional design decision that consists of forcing an allocated SDMA group to span always an integer number of columns in the OFDMA frame. The main reason for this decision is that it enables the previous SDMA group formation problem to be agnostic to the actual frequency region where the data will be transmitted, i.e. the required SINR values in Equation 4 can be derived assuming that an allocation spans the whole frequency range. In addition, this design decision turns the packing of SDMA groups within an OFDMA frame into a one-dimensional packing problem instead of a two-dimensional packing problem. In Section 6 we will show that this simplification allows to design well performing practical solutions. Thus, our OFDMA frame construction problem is formalized in the following way:

Instance: A partition of the set U of MSs into a set $\mathcal{G}$ of non-overlapping and non-empty sets of MSs that cover all of U.

Objective: The packing area for each SDMA group $G_i \in \mathcal{G}$ in the OFDMA frame is controlled by an associated *container*. Find the optimal width $s_i \in [0..DL_{slots}]$ for each container such that the total utility, $u_{(\cdot)}$, carried by the OFDMA frame is maximized, i.e. $\overline{S}^* \in [(s_0, s_1, ..., s_{|\mathcal{G}|})$, where DL_{slots} is the total number of time slots in the DL frame. Here, the first container of width s_0 represents the space reserved for the DL-MAP to signal the data bursts scheduled within the frame.

This optimization problem can be formalized as follows:

$$\{\overline{S}^*\} = \arg\max_{\{\overline{S}\}} \left\{ \sum_{i=1}^{|\mathcal{G}|} \sum_{u \in G_i} \sum_{\substack{p_u \in P_u^{'} \\ P_u^{'} \subseteq P_u}} u_{(\cdot)}(p_u) \right\} \tag{7}$$

subject to:

(I) A container layer space constraint that makes sure that the total size of packed data bursts $P_u^{'}$ in each container layer does not exceed the available space:

$$\sum_{p_u \in P_u^{'}} \mathrm{p2burst}(p_u, \gamma_{u,G_i}) \leq \overline{S_i} \times B \tag{8}$$

where γ_{u,G_i} represents the SINR of MS u in group G_i, $\overline{S_i}$ represents the width of the i-th container, and B represents the number of frequency resources in one time slot (equal to the number of subchannels). The function $\mathrm{p2burst}(\cdot)$ calculates the burst size in slots for a given packet size and SINR value.

(II) A DL-MAP space constraint that makes sure that there is sufficient space for the DL-MAP, so that all packed data bursts $P_u^{'}$ in each layer of each container can be properly signaled. For this purpose, the container at position zero is reserved for the DL-MAP[4].

$$\left\lceil \frac{F_0 + \sum_{u \in U}(F_1 + \sum_{p=1}^{|P_u^{'}|} F_2)}{\mathrm{SC}} \right\rceil \cdot \mathrm{MRep} \leq \overline{S_0} \times B \tag{9}$$

where F_0, F_1, F_2, SC and MRep represent various MAP overheads (Section 6, Table 1).

Complexity: The optimization problem described by Equation 7-9 is a variant of the Multiple-Choice Nested Knapsack Problem [19], because we have to obtain the subset of MPDUs for each layer in each SDMA container that maximizes the utility carried in the OFDMA frame. This problem has been proved to be NP-complete [20].

5. SDMA Greedy Scheduling Algorithm (sGSA)

In this section we present the design of our DL SDMA-OFDMA Greedy Scheduling Solution (*sGSA*). As described in the previous section, in order to implement the assignment of time, frequency and space resources in an efficient way, sGSA addresses sequentially the following two sub-problems: i) first, the problem of creating spatially compatible SDMA groups, and ii) second the problem of scheduling a given set of SDMA groups within an OFDMA frame.

In order to address the problem of creating spatially compatible SDMA groups, sGSA contains three different modules:

1. The *SINR Predictor Algorithm* which allows to obtain an estimate of the SINR experienced by each MS in a candidate SDMA group at a lower complexity than state-of-the-art approaches.
2. The *Cluster-based Grouping Algorithm (CBA)*, that is a heuristic method to solve the SDMA group formation problem described in Equation 4.
3. The *Adaptive Modulation and Coding (AMC) module*, that selects the appropriate MCS to be assigned to each MS within an SDMA group.

Finally, given a set of SDMA groups, sGSA solves the OFDMA frame construction problem described in Equation 7 making use of an *SDMA-OFDMA packing algorithm*.

The interactions between the different modules that compose sGSA are depicted in Figure 5. Next, we provide a detailed description of each module.

5.1. SINR Prediction Algorithm

As described in Section 3.3 we can find in the state of the art grouping metrics like *group capacity* which provide an accurate representation of the interference that each MS in an SDMA group is subject to but involve complex vector/matrix

[4] Packing data bursts below DL-MAP is not allowed in our model.

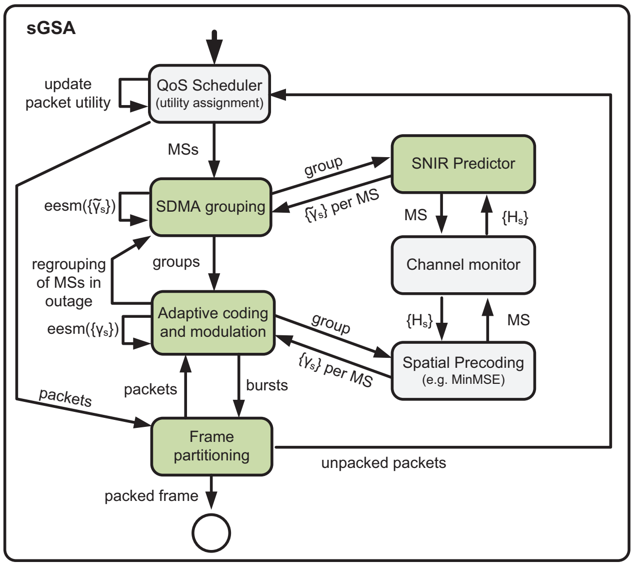


Figure 5: Overview of the proposed sGSA scheduling solution.

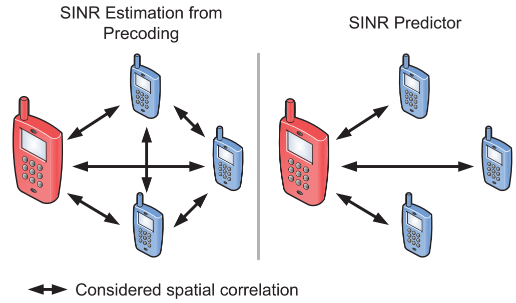


Figure 6: Illustration of two methods for calculating the SINR of a user: (i) from the actual precoding matrix derived by any arbitrary MIMO technique, (ii) using the SINR prediction algorithm.

operations required to compute the actual pre-coding matrixes (beamforming weights). In addition, low complexity grouping metrics are also available that, however, do not accurately measure interference within an SDMA group. In this section we present our SINR Predictor which is a heuristic method to resolve this trade-off.

In particular, the SINR Predictor is a low-complexity algorithm for predicting the SINR value of each MS in a given SDMA group without computing the precoding matrix. Notice that as explained in Section 3.3, if per-subcarrier SINRs are known, then the EESM mapping technique can be used to derive a single equivalent SINR to compute metrics like *group capacity* or *group minimum SINR*. The key question to resolve thus is: how to obtain an accurate estimation of the per-subcarrier SINR without performing expensive pre-coding matrix computations.

Figure 6 illustrates the basic difference between calculating the SINR using a precoding matrix and our proposed SINR Prediction algorithm. When calculating the SINR of a MS (red) within a SDMA group the proposed SINR Prediction method considers only the spatial correlation between the MS of interest and the other MSs within that group. Hence, unlike the case of using a precoding matrix, in the SINR Predictor case the spatial correlations between the other MSs within that group are not considered. Therefore, the SINR Predictor is a lower complexity solution but introduces an estimation error in the per-subcarrier SINR.

In particular, the SINR Predictor estimates the SINR for any SDMA group G and MS $i \in G$ on frequency resource block b in the following way:

$$\tilde{\gamma}_b(u,G) = SNR_{\mathrm{MRC}}(u) \cdot \begin{cases} 1-\chi_u^b, \text{ if } \chi_u^b \leq 1 \\ \beta, \text{ else} \end{cases} \tag{10}$$

where $\chi_u^b = \sum_{u' \in G, u' \neq u} \tau_{u,u'}^b$, i.e. the sum of the squared spatial correlations between MS u and the other MSs in group G on frequency resource block b.

The idea of the SINR Predictor is to consider an initial SINR for a MS u in an SDMA group G and frequency resource block b that is obtained assuming a non-SDMA transmission (MRC beamforming, refer to Equation 3). Then, this initial SINR estimation is corrected by the expected intra-SDMA group interference caused by the other MSs in the group. In particular, the applied correction factor is either χ_u^b if $\chi_u^b \leq 1$, or a constant factor β, which in our case we set to 13.5 dB (0.0443 in linear), if $\chi > 1$. The reason for the constant factor is that our simulations showed that for $\chi > 1$, the MS's SINR was on average 13.5 dB below its SNR_{MRC} value. This applied constant factor though could be tuned in order for the SINR Predictor method to be applied in other scenarios.

In addition, in our system a minimum threshold for packet reception of 5 dB is required to transmit using the lowest MCS. Therefore, a MS u having a $SNR_{\mathrm{MRC}}(u) < 18.5$ and $\chi > 1$ will be considered to be in outage by the SINR Predictor.

In addition, for a given frequency resource block b, the term $\tau_{u,u'}^s$ in Equation 10 is computed as:

$$\tau_{u,u'} = (p_{u,u'})^2 \tag{11}$$

where given the channels h_u and $h_{u'}$ of MSs u and u', respectively, p represents the spatial correlation between these two MSs on a given frequency resource block and is given by the maximum normalized scalar product [21, 11]:

$$p_{u,u'} = \frac{|h_u^{\mathrm{H}} h_{u'}|}{||h_u||_2 ||h_{u'}||_2} \tag{12}$$

Complexity. An arbitrary MIMO beamformer technique that optimizes beamforming weights under a certain criteria does so by solving a system of M linear equations. Using Gaussian elimination for this purpose results in an arithmetic complexity of $\mathcal{O}(M^3)$ [22], where M is the number of antennas in the BS. Next, we will show that the complexity of our proposed SINR Predictor is $\mathcal{O}(M)$.

