Pilot Coordination in CDI Precoded Massive MIMO Systems

Samuele Zoppi
Chair of Communication Networks
Technical University of Munich
Munich, Germany
samuele.zoppi@tum.de

Michael Joham, David Neumann, and Wolfgang Utschick
Associated Institute for Signal Processing
Technical University of Munich
Munich, Germany
{joham, d.neumann, utschick}@tum.de

***Abstract*—Pilot contamination severely degrades the achievable rates in massive MIMO systems. To resolve the problem, many channel state information (CSI) and channel distribution information (CDI) methods were developed. This paper presents a CDI based pilot coordination method that leads to a considerable improvement in the achievable sum rate. It also shows that pilot assignment can drastically affect sum rate performance and describes a method that avoids worst-case conditions.**

I. Introduction

Next generation wireless networks require a huge improvement in spectral efficiency. In order to tackle this problem the massive MIMO approach is considered to be a key technique in 5G systems.

When the number of base station antennas grows, the gain and the angular resolution of the array increase. If the number of users remains small, this leads to better signal to noise and interference ratio (SINR) and more robust spatial multiplexing. The channel vectors of the users tend to be indeed orthogonal. As the array enlarges, the effects of additive noise and fast fading vanish [5] and inter-cell interference remains the main obstacle.

In this scenario, the acquisition of CSI in a timely manner plays a fundamental role. Operating the system in time division duplexing (TDD) mode, the channel reciprocity can be exploited for obtaining CSI at the transmitter [2]. The coherence time of the channel is split in uplink training and data transmission. In this way, resources needed for pilots grow with the number of served users and not with the dimension of the base station's antenna array as for frequency division duplexing (FDD).

As shown in [5], a major issue of massive MIMO is pilot contamination. Orthogonal pilot resources must be reused in every cell due to limited resources. This leads to inter-cell interference during uplink training, that turns into focused interference during downlink transmission. This interference in data transmission is the bottleneck of massive MIMO and it limits the maximum achievable rate [5].

To overcome this problem, several cooperative methods were developed [4], [1]. A promising non-cooperative approach, exclusively based on the channel's second-order statistics, was presented in [7]. It proposes different CDI based linear processing methods to remove the effect of inter-cell interference from pilot contaminated CSI.

In this work, a pilot coordination strategy is presented, that carefully selects groups of interfering users to further enhance the results obtained by CDI precoding. This enhancement is possible with the formulation of a combinatorial network utility maximization problem (NUM). This method is similar to the greedy algorithm for pilot assignment from [8], but instead of considering the channel estimate only, the entire CDI precoding chain is optimized, obtaining tailored performances results for every CDI precoder. We will show that, CDI oriented pilot coordination not only improves the spectral efficiency with respect to the average case, but also avoids rate degradation of worst-case scenarios.

The paper has the following structure. The second section describes the massive MIMO system model in a multi cellular environment. The third section contains the derivation of the achievable rate using the CDI precoding technique. In the fourth and fifth section, pilot coordination is studied and different coordination algorithms are proposed. The former describes the problem of pilot coordination, while the latter calculates the computational complexity of the algorithms. Finally, the sixth section shows the results of pilot coordination using different CDI precoders and algorithms, while in the seventh section, we draw the conclusions and we propose future investigations for CDI precoded pilot coordination.

II. System model

Consider a single-cell scenario where the base station is equipped with M antennas and serves K single-antenna users. In order to apply the massive MIMO paradigm, the condition $M \gg K$ is necessary.

The uplink channel estimation is performed synchronously for all the users, using K pilot sequences out of $N < K$ orthogonal sequences. This condition introduces intra-cell interference during channel estimation. Note that this model can be easily generalized to a multi-cell scenario with inter-cell interference and full reuse of pilots where intra-cell interference is not present, i.e., $N \geq K$ where K denotes the number of users per cell.

The channel between user i and the base station is denoted as the vector $\boldsymbol{h}_i \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \boldsymbol{R}_i) \in \mathbb{C}^M$, where the channel covariance matrix $\boldsymbol{R}_i$ can have any structure.

The pilot assigned to user i for the training phase is the column vector $\boldsymbol{\psi}_i \in \mathbb{C}^N$, where $\boldsymbol{\psi}_i^{\mathrm{H}}\boldsymbol{\psi}_i = 1$. The elements of the Hermitian matrix $\boldsymbol{C}$ are $c_{ij} = \boldsymbol{\psi}_i^{\mathrm{H}}\boldsymbol{\psi}_j \in \mathbb{C}$, $\left|c_{ij}\right| \leq 1$ and represent the correlation between two pilot sequences.

Applying least-squares channel estimation leads to the contaminated channel estimate

$$\hat{\boldsymbol{h}}_i = \sum_{j=1}^{K} \boldsymbol{h}_j c_{ji} + \frac{\boldsymbol{n}_i}{\sqrt{\rho_{\text{tr}}}} \tag{1}$$

where $\boldsymbol{n}_i \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{I}) \in \mathbb{C}^M$ is the additive noise and ρ_{tr} is the uplink training signal-to-noise ratio (SNR).