The complexity of computing the term τ itself is linear on the number of antennas in the BS, i.e. $\mathcal{O}(M)$. In addition, within the context of an SDMA group evaluation, τ must be calculated for each MS to every other interfering MS. This means

Algorithm 2 Cluster Swap Operation in CBA.

```
Require:
G - Set of SDMA groups
procedure CLUSTERSWAP(G)
    Δ ← 0                                              ▷ Capacity gain of best solution
    for i = 1 to K do
        for j = 1 to K do
            if i == j OR cluster(i) == cluster(j) then
                continue                               ▷ MS i and j are in the same cluster
            C ← c(cluster(i)) + c(cluster(j))          ▷ Calc sum capacity of cluster(i) and cluster(j) before swap
            C' ← c(cluster(i)\{i} ∪ {j}) + c(cluster(j)\{j} ∪ {i})   ▷ Calc sum capacity of new cluster(i) and cluster(j) after swapping i and j
            if outageMS(cluster(i)) == ∅ ∧ outageMS(cluster(i)\{i} ∪ {j}) ≠ ∅ then
                continue                               ▷ Do not destroy valid clusters: if no MS was in outage before swap then no MS should be in outage after swap
            if outageMS(cluster(j)) == ∅ ∧ outageMS(cluster(j)\{j} ∪ {i}) ≠ ∅ then
                continue                               ▷ Do not destroy valid clusters
            if C' − C > Δ then
                G* ← updateCluster(G, i, j)            ▷ Save solution with i and j being swapped
                Δ ← C' − C
    if Δ > 0 then
        G ← G*                                         ▷ Process swap + return new cluster config
return G                                               ▷ Return MS clustering/grouping
```

Algorithm 1 Cluster-based SDMA Grouping Algorithm.

```
Require:
 1: K - Number of MSs; M - Number of BS antennas
 2: procedure CBA
 3:     G ← createRndClusters()      ▷ Create ⌈K/M⌉ many clusters with MSs randomly assigned
 4:     while true do
 5:         for x = 1 to K do
 6:             G' ← clusterSwap(G)          ▷ Process cluster swap op.
 7:             G'' ← clusterJump(G')        ▷ Process cluster jump op.
 8:             if G'' == G then             ▷ Clustering does not change
 9:                 break
10:             G ← G''
11:         if getOutageMSs(G) == ∅ then     ▷ Estimate MSs in outage
12:             break
13:         G ← createNewClusters(G)     ▷ Create additional clusters for MSs in outage
14: return G                             ▷ Return MS clustering/grouping
```

that for a given group size g there will be a total of $g(g-1)$ evaluations of the term $\tau_{u,u'}$, where $g \leq M$, hence resulting in an overall complexity of $\mathcal{O}(M^3)$. However, the term $\tau_{u,u'}$ can be pre-computed and stored. Pre-generating the values for every MS pair reduces the complexity of evaluating an SDMA group to $g(g-1)$ simple lookups, and, given that $g \leq M$, the overall complexity is $\mathcal{O}(M^2)$. Finally, for any SDMA grouper that evaluates SDMA groups iteratively through sequentially adding MSs (e.g. BFA or our CBA grouper) the complexity could be reduced even further. The reason is that every time a new MS is added to a group, every MS within the group only needs to add its correlation to the added MS, whereas the newly added MS sums the correlation to every other MS, hence resulting in $(g-1+g-1)$ lookups and a complexity of $\mathcal{O}(M)$, i.e. $\mathcal{O}(N)$ in common complexity terminology where N corresponds in this case to the number of MSs.

5.2. Cluster-based Grouping Algorithm

The SDMA group formation problem described in Equation 4 has a high computational complexity. Therefore, in this section we present a heuristic method to reduce it. Ideally, the proposed heuristic solution should have the following properties:

- Create SDMA groups that maximize a given group capacity metric.
- Keep the number of SDMA groups low, because different groups can only be scheduled in non-overlapping time-frequency resources.
- Ensure that MSs in an SDMA group are not in outage, i.e. their SINR is above a minimum threshold

Our solution to the SDMA group formation problem is the *Cluster-Based grouping Algorithm (CBA)* described in Algorithm 1. The core idea behind CBA is the following. Initially, CBA creates $\lceil K/M \rceil$ clusters, assuming that with M antennas we can spatially separate M MSs, and randomly distributes MSs among these clusters (line 3 in Alg. 1). Afterwards, the initial random groups are iteratively improved by means of two elementary functions (*clusterSwap* and *clusterJump*) (lines 6-7 in Alg. 1), used to exchange the membership of MSs between groups. *ClusterSwap* exchanges two MSs from different clusters with each other, whereas *clusterJump* allows a MS to leave one cluster and join another one. In order to bound execution time, the number of swap and jump operations is bounded by K, i.e. the number MSs (line 5 in Alg. 1). In addition, if at the end of one iteration, some MSs are found to be in outage (meaning that the number of clusters is too small to handle all MSs) new clusters are added and the in-outage MSs are distributed among them (lines 11-13 in Alg. 1). Notice that in each iteration the number of clusters is increased by at least one. Finally, CBA terminates when no more MSs are in outage.

The *clusterSwap* operation is described in Algorithm 2. Its main task is to find the swap between two MSs from different clusters, where the improvement in terms of a given group metric $c(\cdot)$ is largest. Notice, that when computing the group metric, only MSs having a SINR above the minimum threshold are considered, where SINR is calculated using our SINR Prediction algorithm (refer to 5.1). In addition, the swapping operation preserves valid groups; i.e. if before the swap none of the MSs within a cluster is in outage then this property is preserved after performing the swap (lines 10-12 in Alg. 2).

The *clusterJump* operation is similar to *clusterSwap* except that a MS leaves a cluster and joins another cluster so that the improvement in terms of group metric is largest. *ClusterJump* is described in Algorithm 3. In addition, in *clusterJump*, the computed group metric, C', is divided by the number of clusters in order to account for the fact that having MSs grouped in less clusters is more efficient, because they can share the same time-frequency resources (line 11 in Alg. 3). Finally, a jump is only allowed if after the jump all MS in the new cluster are not in outage (line 12-14 in Alg. 3).

Algorithm 3 Cluster Jump Operation in CBA.

Require:
1: $\mathcal{G}$ - Set of SDMA groups
2: **procedure** CLUSTERJUMP($\mathcal{G}$)
3: $\Delta \leftarrow 0$ ▷ Capacity gain of best solution
4: **for** $i = 1$ to K **do**
5: **for** $j = 1$ to K **do**
6: **if** i == j OR cluster(i) == cluster(j) **then**
7: continue ▷ MS i and j are in the same cluster
8: $C \leftarrow (c(\text{cluster}(i)) + c(\text{cluster}(j)))/2$ ▷ Cap. before jump
9: $C' \leftarrow c(\text{cluster}(i)\backslash\{i\}) + c(\text{cluster}(j) \cup \{i\})$ ▷ After jump
10: **if** $\text{cluster}(i)\backslash\{i\} \neq \emptyset$ **then**
11: $C' \leftarrow C'/2$ ▷ cluster(i) is not empty after jump
12: **if** $C' - C > \Delta \wedge outageMS(\text{cluster}(j) \cup \{i\}) \neq \emptyset$ **then**
13: $\mathcal{G}^* \leftarrow \text{updateCluster}(\mathcal{G}, i, j)$ ▷ Save solution
14: $\Delta \leftarrow C' - C$
15: **if** $\Delta > 0$ **then**
16: $\mathcal{G} \leftarrow \mathcal{G}*$ ▷ Process jump + return new cluster config
17: **return** $\mathcal{G}$ ▷ Return MS clustering/grouping

Complexity. Given that *clusterSwap* and *clusterJump* have a complexity of $\mathcal{O}(K^2)$, a maximum of K swap and jump operations are allowed (line 5), and in the worst case there is always at least one MS in outage after each iteration, the worst case computational complexity of CBA regarding its execution time is $\mathcal{O}(K^4)$, i.e. $\mathcal{O}(N^4)$ in common complexity terminology where N corresponds in this case to the number of MSs. For comparison, BFA has an execution time complexity of $\mathcal{O}(N^2)$.

5.3. Adaptive Modulation and Coding (AMC) module

The Adaptive Modulation and Coding (AMC) module receives a set of proposed SDMA groups from CBA and is in charge of deciding the Modulation and Coding Scheme (MCS) to be assigned to each MS in each spatial group. Notice though that the received SDMA groups were created based on the SINR estimation provided by the SINR Predictor which, as we will later show, tends to be optimistic especially as the number of antennas grows. Therefore, one of the goals of the AMC module is to correct any errors introduced by the SINR predictor.

In order to derive the MCS to be assigned to each MS, the AMC module computes the actual pre-coding matrixes per frequency block (e.g. PUSC cluster) that will be used to steer separated beams for each MS in an SDMA group. Once the pre-coding matrixes are computed, the AMC module can obtain a reliable SINR estimate for each MS and frequency block by means of Equation 1. Then, the AMC module collapses the set of per-frequency block SINR values, $\{\gamma_{u,b}\}$, into a single equivalent SINR value for each MS making use of the EESM mapping technique. Finally, a single scalar SINR value is available for each MS and given a target block error rate (1% in our case), the AMC module selects the proper MCS by doing a simple look up operation on a pre-computed *Block Error Rate-SNR* table.

Algorithm 4 Required preprocessing step in the AMC module when SINR prediction algorithm is used for grouping.

Require:
K - Number of MSs; M - Number of BS antennas
procedure PROCESSOUTAGEMSS
$U \leftarrow 1 : K$ ▷ Set of all MSs.
$\mathcal{G}^* \leftarrow \emptyset$ ▷ Final grouping configuration.
for $maxGroupSz = M$ downto 2 **do**
$\mathcal{G} \leftarrow \text{CBA}(U, maxGroupSz)$ ▷ Using SINR predictor
$U \leftarrow \emptyset$ ▷ MSs to be regrouped
for $G_i \in \mathcal{G}$ **do**
while $O = \text{getOutageMSsAfterPrecoding}(G_i) \neq \emptyset$ **do**
$u \leftarrow \arg\max_{u \in O}\{SNR_{\text{MRC}}(u) - \gamma(u, G_i)\}$ ▷ Get the MS which suffered most
$G_i \leftarrow G_i \backslash u$ ▷ Remove u from group G_i
$U \leftarrow U \cup u$ ▷ Add u to MSs to be regrouped.
$\mathcal{G}^* \leftarrow \mathcal{G}^* \cup G_i$ ▷ Valid group G_i is added to final grouping
if $U == \emptyset$ **then** ▷ No more MSs in outage; terminate
break
return $\mathcal{G}$ ▷ Return grouping

Notice that being the SINR Predictor algorithm optimistic, the AMC module might detect some of the MSs in the proposed SDMA groups to be in outage. This situation is dealt with in Algorithm 4, where for each SDMA group G having MSs in outage, the worst MS u (largest $SNR_{\text{MRC}}(u) - \gamma(u, G)$), is removed from the group. Then, a new precoding matrix is calculated for the MSs left in the group. The removed MSs are collected and handed back to CBA that proposes additional SDMA groups containing only the in-outage MSs (Figure 5). In order to decrease the likelihood of these MSs being in outage again, the maximum allowed group size is reduced by one[5]. In the worst case the algorithm terminates when a maximum group size of two is reached.

Complexity. Given a set of SDMA groups, the number of precoding calculations performed by the AMC module corresponds to the product between the number of groups, $|\mathcal{G}|$, and the number of frequency blocks, B. However, additional precoding calculations may be required if there are MSs in outage. In the

[5]In Section 6 we will show that the estimation accuracy of the SINR Predictor is better for small group sizes.

Algorithm 5 The SDMA-OFDMA packing algorithm partitions a frame into containers representing the packing area for SDMA groups.

Require:

1: $\mathcal{G}$ - set of SDMA groups; P - list of packets destined to MSs in $\mathcal{G}$ sorted according to utility per slot; SCH - no. of subchannels; DL_{slots} - number of DL slots; $C_i.sz/C_i.schedSz$ - size demanded/scheduled by container C_i; $C_i.maxM/schedMSz$ - MAP size required/scheduled by container C_i
2: **procedure** framePart($\mathcal{G}$, P, SCH, DL_{slots})
3: $\quad C_{\text{all}} \leftarrow \text{initContainers}(\mathcal{G}, SCH)$ ▷ create container for each group
4: $\quad Q_{\text{up}} \leftarrow C_{\text{all}}; F \leftarrow SCH \times DL_{slots} - \text{FCH}; FS \leftarrow F$ ▷ initially all containers must be updated; calc total and available free space frame space
5: $\quad$ **while** $|Q_{\text{up}}| > 0$ **do**
6: $\quad\quad$ **for** $C_i \in Q_{\text{up}}$ **do** ▷ for each competing container
7: $\quad\quad\quad C_i.\text{maxM} \leftarrow FS - C_i.\text{sz} + C_i.\text{schedSz} + C_i.\text{schedMSz}$ ▷ calculate maximum MAP size available for container C_i
8: $\quad\quad\quad$ **if** $|P(C_i)| > |C_i.\text{schedP}|$ **and**
9: $\quad\quad\quad\quad C_i.\text{maxM} > C_i.\text{schedMSz}$ **then** ▷ Sufficient packets and the maximum possible MAP size is greater than the currently required MAP size
10: $\quad\quad\quad\quad C_i.\text{metric} \leftarrow \text{contMetric}(C_i, P(C_i))$ ▷ update container metric
11: $\quad\quad\quad$ **else if** $FS/SCH < 1$ **then** ▷ if we cannot increase the container
12: $\quad\quad\quad\quad C_i.\text{finished} \leftarrow \text{true}$ ▷ make container size fixed
13: $\quad\quad maxId \leftarrow \text{getMaxId}([C_{\text{all}}.\text{metric}])$ ▷ select best container
14: $\quad\quad C_{\text{all}}(maxId).\text{schedule}()$ ▷ meet demand of best container
15: $\quad\quad FS \leftarrow F - \text{getUsedSpace}(C_{\text{all}})$ ▷ update free space
16: $\quad\quad Q_{\text{up}} \leftarrow \text{incrDemand}(C_{\text{all}}, FS)$ ▷ update container demand (i.e. increased container width) and add them to list
17: $\quad \overline{S} \leftarrow [C_{\text{all}}.\text{schedSz}]; \overline{M} \leftarrow [C_{\text{all}}.\text{schedMSz}]$ ▷ Size of each container and DL-MAP
18: **return** $(\overline{S}, \overline{M})$ ▷ container and map size vector

worst case the SINR predictor constructs $|\mathcal{G}|$ groups; each consisting of M highly correlated MSs. In that case, in each iteration of Algorithm 4 each group is reduced to a group consisting of only one MS, resulting in a total of $\mathcal{O}(|\mathcal{G}|M^2B)$ precoding calculations. Therefore, if $|\mathcal{G}| = \lceil K/M \rceil$ the worst case time-complexity regarding the number of precoding calculations is $\mathcal{O}(BMK)$, i.e. $\mathcal{O}(N)$ in common complexity terminology where N corresponds in this case to the number of MSs. In Section 6 we will show that in practical scenarios, this algorithm operates well below its worst case complexity.

5.4. SDMA-OFDMA Packing Algorithm

Having addressed in the previous sections the SDMA group formation problem described in Equation 4, we present in this section our heuristic solution to the SDMA-OFDMA frame construction problem described in Equation 7.

Given a set of SDMA groups we need to select a subset of groups to be scheduled in the OFDMA frame and allocate for each group a certain space within the frame, which we hereafter refer to as a *container*. In addition, we also need to allocate a certain amount of space for signaling (MAP messages) that varies depending on the selected SDMA groups. Figure 7 illustrates an example frame layout to be generated by sGSA, where we can recognize three containers. Each container is associated with an SDMA group, whereas the container at position zero is reserved for the transmission of signaling data (Preamble, FCH, MAPs). Each container has a variable width (time slots) but always spans all subchannels (subcarriers) of a given time slot. Moreover, each container consists of multiple spatial layers carrying data bursts belonging to the MSs of the associated SDMA group. In the example in Figure 7 the number of spatial layers is two and three for the two data containers respectively.

Thus, our SDMA-OFDMA packing algorithm fulfills the tasks of selecting and allocating SDMA groups within the OFDMA frame, in order to maximize the overall carried utility. Algorithm 5 presents our SDMA-OFDMA packing algorithm that consists of two phases: i) the *preparation* phase (lines 3-4), and ii) the *extension* phase (lines 5-16).

Preparation phase - A container can be either in status *scheduled* or *unscheduled* where at the beginning all containers are set to *unscheduled*. The initial width of a container is the maximum between one time slot, and the minimum number of time slots required to carry at least one packet.

Extension phase - In this phase, upon each iteration, a *container metric* representing the value of each candidate container is used to evaluate different containers (lines 6-12). The way to compute this metric is described in Algorithm 6, where the basic idea is to greedily select the container that brings the highest increase in terms of utility per slot. In addition, a container is considered a suitable candidate only if there is enough space in the frame to include its required signaling (MAP message, line 9). Thus, in each iteration of Algorithm 5 (lines 13-16) either an *unscheduled* container is added to the frame (set to status *scheduled*) or an already *scheduled* container is increased in size by one time slot. This phase is repeated until all time slots in the frame have been assigned.

Algorithm 6 The Container metric is the ratio between the additional utility resulting from the enlargement of the container and the required additional frame space for the MAP and the container itself, i.e. the additional utility per slot gain.

Require:

C_i - container to be evaluated
P - a list of packets destined to the MSs in C_i
procedure contMetric(C_i, P)
$\quad P_{\text{old}} \leftarrow \text{contPackEST}(C_i.\text{gr}, P, C_i.\text{schedSz}, C_i.\text{schedMSz})$
$\quad P_{\text{new}} \leftarrow \text{contPackEST}(C_i.\text{gr}, P, C_i.\text{sz}, C_i.\text{maxM})$
$\quad m \leftarrow \frac{u_{(\cdot)}(P_{\text{new}}) - u_{(\cdot)}(P_{\text{old}})}{\max((\text{map\_sz}(P_{\text{new}}) - \text{map\_sz}(P_{\text{old}})) + (C_i.\text{sz} - C_i.\text{schedSz}), 1_{\text{slot}})}$
▷ Container size of P_{old} and P_{new} is C_i.schedSz and C_i.sz respectively
return (m)

Finally, the container metric procedure depicted in Algorithm 6 needs a way to obtain the list of the packets that can

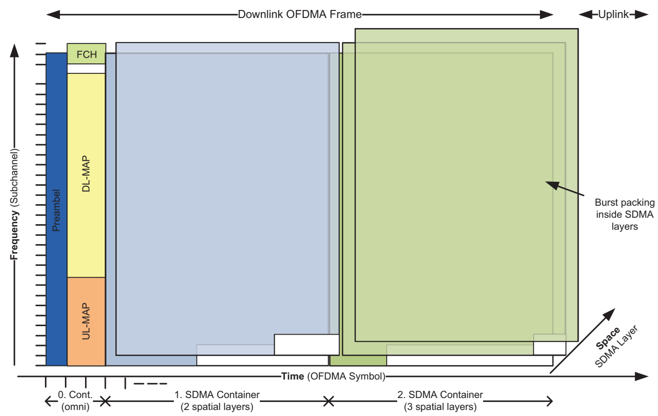


Figure 7: WiMAX SDMA-OFDMA frame layout.

be packed in the spatial layers of a given container while respecting a maximum MAP size (lines 4-5, Algorithm 6). For this purpose, Algorithm 7 greedily iterates over the list of packets addressed to the MSs of the corresponding SDMA group and adds them to their respective spatial layers assuming burst concatenation[6] until the available capacity is reached. The considered packet list is pre-sorted according to utility per slot in descending order to ensure that the most valuable packets are packed first. In case the required MAP size is larger than the maximum allowed MAP size the excess number of packets is calculated and the packets with the lowest utility are removed (lines 13-14 in Algorithm 7).