The $i-$th contaminated channel estimate $\hat{\boldsymbol{h}}_i$ is zero-mean with covariance matrix

$$\hat{\boldsymbol{R}}_i = \mathrm{E}\left[\hat{\boldsymbol{h}}_i\hat{\boldsymbol{h}}_i^{\mathrm{H}}\right] = \frac{1}{\sqrt{\rho_{\text{tr}}}}\mathbf{I} + \sum_{j=1}^{K} \boldsymbol{R}_j c_{ji} \tag{2}$$

when assuming the channels and the noise to be mutually independent.

After the training phase, the channel vectors are properly precoded in order to beamform the users' signals in the downlink. The beamforming vectors are given by

$$\boldsymbol{w}_i = \boldsymbol{A}_i \boldsymbol{L}_i^{-1}\hat{\boldsymbol{h}}_i. \tag{3}$$

Here, $\boldsymbol{A}_i$ is the CDI precoding matrix introduced in [7] and it only depends on $\boldsymbol{R}_i, i \in \{1, ..., K\}$, the covariance matrices of the K channel vectors. The channel estimate is whitened by the inverse of $\boldsymbol{L}_i$, which is defined by $\hat{\boldsymbol{R}}_i = \boldsymbol{L}_i\boldsymbol{L}_i^{\mathrm{H}}$, making it possible to satisfy the per-user power constraint

$$\mathrm{E}\left[\boldsymbol{w}_i^{\mathrm{H}}\boldsymbol{w}_i\right] = \boldsymbol{A}_i\boldsymbol{A}_i^{\mathrm{H}} \leq 1 \tag{4}$$

or alternative power constraints.

As explained in [7], CDI precoding of least-squares channel estimates exploits the benefits of the asymptotic behavior of massive MIMO, and, employing a very large number of antennas, it should be able to outperform classical matched filter precoding of MMSE channel estimates.

III. Achievable rate

The base station weights every beamformer with the information signal $s_i \sim \mathcal{N}_{\mathbb{C}}(0, 1)$ of the corresponding user and transmits:

$$\boldsymbol{x} = \sqrt{\rho_{\text{dl}}} \sum_{i=1}^{K} \boldsymbol{w}_i s_i. \tag{5}$$

The normalized received signal for the i-th user reads as

$$y_i = \boldsymbol{h}_i^{\mathrm{H}} \sum_{j=1}^{K} \boldsymbol{w}_j s_j + \frac{n_i}{\sqrt{\rho_{\text{dl}}}} \tag{6}$$

where $n_i \sim \mathcal{N}_{\mathbb{C}}(0, 1)$ is the additive noise and ρ_{dl} the downlink SNR.

Following the approach of [7], we evaluate the lower bound for the achievable rate introduced in [6]. It is possible indeed to compute the average and the variance of the total channel $\mathrm{E}\left[\boldsymbol{h}_i^{\mathrm{H}}\boldsymbol{w}_i\right]$. The lower bound to the rate is based on

$$\begin{aligned} y_i = &\overbrace{\mathrm{E}\left[\boldsymbol{h}_i^{\mathrm{H}}\boldsymbol{w}_i\right]}^{\text{Avg channel}} s_i + \overbrace{\left[\boldsymbol{h}_i^{\mathrm{H}}\boldsymbol{w}_i - \mathrm{E}\left[\boldsymbol{h}_i^{\mathrm{H}}\boldsymbol{w}_i\right]\right]}^{\text{Difference w.r.t. avg}} s_i \\ &+ \overbrace{\sum_{\substack{j=1 \\ j\neq i}}^{K} \boldsymbol{h}_i^{\mathrm{H}}\boldsymbol{w}_j s_j}^{\text{Intra-cell IF}} + \overbrace{\frac{n_i}{\sqrt{\rho_{\text{dl}}}}}^{\text{Noise}}. \end{aligned} \tag{7}$$

The resulting effective SINR expression for the i-th user can be written as

$$\gamma_i = \frac{\left|\mathrm{E}\left[\boldsymbol{h}_i^{\mathrm{H}}\boldsymbol{w}_i\right]\right|^2}{\frac{1}{\rho_{\text{dl}}} + \mathrm{var}\left[\boldsymbol{h}_i^{\mathrm{H}}\boldsymbol{w}_i\right] + \sum_{\substack{j=1 \\ j\neq i}}^{K} \mathrm{E}\left[\left|\boldsymbol{h}_i^{\mathrm{H}}\boldsymbol{w}_j\right|^2\right]}. \tag{8}$$

Incorporating the statistical properties of the channel yields [7]

$$\gamma_i = \frac{\left|\mathrm{tr}\left[\boldsymbol{R}_i\boldsymbol{A}_i\boldsymbol{L}_i^{-1}\right]\right|^2}{\frac{1}{\rho_{\text{dl}}} + \sum_{j=1}^{K}\mathrm{tr}\left[\boldsymbol{R}_i\boldsymbol{A}_j\boldsymbol{A}_j^{\mathrm{H}}\right] + \sum_{\substack{j=1 \\ j\neq i}}^{K}\left|\mathrm{tr}\left[\boldsymbol{R}_i\boldsymbol{A}_j\boldsymbol{L}_j^{-1}\right]c_{ji}\right|^2}. \tag{9}$$

Finally, based on [6], the lower bound of the achievable rate for user i is

$$r_i = \log_2(1 + \gamma_i)\left[\frac{\text{bit/s}}{\text{Hz}}\right]. \tag{10}$$

IV. Coordinated assignment algorithms

In order to formulate the pilot assignment problem, we allocate users with the same pilot sequence in separate sets, $\mathcal{S}^i, i = 1, ..., N$. We define $\mathcal{U}$, the set that contains all the users, $\mathcal{U} = \{1, ..., K\}$, and $\mathcal{G}$, the super-set that contains non-empty sets $\mathcal{S}^i$, $\mathcal{G} = \left\{\mathcal{S}^1, ..., \mathcal{S}^N\right\}$, $\mathcal{S}^i \neq \emptyset$. This means that there will be N sets, containing at most $N_{\text{g}} = \lceil \frac{K}{N} \rceil$ interfering users each.

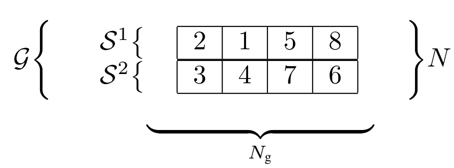

Fig. 1: Visual representation of the sets.

We will present several algorithms that are able to coordinately build the sets of interfering users such that a network utility function is maximized. To this end, we use a version of the greedy algorithm from [8], where at each step of the algorithm, we solve the NUM problem. The network utility

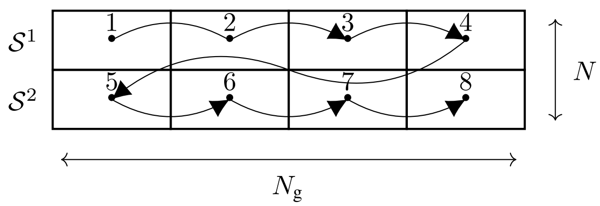

Fig. 2: Sequential Horizontal algorithm.

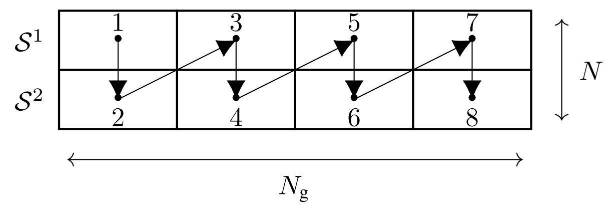

Fig. 3: Sequential Vertical algorithm.

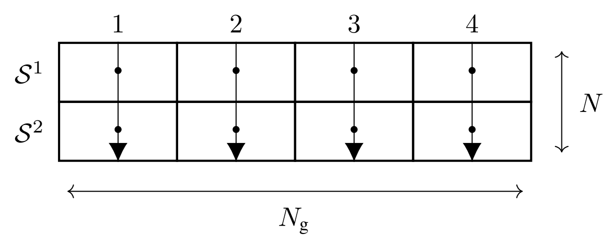

Fig. 4: Parallel Vertical algorithm.

function that we adopt is the lower bound of the achievable sum rate [cf. (10)], i.e.,

$$F(\mathcal{G}) \triangleq \sum_{\mathcal{S}^n \in \mathcal{G}} \sum_{u \in \mathcal{S}^n} \log_2 \left[1 + \gamma_u (\mathcal{G})\right]. \tag{11}$$

The first algorithm is called *Sequential Horizontal* (SH) and executes the following procedure. First, all the sets are initialized as empty sets, i.e., $\mathcal{S}^n = \emptyset$. Then, for every empty set, the first element among the unassigned users in $\mathcal{U}$ is chosen and temporarily added to the current set. Afterwards, the network utility function $F(\mathcal{G})$ is computed for every combination of the users in the set with every other unassigned user. The unassigned user that gives the biggest contribution to the sum rate, i.e., to the network utility function, will be selected and added to the current set. The next user that will be added to the current set $\mathcal{S}^n$ is,

$$u = \arg\max_{i \in \mathcal{U}} F\left(\left\{\mathcal{S}^1, ..., \mathcal{S}^n \cup \{i\}, \mathcal{S}^N\right\}\right) \tag{12}$$

This operation is repeated until every user from $\mathcal{U}$ is assigned to a set $\mathcal{S}^n$ and the superset $\mathcal{G}$ is full. The steps of the algorithm are summarized in Algorithm 1.