Complexity. In the worst case all MS are fully correlated and we have K containers, i.e. one MS per SDMA group. According to Algorithm 5 in each iteration the width of an already scheduled container is increased by one or an unscheduled container with its initial width is added to the frame (lines 13-16). Therefore, Algorithm 5 terminates after DL_{slots} iterations, where DL_{slots} is the number of time slots in the downlink OFDMA frame. In the worst case in each iteration the container metric (Algorithm 6) has to be calculated for each group due to a changing constraint for the maximum allowed signaling ($\mathrm{MAP}_{\max}$). Thus the total number of times the container metric is executed is K times for the initial K containers and $DL_{slots} \times K$ times until the algorithm terminates. The container metric makes use of the packing estimator (Algorithm 7) which itself has computational complexity of $\mathcal{O}(P)$, where P is the number of packets addressed to a given SDMA group (container). Therefore the overall complexity of our SDMA-OFDMA frame construction algorithm is $\mathcal{O}(K + DL_{slots} \times K) \times \mathcal{O}(P)$. Notice though that DL_{slots} is bounded by the frame size, and if required the number of packets P can be limited by the QoS scheduler. Thus, resulting in a complexity of $\mathcal{O}(K)$, i.e. $\mathcal{O}(N)$ in common complexity terminology where N corresponds in this case to the number of MSs in the system.

[6] Concatenated packets are packets addressed to the same MS that are coded and modulated together. In this way a single entry in the MAP is required that reduces overhead. In sGSA the MCS of the resulting concatenated burst is set to the lowest MCS among the concatenated packets, as given by the AMC module.

Algorithm 7 Algorithm estimates the set of packets which can be packed in a given SDMA container where the maximum map size is restricted.

Require:
1: G - set of MSs representing a SDMA group
2: P - packet list for MSs in G sorted by utility per slot in descending order
3: C_{sz} - the capacity of the container in slots
4: $\mathrm{MAP}_{\max}$ - maximum map size which can be used (in slots)
5: **procedure** $\mathrm{contPackEST}(G, P, C_{\mathrm{sz}}, \mathrm{MAP}_{\max})$
6: $\quad P^* \leftarrow \emptyset$
7: $\quad C \leftarrow new\ C(G, C_{\mathrm{sz}})$ ▷ create container
8: $\quad$ **for** $p \in P$ **do**
9: $\quad\quad$ **if** $C.\mathrm{canAddToLayer}(p)$ **then** ▷ if sufficient space in spatial layer
10: $\quad\quad\quad P^* \leftarrow [P^*\ p]$ ▷ append to packet list
11: $\quad\quad\quad C.\mathrm{addToLayer}(p)$ ▷ add packet to corresponding layer
12: $\quad map_{sz} \leftarrow \mathrm{map\_sz}(P^*)$ ▷ required MAP size to address packets
13: $\quad$ **if** $map_{sz} > \mathrm{MAP}_{\max}$ **then** ▷ if max MAP size exceeded
14: $\quad\quad P^* \leftarrow \mathrm{removeLowestUtility}(\mathrm{P}^*, map_{sz} - \mathrm{MAP}_{\max})$ ▷ remove packets with smallest utility
15: **return** P^*

6. Performance Evaluation

The performance of our proposed sGSA algorithm is analyzed in this section by means of simulations. First, we describe our simulation framework providing information about the level of detail considered in the modeling as well as the configuration parameters and settings considered. Second, a thorough performance evaluation of the different sGSA algorithm components described in the previous section is provided followed by an overall system study.

6.1. Simulation Framework

6.1.1. Simulation Parameters

A BS with a Uniform Linear Array (ULA) of M elements separated by 0.5λ is assumed. In each simulation an 802.16 OFDMA frame is considered having a duration 5 ms, containing a single PUSC zone and a 35/12 DL/UL ratio. We explicitly simulate PUSC using the permutation scheme defined in the standard [3]. For robustness, one MAP repetition and QPSK 1/2 was used for the DL-MAP, which is transmitted omnidirectionally. The size of the DL-MAP was calculated according to the parameters given in Table 2. A system bandwidth of 10 MHz is considered, which results in an 802.16 DL subframe composed of 30 logical subchannels (60 PUSC clusters). An equal transmit power and the same average noise power is assumed for all subcarriers. Since we assume low mobility, the channel transfer matrix is considered stable for the duration of one OFDMA frame, which is supported by the fact that the typical coherence time in 802.16 for a carrier frequency of 2.5 GHz and a mobility speed of 2 km/h is 200 ms [23], and is a common assumption in the SDMA literature [1].

We consider that the BS acquires channel feedback from each MS using UL Sounding with decimation, as described in Section 2. In particular, we assume that for each MS the BS

Parameter	Value
System bandwidth	10 MHz
Subcarrier bandwidth	10.9375 kHz
FFT size	1024
Center frequency	2.5 GHz
Frequency reuse pattern	3x1x1
Transmit power	46 dBm
MS noise density (dBm/Hz)	-167 dBm/Hz
Cell radius	$\sim 288\,m$
WIM scenario	C2 (urban macro-cell, LOS)
Number of antennas at BS (M)	1-5 omni elements separated by half wavelength
MSs placement	uniform
MSs' speed	2 km/h
OFDMA frame duration	5 ms
Tx Precoding algorithm	MinMSE
No. of $H_{u,s}$ per MS	30 ($D = 24$)
DL/UL ratio	35/12
MAP	QPSK 1/2
F_0	fixed_dl_map_overhead (72 bits)
F_1	dl_map_ie_fixed_overhead (44 bits)
F_2	dl_map_ie_cid_size (16 bits)
SC	num_subcarriers_slot (48)
MRep	1 MAP repetition
WiMAX permutation scheme	PUSC
Fading outage margin	3 dB
Packet buffer size	$K \times 12.96$ KiB
No. of placement seeds	256

Table 1: Simulation Parameters.

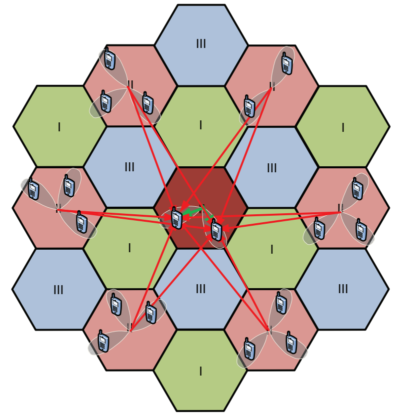


Figure 8: Frequency reuse patterns 3x1x1 used in simulations. The roman numerals depict the used frequencies.

Variable	Value
F_0	fixed_dl_map_overhead (72 bits)
F_1	dl_map_ie_fixed_overhead (44 bits)
F_2	dl_map_ie_cid_size (16 bits)
SC	num_subcarriers_slot (48)

Table 2: Parameter used for the calculation of the DL-MAP size.

knows the channel transfer matrix ($h_{u,f}$) of every 28th subcarrier, hence resulting in 30 $h_{u,f}$ coefficients per MS, one per logical subchannel. We assume that the BS has full CSI on each $h_{u,f}$, i.e. there are no estimation errors.

MSs are placed uniformly within the cell around the BS. The transmit power of the BS is adjusted so that the mean SNR of MSs placed at the cell edge is equal to the minimum SNR needed for transmission plus an additional outage margin of 5 dB. The objective is to keep outage due to fading effects[7] low. The spatial and temporal characteristics of each signal path between MSs and BS are modeled according to the LoS C1 Suburban macro-cell scenario from the WINNER Phase II Project [24]. The most important parameters used in our simulations are summarized in Table 1.

6.1.2. Error Model

The performance of an SDMA-OFDMA Scheduler has to be evaluated in terms of the data that can be successfully decoded by the spatially multiplexed MSs. For this purpose, we implement the following error model. First, besides the intra-group interference, we explicitly simulate the effect of inter-cell interference using a typical 3x1x1 deployment. Our evaluation scenario is depicted in Figure 8, where we analyze the performance achieved in the target cell while explicitly considering the tier-one interfering cells. The interfering cells also implement our sGSA scheme, hence creating random interference on the target cell.

Specifically, our error model is implemented as follows. For each burst packed by sGSA in the OFDMA frame, the real SINR for all used subcarriers considering the actual intra-cell as well as inter-cell interference is computed as:

$$\tilde{\gamma}_{u,b} = \frac{P_{u,b} \cdot ||W^{\mathrm{H}}_{u,b} \cdot H_{u,b}||_2^2}{\delta_{u,b} + \sum_{c=1}^{\#\mathrm{BSIntf}} P_{u_c,b} \cdot ||H_{u_c,b}||_2^2}, \quad (13)$$

where $\delta_{u,b}$ represents the intra-cell interference plus noise given by the denominator term of Equation 1 and $\#\mathrm{BSIntf}$ is the number of tier-one interfering BSs. Then, the set of obtained SINR values is collapsed into an equivalent SNR over an AWGN channel using the EESM mapping. Given the equivalent SNR and the used MCS, a block error rate is obtained from a precomputed table look up (BuER). Finally, after sGSA completes the construction of an SDMA-OFDMA frame, the utility/Bytes carried in each burst is multiplied by (1 - BuER) to account for retransmissions.

Percentage	MPDU Size (bytes)
18.89%	40
12.09%	1500
6.14%	62
4.67%	1420
4.60%	52
53.61%	uniform(40,1500)

Table 3: MPDU Size Distribution

6.1.3. Traffic Model

The following traffic model is used in our evaluation. The BS maintain per-MS packet buffers, and packets are generated based on a distribution derived from a data collection campaign by SPRINT[8]. The obtained packet size distribution, given in

[7]Such a high outage margin was necessary due to fairness comparison reasons when comparing with the baseline setup (single antenna BS).

[8]https://research.sprintlabs.com/packstat/

Table 3, is dominated by TCP flows. We do not considered the offered load to be equally distributed among all active MSs, but instead consider a more unbalanced situation where 50% of the MSs generate 80% of the traffic at the MAC layer [9]. The effect of user density will be studied in the performance evaluation.

Thus, in order to collect the performance metrics that will be shown in the next section, we run 256 independent simulations (i.e. MSs placement drops) of duration 1 s, explicitly simulating the OFDMA frame construction process for all OFDMA frames transmitted within 1 s.

6.1.4. QoS Scheduler Model

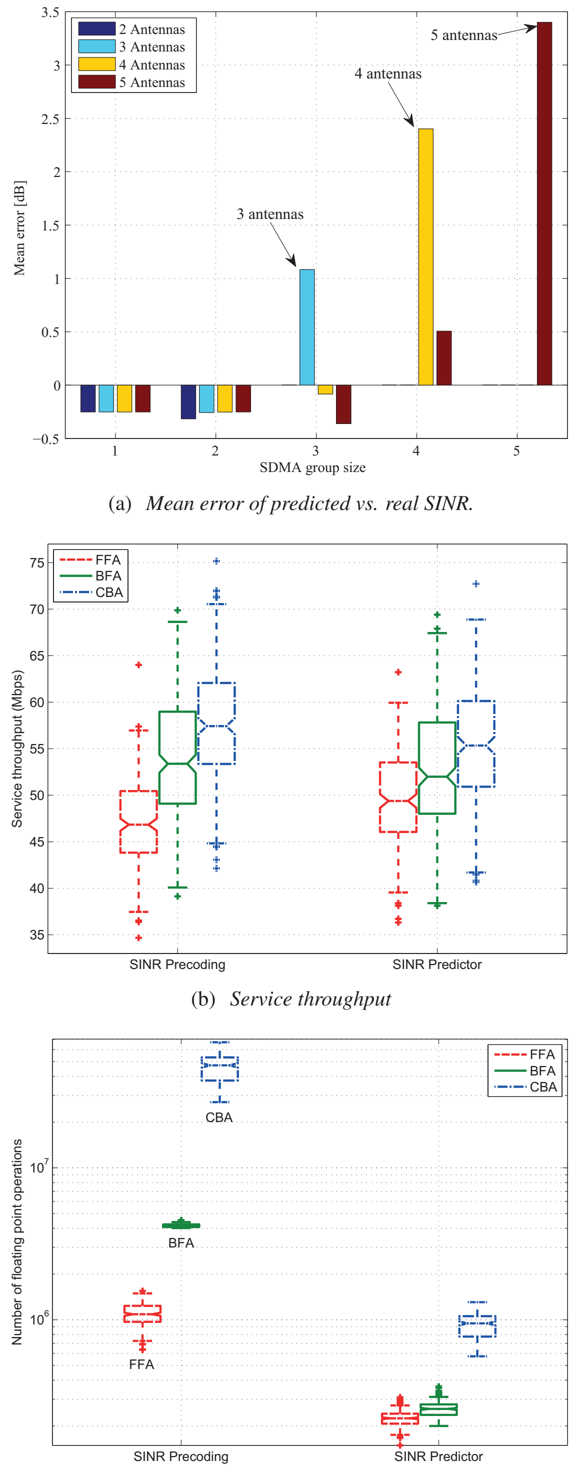
The QoS scheduler is in charge of selecting the packets available in the BS queues, and assigning to each packet a utility value, u_i. Throughout our evaluation two different utility functions will be considered that capture possible policies considered by a service provider: *Proportional Fair Utility (PF)* [18], $u_{\mathrm{pf}}()$ - The widely used Proportional Fair policy assigns to a given packet a utility of $U_{packet} = B/T$, where T is an Exponentially Weighted Moving Average (EWMA) of the Bytes that have been scheduled for the flow this packet belongs to in previous frames, i.e.: $T(i) = (1-\alpha)\cdot T(i-1) + \alpha \cdot N(i-1)$, where $N(i-1)$ is the actual number of Bytes that were scheduled for this flow in frame $i-1$, being $N(i-1) = 0$ if there is no data to send in frame $i-1$ [10]. In addition, B is the packet size in Bytes to be scheduled for this flow in frame i. *Random Utility*, $u_{\mathrm{rnd}}()$ - With this utility function we model an arbitrary policy set by a service provider. Each packet is assigned a random utility between zero and B, i.e. $U_{packet} = unif(0, B)$, where B is the packet size in Bytes.

6.2. Simulation Results

In this section we present a thorough evaluation of the performance of sGSA that is organized as follows. First, in section 6.2.1, we study the performance and justify the need of our novel SINR Predictor module. Second, in section 6.2.2, we evaluate the performance of our proposed CBA grouping algorithm as compared to other grouping algorithms in the state of the art. Third, in section 6.2.3 we validate the assumptions behind our SDMA-OFDMA packing algorithm. Finally, in section 6.2.4, we evaluate the overall performance of sGSA against that of other algorithms in the state of the art, under a variety of system parameters.

6.2.1. SINR Prediction

In order to illustrate the accuracy of the *SINR Predictor* module introduced in Section 5.1, Figure 9(a) depicts the mean error between the predicted and the real per-subcarrier SINR (after precoding) achieved by the *SINR Predictor* for different number of BS antennas and SDMA group sizes. As depicted in the figure, the SINR Predictor tends to overestimate the per-subcarrier SINR, especially as the number of antennas in the BS grows, i.e. the SINR Predictor tends to be optimistic (up to $3\ dB$ for $M = 5$ antennas). The reason for this error is the fact



(a) *Mean error of predicted vs. real SINR.*

(b) *Service throughput*

(c) *Number of floating point operations*

Figure 9: Performance of the proposed SINR prediction algorithm. The *SINR Predictor* tends to be optimistic when the number of antennas grows. However, these estimation errors have a negligible impact on the performance of an SDMA grouping algorithm.

[9]We also evalulated the homogeneous case where all MS generate the same amount of traffic and the results were similiar.

[10]The value for α in the EWMA filter is set to 0.2

that our proposed *SINR Predictor* does not consider the spatial correlations between all MSs in the SDMA group, but only the spatial correlation between the new MS to be added to the group and the rest of MSs that are already part of the SDMA group, as depicted in Figure 6. However, as we illustrate next, this estimation error does not have a significant impact on performance.

Figure 9(b) illustrates, for a scenario with $M = 5$ antennas in the BS and $K = 20$ MSs in the cell, the achieved service throughput for different SDMA grouping algorithms, namely *CBA*, *BFA* and *FFA*. These SDMA grouping algorithms use *group capacity* as its grouping metric, and in order to obtain the per-MS SINRs required to derive the group capacity metric, we compare two different methods: i) *SINR Precoding*, where a full precoding matrix is computed every time a new SDMA group has to be evaluated, and ii) our *SINR Predictor*. As observed in Figure 9(b), for all considered grouping algorithms, the *SINR Predictor* has a very small impact on performance[11]. Note, that when using the *SINR Predictor* FFA behaves better than with precoding. The reason for this is that FFA does not consider some groups to be in outage (because it is optimistic; ref. Sec. 5.1). However, this mistake is corrected by the AMC module afterwards. Therefore, we conclude that the estimation errors introduced by the SINR Predictor module have a negligible impact on the performance of an SDMA grouping algorithm.

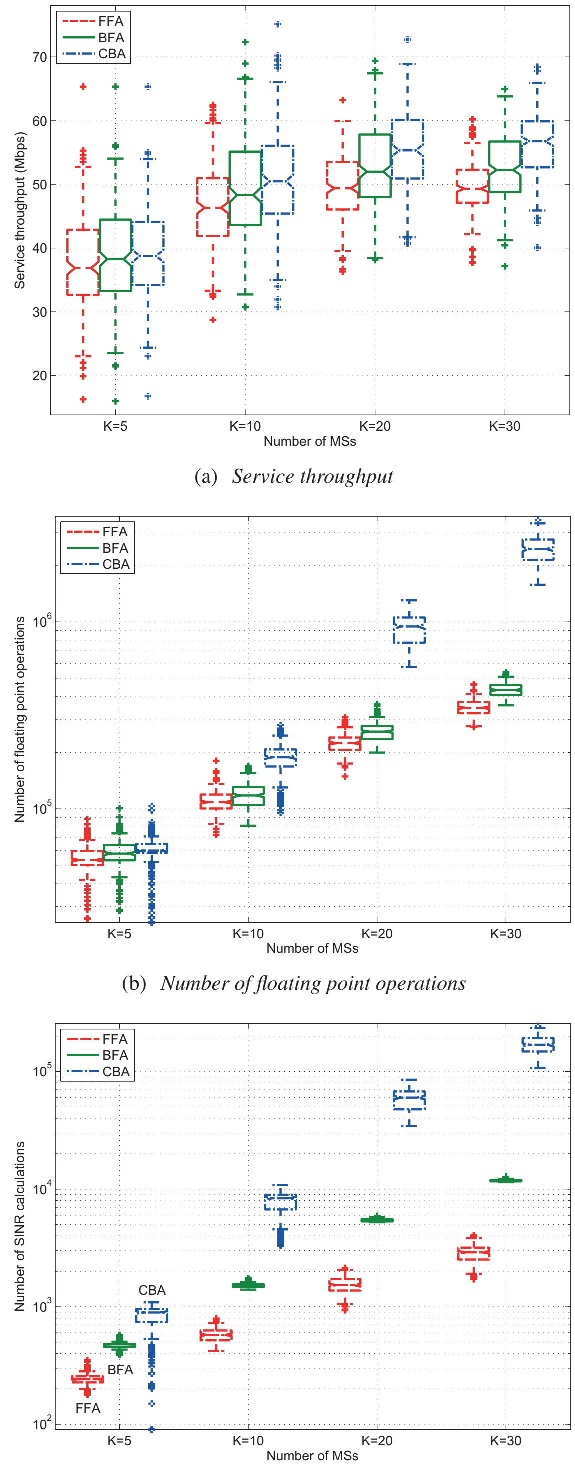
In order to evaluate the reduction in complexity achieved by the *SINR Predictor* with respect to the traditional *SINR Precoding* method, we define as our measure of complexity the total number of executed floating point operations (FLOPs) when using each of our algorithms under study. The number of FLOPs required in typical matrix operations are obtained from [25].