Algorithm 1 Sequential Horizontal algorithm

$\mathcal{S}^n \leftarrow \emptyset, n \in \{1, ..., N\}$ ▷ Initialize the empty sets
$\mathcal{U} = \{1, ..., K\}$ ▷ Initialize the set of available users
for $n = 1$ to N **do** ▷ For every set
  $u \leftarrow \text{rand}(\mathcal{U})$ ▷ Randomly pick a user
  $\mathcal{S}^n \leftarrow \mathcal{S}^n \cup \{u\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{u\}$
  for $i = 2 : N_g$ **do** ▷ For every other element
    $u \leftarrow \arg\max_{j \in \mathcal{U}} F\left(\left\{\mathcal{S}^1, ..., \mathcal{S}^n \cup \{j\}, \mathcal{S}^N\right\}\right)$
    ▷ Select the user that maximizes F
    $\mathcal{S}^n \leftarrow \mathcal{S}^n \cup \{u\}$
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \{u\}$
  end for
end for

Algorithm 1 allocates the best resources to the users in the sets visited first, and the worst resources to the users visited afterwards, therefore, this is an opportunistic pilot allocation strategy.

We will also simulate a second algorithm that takes decisions alternating the different sets of interfering users, the *Sequential Vertical* (SV). The complexity of the algorithm is the same as that of the previous one, but it allocates pilot resources in a fair way to the users of the different sets. The steps of this algorithm are summarized in Algorithm 2.

Algorithm 2 Sequential Vertical algorithm

$\mathcal{S}^n \leftarrow \emptyset, n \in \{1, ..., N\}$ ▷ Initialize the empty sets
$\mathcal{U} = \{1, ..., K\}$
for $n = 1$ to N **do** ▷ Initialize the first element of every set randomly
  $u \leftarrow \text{rand}(\mathcal{U})$
  $\mathcal{S}^n \leftarrow \mathcal{S}^n \cup \{u\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{u\}$
end for
for $i = 2 : N_g$ **do** ▷ For every other element of the set
  for $n = 1$ to N **do** ▷ For every set
    $u \leftarrow \arg\max_{j \in \mathcal{U}} F\left(\left\{\mathcal{S}^1, ..., \mathcal{S}^n \cup \{j\}, \mathcal{S}^N\right\}\right)$
    ▷ Select the user that maximizes F
    $\mathcal{S}^n \leftarrow \mathcal{S}^n \cup \{u\}$
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \{u\}$
  end for
end for

A further improvement of the algorithm is possible if we try to allocate more than one user at a time for every evaluation of the network utility function. In the *Parallel Vertical* (PV) algorithm, we will find the best allocation for N users every time. At each step, the algorithm will test all the permutations of the possible combinations of groups of N users from the available users. With this approach, we increase the complexity of our algorithm, seeking for better results. The steps of this algorithm are summarized in Algorithm 3.

Finally, we implement as a reference the algorithm that explores all the possible pilot allocations, the *Exhaustive Search* (ES) algorithm. This algorithm will always be able to find the best pilot assignment, and its results will be used to determine the performance gap introduced by the greedy algorithms. The steps of the exhaustive search algorithm are summarized in Algorithm 4.

Algorithm 3 Parallel Vertical algorithm

$\mathcal{S}^n \leftarrow \emptyset, n \in \{1, ..., N\}$ ▷ Initialize the empty sets
$\mathcal{U} = \{1, ..., K\}$
for $n = 1$ to N **do** ▷ Randomly assign the first user of every set
  $u \leftarrow \text{rand}(\mathcal{U})$
  $\mathcal{S}^n \leftarrow \mathcal{S}^n \cup \{u\}, \mathcal{U} \leftarrow \mathcal{U} \setminus \{u\}$
end for
for $i = 2 : N_{\text{g}}$ **do** ▷ For every other element of the sets
  $\mathcal{C} \leftarrow \text{Comb}(N, \mathcal{U})$
  ▷ Group combinations of N users from $\mathcal{U}$
  $\mathcal{P} \leftarrow \text{Perm}(\mathcal{C})$ ▷ Permute all the combinations
  $u \leftarrow \arg\max_{p \in \mathcal{P}} F(\mathcal{G} \cup p)$
  ▷ Select the combination that maximizes F
  $\mathcal{G} \leftarrow \mathcal{G} \cup \{c\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{c\}$
end for

Algorithm 4 Exhaustive Search algorithm

$\mathcal{U} = \{1, ..., K\}$ ▷ Initialize the set of available users
$\mathcal{C} \leftarrow \text{Comb}(K, \mathcal{U})$ ▷ Generate all the combinations of K users from $\mathcal{U}$
$\mathcal{P} \leftarrow \text{Perm}(\mathcal{C})$ ▷ Permute all the combinations of $\mathcal{C}$
$\mathcal{G} \leftarrow \arg\max_{c \in \mathcal{P}} F(\mathcal{G} \cup \{p\})$ ▷ Select the combination that maximizes F

V. Complexity

Before discussing the benefits of the pilot coordination strategies, it is interesting to calculate the computational complexity of the several algorithms. It turns out, that it depends only on the number of users K and the *degree of orthogonality*:

$$\alpha = \frac{N}{K} \qquad \alpha \in \left\{ \frac{1}{K}, \frac{2}{K}, ..., 1 \right\} \tag{13}$$

To better understand how α and K affect the computational complexity, we consider the case $N_{\text{g}} \in \mathbb{N}$. The exact calculations with arbitrary values of N_{g} are shown in the Appendix. We can write N and N_{g} as:

$$N_{\text{g}} = \frac{K}{N} = \frac{1}{\alpha} \qquad N = \alpha K. \tag{14}$$

The complexity of the Sequential Vertical and Sequential Horizontal algorithms is similar: they both allocate one user per iteration. The total number of sum rate evaluations for the Sequential Horizontal algorithm is [cf. (25)]

$$N_{\text{SR}}^{\text{SH}} = \frac{K^2 [1 - \alpha]}{2} \sim \begin{cases} 0 & : \alpha = 1 \\ K^2 & : \alpha = \frac{1}{K}. \end{cases} \tag{15}$$

As we can see from above, the complexity of the sequential algorithms strongly depends on the degree of orthogonality. With many orthogonal resources the complexity grows linearly with the number of users. On the other hand, with few orthogonal pilots, the complexity assumes a quadratic behavior with respect of K.

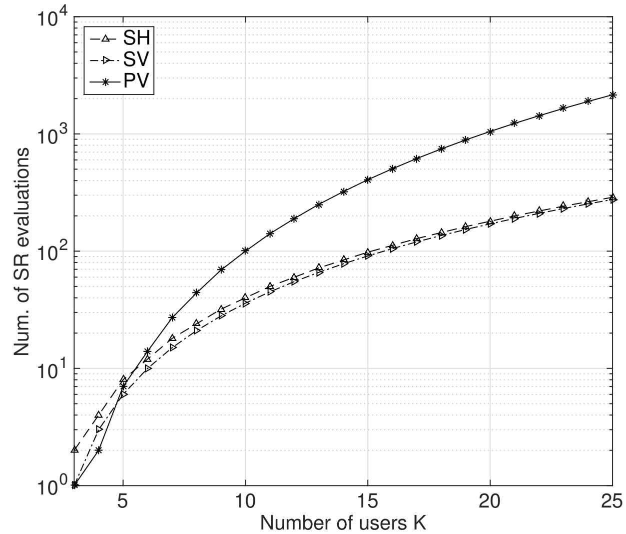

Fig. 5: Average computing time of the greedy algorithms, CDI MF precoder.

We will repeat the same analysis for the Parallel Vertical greedy algorithm [cf. (27)]

$$N_{\text{SR}}^{\text{PV}} = \sum_{i=1}^{\frac{1}{\alpha}-1} \binom{\alpha K}{K[1 - i\alpha]} [\alpha K]! \sim \begin{cases} K! & : \alpha = 1 \\ \sum_{i=1}^{K-1} \binom{1}{K-i} & : \alpha = \frac{1}{K}. \end{cases} \tag{16}$$

The complexity of the Parallel Vertical algorithm behaves in an opposite way with respect of the sequential algorithms. The less orthogonal resources N, the smaller number of operations will the algorithm compute per iteration, and, in the limiting case of $N = 1$, it behaves exactly as a sequential algorithm. On the contrary, when the number of interfering sets grows, i.e., more orthogonal resources are present, the algorithm will calculate more operations, reaching the limiting case of $K!$ in presence of full orthogonal pilots $N = K$. In this case, it calculates all the possible combinations as the exhaustive algorithm does.

Finally, we can visually compare the computational complexity of the different algorithms in a Matlab simulation. In the next figures, it is possible to see the number of sum rate evaluations for the different algorithms as the number of users grows.

In Fig.5, we can observe the behavior of the greedy algorithms only. As anticipated, the complexity of the sequential algorithms is similar, while the Parallel Vertical is more complex and requires more evaluations for $K \geq 5$.

Next, in Fig.6, we compare the greedy algorithms with the Exhaustive Search algorithm. The number of operations of the non-greedy algorithm is huge, for this reason, it is sufficient to compute the number of evaluations up to $K = 8$ users to understand its weakness with respect to the greedy algorithms.

As we can see, the Exhaustive Search's complexity terribly scales with K, on the contrary, all the greedy algorithms can

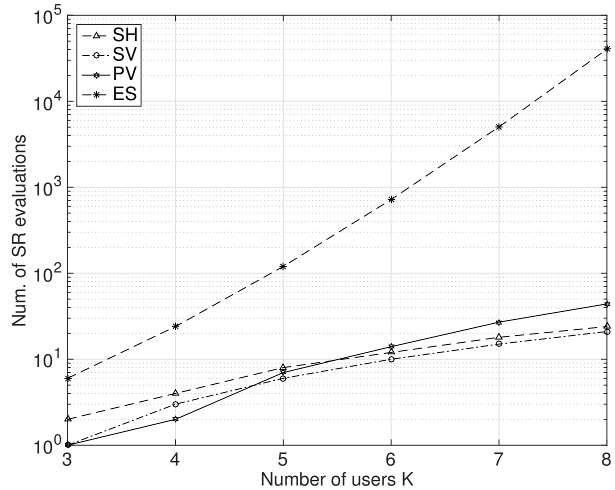

Fig. 6: Average computing time all the algorithms, CDI MF precoder.

efficiently adapt to an increasing number of users.