Figure 9(c) shows the corresponding total number of FLOPs executed with the *CBA*, *BFA* and *FFA* SDMA grouping algorithms, when using the *SINR Precoding* method, i.e. full precoding matrix computation, and our *SINR Predictor*. As clearly observed in the figure, the reduction in the number of FLOPs achieved by the *SINR Predictor* is very significant for all grouping algorithms, resulting in a reduction of almost two orders of magnitude in the case of *CBA*. The main reasons for the smaller complexity required by the *SINR Predictor* method are: i) the reduced number of precoding matrix calculations, i.e. Equation 2, that in this case are only required in the AMC module, and ii) the possibility of reusing the correlation coefficients between pairs of MSs when evaluating different candidate SDMA groups.

Hereafter, we consider that the per-MS SINR computations required to derive the group capacity metric are always computed using our proposed *SINR Predictor*.

6.2.2. Cluster-based Grouping Algorithm (CBA)

In this section we investigate further into the performance trade-offs of our *CBA* grouping algorithm, as compared to the traditional *BFA* and *FFA* algorithms. In particular, we are interested in studying the effect that the number of MSs in the cell, K, has on the performance/complexity trade-off of these



(a) *Service throughput*

(b) *Number of floating point operations*

(c) *Number of evaluated SDMA groups*

Figure 10: Performance of the proposed CBA grouping algorithm compared with FFA and BFA when using the SINR prediction algorithm. *CBA* is the most complex algorithm and achieves the highest throughput.

[11] We have seen similar results across a wide range of system parameters. We do not report these results here for the sake of brevity.

SDMA grouping algorithms. Notice, that as described in Section 5.2, *CBA* has a worst-case complexity of $\mathcal{O}(K^4)$, while BFA has a worst-case performance of $\mathcal{O}(K^2)$. In addition, all the considered SDMA grouping algorithms use *group capacity* as grouping metric.

Figure 10(a) illustrates the service throughput achieved by the algorithms under study, when considering $M = 5$ antennas in the BS and $K = 5, 10, 20, 30$ MSs in the target cell. As we can see in the figure, *CBA* always achieves the highest service throughput, followed respectively by *BFA* and *FFA*. In particular, the gain of *CBA* is higher as the number of MSs in the cell, K, increases, and results in around a 10% gain for $K = 30$ MSs. Notice that as K grows, the task of finding *optimal* groups becomes harder.

In order to evaluate the computational complexity of the different grouping algorithms, we depict in Figure 10(b) the number of FLOPs executed by each algorithm. As expected, *CBA* is the most complex algorithm, followed by *BFA* and *FFA*, with its relative complexity increasing as the number of MSs in the cell increases. To further illustrate the reasons behind the performance-complexity trade-off exhibited by the different grouping algorithms, Figure 10(c) illustrates for each algorithm the number of candidate SDMA groups evaluated. We can clearly see that *CBA*, by means of its *clusterJump* and *clusterSwap* operations, is the algorithm evaluating the highest number of candidate SDMA groups, which hence translates into a higher performance but also a higher complexity.

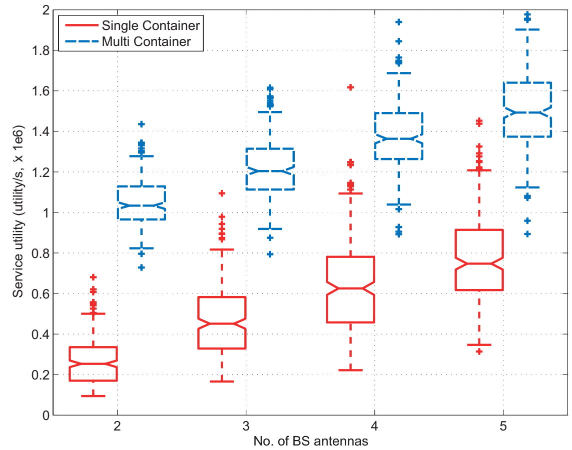
Therefore, we conclude that *CBA* and *BFA* provide a complementary performance/complexity trade-off, both being better than *FFA*, where a particular vendor could select *CBA* if it is more interested in performance and *BFA* if it is more interested in reducing complexity. In the rest of the paper we assume that *CBA* is the SDMA grouping algorithm being used.

6.2.3. SDMA-OFDMA Packing algorithm

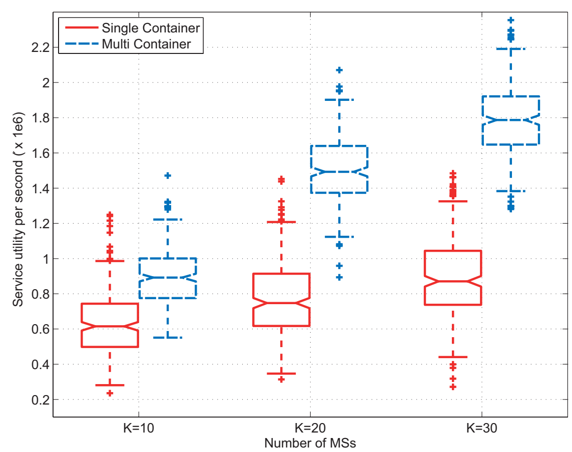
In this section we evaluate one of the main assumptions behind our SDMA-OFDMA packing algorithm, described in section 5.4. This assumption is our claim that packing several SDMA groups within a single OFDMA frame, instead of packing only the best group, delivers significant performance gains.

For this purpose, Figure 11 compares the performance of a scheduling algorithm that only packs the best SDMA group in each OFDMA frame, *Single Container*, and our proposed packing algorithm, *Multi Container*, that selects from all the available SDMA groups the ones that have to be packed in each OFDMA frame. For the purpose of this experiment a *random* utility policy is used, and we vary the number of antennas in the BS, M, and the number of MSs in the cell, K. In particular, the selected random utility policy considers that 20% of the MS are premium MSs, and thus multiplies their packet utility by a factor of ten in order to account for their importance. Notice that this utility policy may represent a real-life scenario where a service provider prioritizes some MSs over the rest.

Figure 11 depicts the results of an experiment where $K = 20$ MSs are considered, and the number of BS antennas varies from $M = 2$ to $M = 5$. As we can see in the figure the *Multi Container* approach results in significant gains, up to a four-fold



(a) *Impact of number of antennas*



(b) *Impact of number of MSs*

Figure 11: Efficiency comparison (utility per second) for different SDMA scheduling policies. The *Multi Container* approach results in significant gains compared to the *Single Container* approach.

increase, compared to the *Single Container* approach. Similarly, Figure 11(b) depicts the same results for an experiment where the number of antennas in the BS is fixed to $M = 5$, and the number of MSs in the cell increases from $K = 10$ to $K = 30$. As observed in the figure, the *Multi Container* approach outperforms the *Single Container* approach, especially as K grows. Notice that the main reason behind the gains achieved by our *Multi Container* approach is its higher flexibility in being able to conveniently schedule high priority packets from different MSs, which may not be spatially compatible, in different SDMA groups.

Regarding computational complexity, the *Single Container* algorithm has of course lower complexity than the *Multi Container* algorithm. In particular, the *Single Container* algorithm has a worst-case complexity of $\mathcal{O}(K) \times \mathcal{O}(P)$, and the *Multi Container* algorithm a complexity of $\mathcal{O}(K \times (1 + DL_{slots})) \times \mathcal{O}(P)$, where K is the number of MSs in the cell, P the number of packets[12], and DL_{slots} the number of OFDMA slots in the DL subframe. However, these differences are not significant when considering the overall complexity of an SDMA-OFDMA scheduler, because the biggest part of the complexity corresponds to the SDMA grouping algorithm that in the case of *sGSA* is optimized using the *SINR Predictor*.

Thus, given the observed results, we conclude that our proposed approach of packing multiple SDMA groups within a single OFDMA frame may translate in significant performance gains in practical scenarios. In the rest of the paper we consider the SDMA-OFDMA packing algorithm described in section 5.4 as the packing solution employed by sGSA.

6.2.4. Overall performance of sGSA

In this section we evaluate the overall performance of the following algorithms:

- *Opt+SDMA*: This scheduler is an algorithm that solves the same problem than sGSA but using exhaustive methods. In particular: i) this scheduler assumes a perfect knowledge of the per-MS SINR in order to compute the group capacity metric, ii) it generates SDMA groups by doing an exhaustive search among all possible groups, and iii) performs an exhaustive search in order to select the SDMA groups that are finally packed in the OFDMA frame. Notice that this algorithm is not implementable in practice, but can be used to benchmark the performance of sGSA.

- *sGSA+SDMA*: This is our *sGSA* algorithm, composed by all the modules described in section 5.

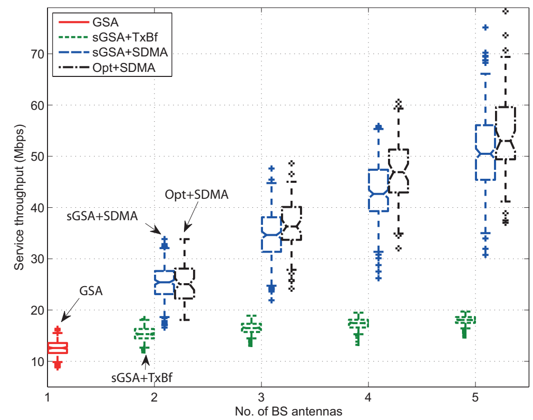
- *sGSA+TxBf*: If $M > 1$ antennas are available in the BS, instead of doing SDMA another possibility is to do *Transmit Beamforming*. Thus, instead of transmitting to multiple MSs at the same time, *sGSA+TxBf* schedules a single MS in each time-frequency region but boosts the SINR perceived by this MS using beamforming[13]. Hence, with a higher SINR, MSs can use more efficient Modulation and Coding Schemes that should increase capacity.

- *GSA*: GSA [17] is an OFDMA scheduling solution that uses two-dimensional packing. GSA though can only be used with $M = 1$ antennas in the BS.

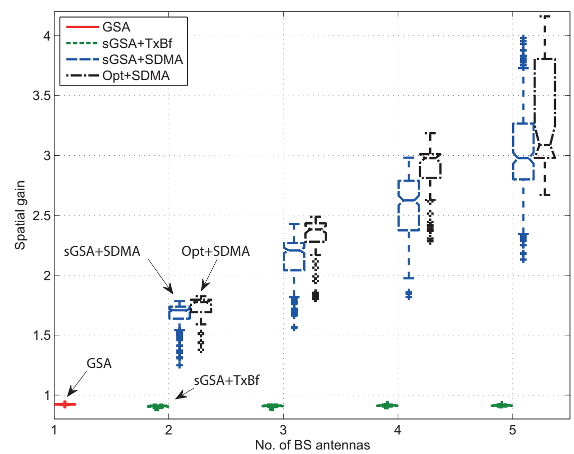
In order to analyze the previous algorithms we look at the following aspects: i) how performance scales with the number of antennas available in the BS, and ii) how the performance of each algorithm depends on the offered load available in the BS. For all the experiments presented in this section we consider that the QoS Scheduler uses a Proportional Fair (PF) utility.

Impact of the number of antennas in the BS

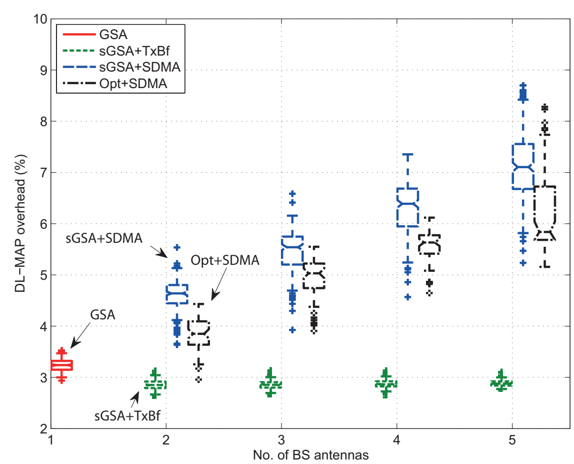
Figure 12(a) depicts the service throughput achieved by our algorithms under study in a scenario where we have $K = 10$ active MSs in the target cell. The most remarkable result in this figure is that *sGSA+SDMA* clearly outperforms *sGSA+TxBf*, up to a three-fold gain for $M = 5$ antennas in the BS, and performs



(a) *Service throughput*



(b) *Spatial gain*



(c) *DL-MAP overhead*

Figure 12: Comparison under the PF utility for different tx antennas at BS and precoding techniques in LOS scenario for $K = 10$ MSs. *sGSA+SDMA* clearly outperforms *sGSA+TxBf* and performs very close to the optimal algorithm.

[12] Notice that P can be controlled by the QoS scheduler.

[13] In this case beamforming weights are obtained using Equation 3.

very close to the *Opt+SDMA* algorithm. We analyze the previous two results separately.

First, the performance of *sGSA+TxBf* is low in this scenario, because with $K = 10$ MSs in the cell and the employed PF utility, the *Multi-User Diversity* gain is already enough for the BS to transmit to MSs that experience good channel conditions. Hence, transmit beamforming results in small gains in a scenario where MSs already experience relatively good channel conditions. Instead, a dramatic gain is obtained when the BS uses SDMA and is able to simultaneously transmit to more than one MS.

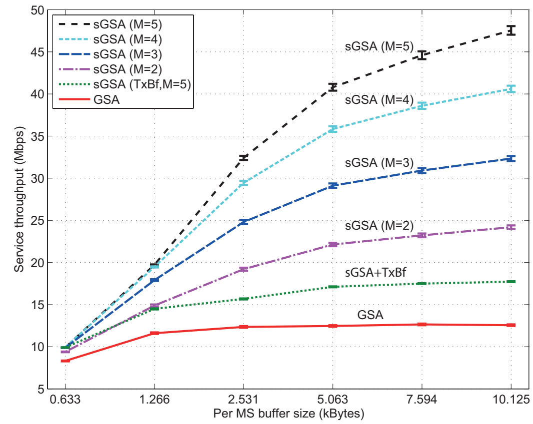
Second, although the performance of *sGSA+SDMA* is very close to the performance of the *Opt+SDMA* algorithm in Figure 12(a), we want to be cautious in generalizing this result. The reason is that as the number of MSs in the cell, K, increases, we expect the gain of *Opt+SDMA* over *sGSA+SDMA* to increase, because as K increases it becomes more difficult for *sGSA* to generate SDMA groups that are close to the optimal ones. Given the computational requirements of the *Opt+SDMA* algorithm though, we leave as future work the study of its performance for large K's.

In order to gain a deeper insight on the previous results, Figure 12(b) and Figure 12(c) depict respectively for each algorithm the spatial gain, i.e. gain due to having more antennas available in the BS, and the incurred signaling (DL-MAP) overhead. As expected, *sGSA+SDMA* delivers a linear spatial gain, which is slightly below M. For example, the spatial gain is around 3 for $M = 5$ antennas, meaning that with five antennas *sGSA+SDMA* comes up on average with SDMA groups of size three. Given the observed gain, *sGSA+SDMA* seems to be a logical extension to *GSA* when multiple antennas are available at the BS. Finally, regarding the signaling overhead depicted in Figure 12(c), the algorithms using SDMA logically result in a higher overhead because more MSs need to be signaled in the MAP. However, *sGSA+SDMA* performs significantly close to *Opt+SDMA*. Finally, it is also interesting to notice that due to the heavy use of concatenation, a very small signaling overhead is introduced by *sGSA+TxBf*.

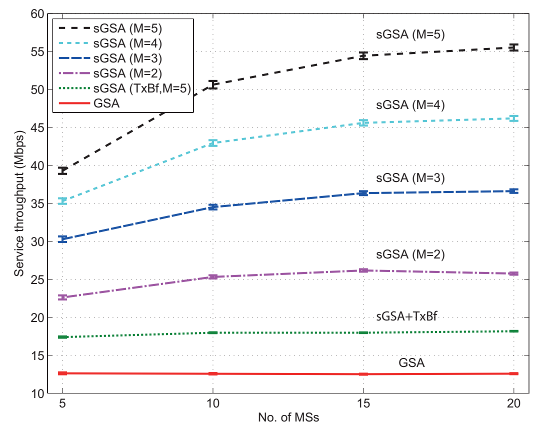
Impact of the Offered Load in the BS

We now evaluate the performance of the different algorithms under study when varying the amount of offered load available in the BS. In particular, we vary the offered load in the BS by varying: i) the amount of data available in the per-MS buffers held in the BS when considering $K = 10$ MSs in the cell, in Figure 13(a), and ii) the number of MSs in the cell when considering a fixed per-MS buffer size of $B = 12.66$ KB, in Figure 13(b). In addition, we compare the performance of *GSA* with $M = 1$ BS antennas, *sGSA+TxBf* with $M = 5$ BS antennas, and of *sGSA+SDMA* for $M = 2$ to $M = 5$ BS antennas. For the sake of clarity, we have not included the performance of the *Opt+SDMA* algorithm which was slightly above the one of *sGSA+SDMA*.

As we can see in Figure 13(a) and Figure 13(b), the algorithms under study are more sensitive to small buffer sizes than to a reduced number of MSs in the cell. The reason for this is



(a) *Impact of buffer size*



(b) *Impact of number of MSs*

Figure 13: Impact of buffer size and number of MSs. The algorithms are more sensitive to small buffer sizes than to a reduced number of MSs in the cell.

related to the packing algorithm employed by *sGSA* that allocates full columns to each individual SDMA layer, and hence can result in wasted resources if a scheduled MS does not have enough data to fill at least one column. Notice that enhanced algorithms could be devised, that depending on the load offered by each MS would decide on whether this MS should be transmitted using *sGSA+SDMA* or using another scheduler that allows a more efficient packing of small amounts of data, like *GSA*. We leave the study of these algorithms as future work. Looking at Figure 13(b), we see that increasing the number of MSs in the cell also leads to capacity gains, especially as the number of antennas in the BS grows, due to the ability of *sGSA* to come up with better SDMA groups.

It is also interesting to see how *sGSA+SDMA* is more efficient than *sGSA+TxBf* even with a reduced number of antennas in the BS, e.g. *sGSA+SDMA* with $M = 2$ antennas achieves a higher capacity than *sGSA+TxBf* with $M = 5$ for a wide range of MSs in the cell and considered buffer sizes. We believe that the reasons for this behavior are tightly bound to the used Proportional Fair (PF) utility, that reduces the capacity gains achieved by *sGSA+TxBf* as the number of MSs increases, because in order to maintain fairness the PF scheduler eventually selects all MSs. Instead, if enough antennas are available and SDMA is used, fairness can be maintained and still capacity gains are possible due to scheduling MSs in the spatial domain.