VI. Results

In this section, the performance of the greedy algorithm is evaluated in a well defined scenario. In order to compute the achievable sum rate, we use the CDI *Zero Forcing* matrix $\boldsymbol{A}_i^{\mathrm{ZF}}$ and *Matched Filter* $\boldsymbol{A}_i^{\mathrm{MF}}$ from [7] with

$$\boldsymbol{A}_i^{\mathrm{MF}} = \boldsymbol{R}_i \tag{17}$$

$$\boldsymbol{A}^{\mathrm{ZF}} = \boldsymbol{R}^{+} \tag{18}$$

$$\boldsymbol{A}^{\mathrm{ZF}} = [\boldsymbol{a}_1, ..., \boldsymbol{a}_K], \boldsymbol{a}_i = \mathrm{vect}\left(\boldsymbol{A}_i^{\mathrm{ZF}}\right) \tag{19}$$

$$\boldsymbol{R} = [\boldsymbol{r}_1, ..., \boldsymbol{r}_K]^{\mathrm{H}}, \boldsymbol{r}_i = \mathrm{vect}\left(\boldsymbol{R}_i\right). \tag{20}$$

We adopt the channel model used in [7], a multi-path model for a receiving ULA at the base station. Every channel vector includes fast and slow fading effects. The fast fading is the result of the evaluation of a big number of paths in different points of the M antenna ULA. The number of paths for every channel vector is $5M$ and every path is characterized by an angle of arrival (AoA) θ_{ip} and fading coefficient $\alpha_{ip} \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma_{ip}^2\right)$, whose values are generated according to the ITU-R guidelines in [3]. This model takes into account the increasing spatial resolution of the array, as the array grows, it will also be able to distinguish more micro-paths. The slow fading is described by the path-loss model, i.e.,

$$\beta_i = \left(\frac{d_i}{R}\right)^{-\delta_{\mathrm{PL}}} \tag{21}$$

where d_i is the distance of the i-th user from the base station and its position is uniformly distributed in the hexagonal cell. Moreover, R is the cell radius and δ_{PL} is the path-loss exponent.

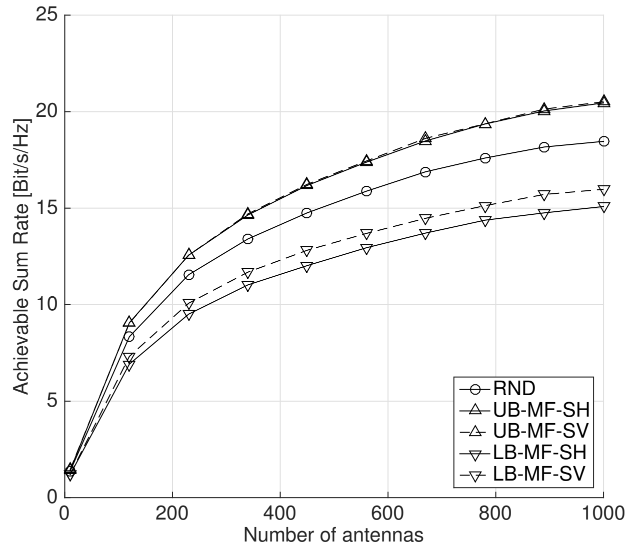

Fig. 7: All algorithms, MF, K=10, N=2

The i-th channel vector is:

$$\boldsymbol{h}_i = \sqrt{\beta_i}\sum_{p=1}^{5M}\boldsymbol{b}\left(\theta_{ip}\right)\alpha_{ip} = \sqrt{\beta_i}\boldsymbol{B}_i\boldsymbol{\alpha}_i \tag{22}$$

with the array steering vector $\boldsymbol{b}\left(\theta_{ip}\right) \in \mathbb{C}^M$. The resulting covariance matrix is

$$\boldsymbol{R}_i = \beta_i\boldsymbol{B}_i\,\mathrm{diag}\left(\sigma_{i1}^2, ..., \sigma_{i5M}^2\right)\boldsymbol{B}_i^{\mathrm{H}}. \tag{23}$$

In Fig. 7 and 8, we simulate $K = 10$ users, $N = 2$ orthogonal pilots and the same SNR $\rho_{\mathrm{dl}} = 1$ for downlink, respectively, CDI MF and ZF. The plots show the achievable sum rate versus the number of base station antennas, for all the greedy algorithms. The result of the NUM based assignment problem (UB) is compared to a random assignment of pilots (RND) and a lower bound (LB) for the coordination problem. The random allocation represents the average case of pilot coordination when the resources are randomly assigned. The lower bound is calculated by minimizing the network utility function, it allocates pilots to the user with the least contribution to the sum rate. This curve shows the worst-case condition in pilot coordination.