7. Conclusions and Future Work

To address future wireless networks forecasted growth in mobile traffic volume, in this paper we took a comprehensive view at SDMA-OFDMA systems which are expected to be a key technology building block to increase current spectral efficiencies. SDMA-OFDMA systems have to allocate resources in time, frequency and space dimensions to different mobile stations, resulting in a highly complex resource allocation problem. In our work, we designed and analyzed a SDMA-OFDMA Greedy Scheduling Algorithm (sGSA) for WiMAX networks. Our solution considers feasibility constraints in order to allocate resources for multiple mobile stations on a per packet basis by using i) a low complexity SINR prediction algorithm, ii) a cluster-based SDMA grouping algorithm and iii) a computationally efficient frame layout scheme which allocates multiple SDMA groups per frame according to their packet QoS utility. The overall performance of the different sGSA algorithm components was evaluated by system-level simulations and compared to state of the art approaches.

The main contributions of our work in this paper are summarized as follows. First, a complete WiMAX SDMA-OFDMA downlink scheduling solution was presented and its performance benefits quantified comparing against related work solutions. Second, a novel low complexity algorithm for predicting the SINR of mobile stations in SDMA groups was designed and the corresponding complexity reduction evaluated, reaching up to two orders of magnitude improvement in some cases. Third, a computationally efficient frame layout scheme was proposed which allocates multiple SDMA groups per frame according to their packet QoS utility at low complexity costs. Finally, the SDMA-OFDMA system signaling overhead was analyzed with respect to the usage of the overall radio resources.

As future work we consider the following steps. First, the MAP overhead reduction is a promising direction where significant additional capacity gains could be achieved, specially for a large number of antennas. More efficient MAP signaling schemes could be feasible if precoding techniques would be used for transmitting MAPS (i.e. private MAPs). Second, the impact of noise and delay in the received CSI feedback on the performance of sGSA should be studied. Finally, although in this paper we have focused in 802.16 systems, we believe that our proposed algorithms can be easily adapted to other technologies using SDMA-OFDMA, like LTE.

References

[1] T. F. Maciel, Suboptimal Resource Allocation for Multi-User MIMO-OFDMA Systems, Ph.D. thesis, Technical University of Darmstadt (2008).

[2] F. Shad, T. D. Todd, V. Kezys, J. Litva, Dynamic Slot Allocation (DSA) in Indoor SDMA/TDMA Using a Smart Antenna Basestation, IEEE/ACM Transactions on Networking 9.

[3] IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Broadband Wireless Access Systems, IEEE 802.16-2009 Standard (May 2009).

[4] A. Zubow, J. Marotzke, D. Camps-Mur, X. Pérez-Costa, sGSA: An SDFMA-OFDMA Scheduling Solution, submitted to European Wireless Conference (2012).

[5] M. Costa, Writing on dirty paper (Corresp.), Information Theory, IEEE Transactions on 29 (3) (1983) 439–441. doi:10.1109/TIT.1983.1056659. URL http://dx.doi.org/10.1109/TIT.1983.1056659

[6] R. D. Wesel, J. M. Cioffi, Achievable rates for Tomlinson-Harashima precoding, Information Theory, IEEE Transactions on 44 (2) (1998) 824–831. doi:10.1109/18.661530. URL http://dx.doi.org/10.1109/18.661530

[7] C. B. Peel, B. M. Hochwald, A. L. Swindlehurst, A Vector-Perturbation Technique for Near-Capacity Multiantenna Multiuser Communication - Part I: Channel Inversion and Regularization, Communications, IEEE Transactions on 53 (1) (2005) 195–202. doi:10.1109/TCOMM.2004.840638. URL http://dx.doi.org/10.1109/TCOMM.2004.840638

[8] F. Gross, Smart Antennas for Wireless Communications: With MATLAB (Professional Engineering), 1st Edition, McGraw-Hill Professional, 2005.

[9] T. Yoo, A. Goldsmith, On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming, Selected Areas in Communications, IEEE Journal on 24 (3) (2006) 528–541. doi:10.1109/JSAC.2005.862421. URL http://dx.doi.org/10.1109/JSAC.2005.862421

[10] M. Sharif, B. Hassibi, A Comparison of Time-Sharing, DPC, and Beamforming for MIMO Broadcast Channels With Many Users, Communications, IEEE Transactions on 55 (1) (2007) 11–15. doi:10.1109/TCOMM.2006.887480. URL http://dx.doi.org/10.1109/TCOMM.2006.887480

[11] I. Frigyes, J. Bito, P. Bakki (Eds.), Advances in Mobile and Wireless Communications: Views of the 16th IST Mobile and Wireless Communication Summit (Lecture Notes in Electrical Engineering), 1st Edition, Springer, 2008. URL http://www.springer.com/engineering/signals/book/978-3-540-79040-2

[12] C. Farsakh, J. A. Nossek, A Real Time Downlink Channel Allocation Scheme for an SDMA Mobile Radio System, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 3 (1996) 1216–1220 vol.3. doi:10.1109/PIMRC.1996.568477. URL http://dx.doi.org/10.1109/PIMRC.1996.568477

[13] A. Nascimento, J. Rodriguez, A Joint Utility Scheduler and SDMA Resource Allocation for Mobile WiMAX Networks, in: Proceedings of the 5th IEEE international conference on Wireless pervasive computing, ISWPC'10, IEEE Press, Piscataway, NJ, USA, 2010, pp. 589–594. URL http://dl.acm.org/citation.cfm?id=1856330.1856432

[14] C. Y. Huang, C.-Y. Chang, P.-H. Chen, M.-H. Wu, Performance Evaluation of SDMA Based Mobile WiMAX Systems, International Wireless Communications and Mobile Computing Conference (2008) 1042–1046doi:10.1109/IWCMC.2008.181. URL http://dx.doi.org/10.1109/IWCMC.2008.181

[15] L. Hentilä, P. Kyösti, M. Käske, M. Narandzic, M. Alatossava, MATLAB implementation of the WINNER Phase II Channel Model ver1.1. URL https://www.ist-winner.org/phase_2_model.html

[16] C. Oestges, B. Clerckx, MIMO Wireless Communications: From Real-World Propagation to Space-Time Code Design, 1st Edition, Academic Press, 2007.

[17] A. Zubow, D.Camps-Mur, X.Pérez-Costa, P. Favaro, Greedy Scheduling Algorithm (GSA) - Design and Evaluation of an Efficient and Flexible WiMAX OFDMA Scheduling Solution, Elsevier Computer Networks Journal, Volume 54, Issue 10 (July 2010).

[18] J. Lakkakorpi, A. Sayenko, J. Moilanen, Comparison of Different Scheduling Algorithms for WiMAX Base Station: Deficit Round-Robin vs. Proportional Fair vs. Weighted Deficit Round-Robin, IEEE Wireless Communications and Networking Conference (2008) 1991–

1996doi:10.1109/WCNC.2008.354.
URL http://dx.doi.org/10.1109/WCNC.2008.354

[19] R. D. Armstrong, P. Sinha, A. A. Zoltners, The Multiple-Choice Nested Knapsack Model, Management Science 28 (1) (1982) pp. 34–43.
URL http://www.jstor.org/stable/2631072

[20] H. Kellerer, U. Pferschy, D. Pisinger, Knapsack Problems, 1st Edition, Springer, 2004.

[21] D. B. Calvo, Fairness Analysis of Wireless Beamforming Schedulers, Ph.D. thesis, Technical University of Catalonia, Spain (2004).

[22] R. Farebrother, Linear Least Squares Computations (Statistics: A Series of Textbooks and Monographs), 1st Edition, CRC Press, 1988.

[23] J. G. Andrews, A. Ghosh, R. Muhamed, Fundamentals of WiMAX: Understanding Broadband Wireless Networking, 1st Edition, Prentice Hall, 2007.

[24] P. Kyösti, J. Meinilä, L. Hentilä, WINNER II Channel Models, IST-4-027756 WINNER II D1.1.2 V1.1.
URL http://www.ist-winner.org/WINNER2-Deliverables/D1.1.2v1.1.pdf

[25] R. Hunger, Floating Point Operations in Matrix-Vector Calculus, Tech. rep., Technische Universität München, Associate Institute for Signal Processing (2007).

Anatolij Zubow is a senior researcher in the Wireless Networking Group at the department of Computer Science at Humboldt-Universität zu Berlin, Germany. He received his PhD degree from Humboldt-Universität in 2009. His main research interest is cooperative mesh networking in wireless ad hoc networks. In addition he worked on several WiFi and WiMAX related projects.

Johannes Marotzke is a research assistant at the department of Computer Science at Humboldt-Universität since 2010, where he is currently completing his Diploma thesis in Computer Science. His main area of interests lie within the field of wireless MIMO communication systems and WiFi/WiMAX networks.

Daniel Camps Mur is a senior researcher in the Mobile and Wireless Networks group at NEC Network Laboratories in Heidelberg, Germany. He holds since 2004 a Masters degree and since 2012 a PhD. from the Polytechnic University of Catalonia (UPC). His current research interests include QoS and energy efficiency in wireless networks. Daniel is also an active contributor to the IEEE 802.11 group and to the Wi-Fi Alliance. In addition, his master thesis work received the 'Mobile Internet and 3G Mobile Solutions' award from the Spanish Association of Telecommunication Engineers (COIT), and in the field of network simulations he also received the OPNET's Significant Technical Challenge Solved Award.

Xavier Pérez Costa is Chief Researcher at NEC Laboratories Europe in Heidelberg, Germany, where he has managed several projects related to wireless networks. In the WiFi area, he led a project contributing to 3G/WiFi mobile phones evolution and received NECs R&D Award for his work on N900iL, NECs first 3G/WiFi phone. In the LTE area he managed a team researching on future product enhancements and received NEC's R&D Award for related contributions to WiMAX products. In the Future Wireless Internet area, he is responsible for a team contributing to the EU FP7 project Flexible Architecture for Virtualizable Wireless Future Internet Access (FLAVIA). He has participated in IEEE standardization working groups as well as their corresponding certification bodies, and has been included in the major contributor list of IEEE 802.11e and 802.11v. He received both his M.Sc. and Ph.D. degrees in Telecommunications from the Polytechnic University of Catalonia (UPC) and was the recipient of the national award for the best Ph.D. thesis on multimedia convergence in telecommunications from COIT.