As we can see, the random assignment performs in the same way for all the algorithms, the sequential algorithms achieve similar results and the Parallel Vertical always provides a modest gain over them.

In Fig. 9 and 10 we want to compare the performances of the Parallel Vertical algorithm with the Exhaustive Search. Due to the high complexity of the Exhaustive algorithm, we will simulate only $K = 6$ users, while all the other parameters will remain the same.

As expected, the Exhaustive Search is able to further increase the results of the Parallel Vertical, selecting always the best pilot allocation. On the other hand, we can notice that

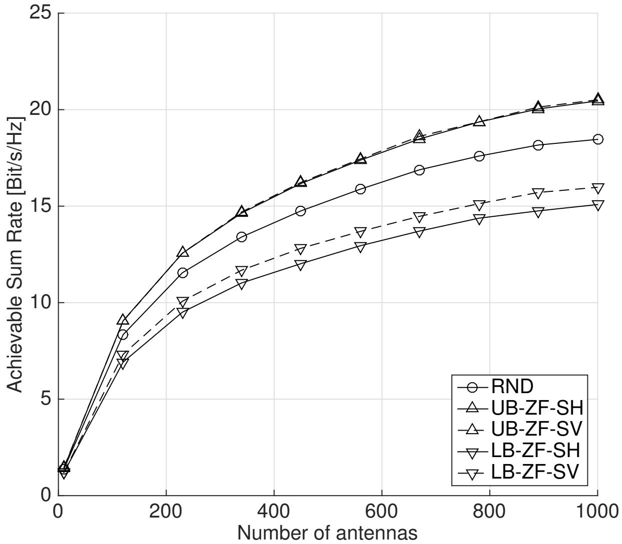

Fig. 8: All algorithms, ZF, K=10, N=2

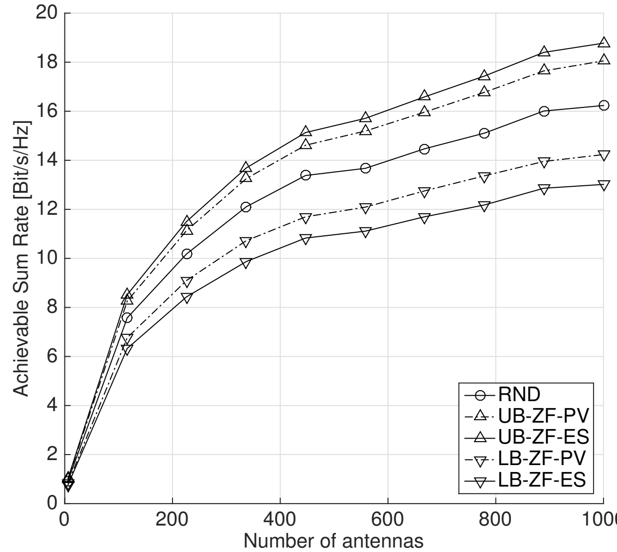

Fig. 10: Parallel Vertical(PV) and Exhaustive Search(ES), ZF, K=6, N=2

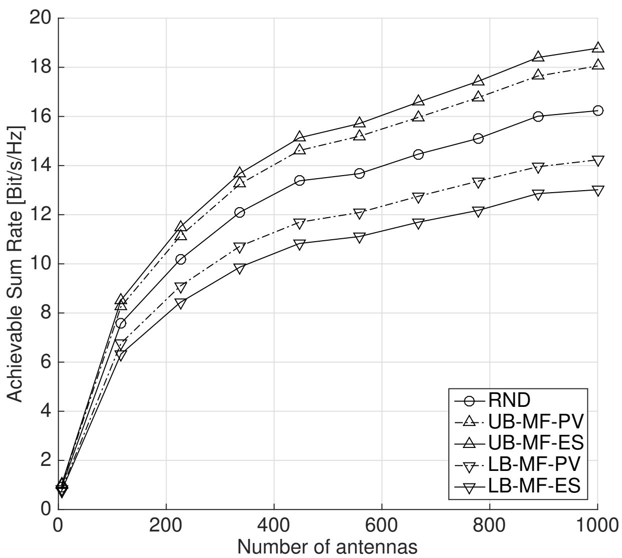

Fig. 9: Parallel Vertical (PV) and Exhaustive Search(ES), MF, K=6, N=2

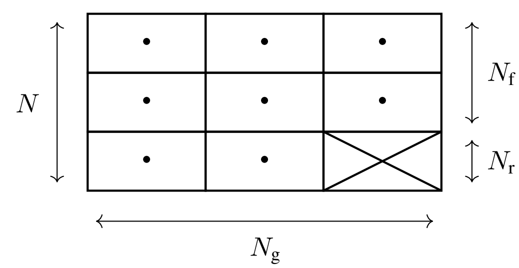

Fig. 11: Model used to calculate the complexity.

difference between the upper bounds of the two algorithms depends on the precoder, and it is very small for ZF with respect to the MF. In general, we observe a bigger bounded region using the MF rather than the ZF.

VII. Conclusion and further research

This paper shows the effect of coordinated pilot assignment in CDI precoding. We can notice that the proposed method considerably improves performances with respect to the average case. At the same time it avoids worst-case situations where the sum rate drastically reduces.

The sequential algorithms provide a good solution in terms of scalability with the number of user, especially with a large degree of orthogonality. The parallel vertical greedy algorithm represents an intermediate step between the sequential and the full search algorithm. It always provides a modest performance gain over the sequential algorithms, and good complexity for a low degree of orthogonality. The exhaustive search of the maximum provides a good gain in performances, but is completely unfeasible from the complexity point of view, and it is hard for it to find application in real systems.

A further inspection to be considered is the application of the discussed solution for performance improvements of different strategies, such as the cooperation-based ones. We will also perform tests on different CDI precoders and on predefined patterns of the channel's covariance matrix.

Appendix

We calculate here the the number of sum rate evaluations performed by the algorithms. To fully analyze the complexity, as shown in Fig. 11, we define two additional variables

$$N_{\mathrm{f}} = K - \left[N_{\mathrm{g}} - 1\right] N \qquad N_{\mathrm{r}} = N - N_{\mathrm{f}} \tag{24}$$

It is possible to write the number of sum rate evaluations for the Sequential Horizontal algorithm as

$$N_{\text{SR}}^{\text{SH}} = \sum_{i=0}^{N_{\text{f}}N_{\text{g}}-1} K - i + \sum_{i=N_{\text{f}}N_{\text{g}}}^{N_{\text{f}}N_{\text{g}}+N_{\text{r}}[N_{\text{g}}-1]-1} K - i - \sum_{i=0}^{N_{\text{f}}} K - iN_{\text{g}} - \sum_{i=1}^{N_{\text{r}}-1} K - \Big[N_{\text{f}}N_{\text{g}} + i\left[N_{\text{g}}-1\right]\Big] \quad (25)$$

In the same way, the number of sum rate evaluations for the Sequential Vertical algorithm is

$$N_{\text{SR}}^{\text{SV}} = \sum_{i=N}^{N[N_{\text{g}}-1]} K - i + \sum_{i=1}^{N_{\text{f}}-1} i \quad (26)$$

The number of sum rate evaluations for the Parallel Vertical algorithm is:

$$N_{\text{SR}}^{\text{PV}} = \sum_{i=1}^{N_{\text{g}}-2} \binom{N}{K-iN} N! + \binom{N_{\text{f}}}{K-[N_{\text{g}}-1]N} N_{\text{f}}! \quad (27)$$

Finally, the number of sum rate evaluations for the Exhaustive Search algorithm is

$$N_{\text{SR}}^{\text{Ex}} = K! \quad (28)$$

References

[1] A. Ashikhmin and T. Marzetta. "Pilot contamination precoding in multi-cell large scale antenna systems". In *International Symposium on Information Theory Proceedings (ISIT), 2012 IEEE*, pages 1137–1141, July 2012.

[2] M. Guillaud, D.T.M. Slock, and R. Knopp. "A practical method for wireless channel reciprocity exploitation through relative calibration". In *Proceedings of the Eighth International Symposium on Signal Processing and Its Applications, 2005.*, volume 1, pages 403–406, August 2005.

[3] ITU-R. Guidelines for evaluation of radio interface technologies for IMT-advanced. Technical report, International Telecommunication Union (ITU), ITU-R M.2135-1, 2009.

[4] J. Jose, A. Ashikhmin, T.L. Marzetta, and S. Vishwanath. "Pilot Contamination and Precoding in Multi-Cell TDD systems". *IEEE Transactions on Wireless Communications*, 10(8):2640–2651, August 2011.

[5] T.L. Marzetta. "Noncooperative Cellular Wireless with Unlimited numbers of base station antennas". *IEEE Transactions on Wireless Communications*, 9(11):3590–3600, November 2010.

[6] M. Medard. "The effect upon channel capacity in wireless communications of perfect and imperfect knowledge of the channel". *IEEE Transactions on Information Theory*, 46(3):933–946, May 2000.

[7] David Neumann, Michael Joham, and Wolfgang Utschick. "CDI precoding for massive MIMO". In *SCC 2015; 10th International Systems, Communications and Coding*, pages 1–6, Feb 2015.

[8] Haifan Yin, D. Gesbert, M. Filippou, and Yingzhuang Liu. "A Coordinated Approach to Channel Estimation in Large-Scale Multiple-Antenna Systems". *IEEE Journal on Selected Areas in Communications*, 31(2):264–273, February 2013.