Signal Processing 175 (2020) 107666

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Efficient pilot allocation for sparse channel estimation in UWB OFDM systems

Taoyong Li[a,b,*], Nele Noels[a], Heidi Steendam[a]

[a] Telecommunications and Information Processing Department of Ghent University / IMEC, Ghent, Belgium
[b] Collaborative Innovation Center of Information Sensing and Understanding, Xi'an, China

ARTICLE INFO

Article history:
Received 4 February 2020
Revised 18 May 2020
Accepted 21 May 2020
Available online 23 May 2020

Keywords:
Ultra-wideband
Sparse channel estimation
Orthogonal frequency-division multiplexing
Measurement matrix mutual coherence
Pilot allocation

ABSTRACT

In ultra-wideband (UWB) orthogonal frequency-division multiplexing (OFDM) systems, compressive sensing (CS) is often employed to produce an estimate of the typically sparse channel from the observation of a limited number of allocated pilot subcarriers. Moreover, it is well established that proper pilot subcarrier allocation (PA) design is the key for CS-based channel estimates to achieve a good performance. In this paper, we propose three novel PA algorithms. These are referred to as deterministic greedy algorithm (DGA), simplified deterministic greedy algorithm (SDGA) and simulated annealing (SA). The proposed algorithms are attractive alternatives for the state-of-the-art method because they are orders of magnitude less complex but yield only a small system performance degradation (SDGA), or they yield a better system performance as well as a lower complexity (DGA and SA).

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Ultra-wideband (UWB) communication has recently received a lot of attention from both scientific community and industry [1–6]. In UWB communication, multi-band orthogonal frequency-division multiplexing (MB-OFDM) is considered as one of the most promising techniques to achieve high-data-rate transmission at a relatively low cost [7,8]. Practical UWB MB-OFDM systems require channel estimation at the receiver. As the UWB channel can often be modeled as sparse, channel estimation methods based on compressive sensing (CS) can be used to obtain an accurate estimate of the channel impulse response (CIR) with low overhead (i.e., from a small amount of pilot subcarriers) [9–13].

CS enables us to accurately recover a signal from a small number of linear measurements if the signal itself is sparse or sparse in a certain transform domain. The performance of the sparse vector recovery is determined by three important aspects of CS applications, which are the reconstruction algorithm, the dictionary matrix and the measurement matrix. In [14–20], several algorithms are proposed to effectively reconstruct the sparse signal. Any of these algorithms can be applied to estimate sparse channels. The dictionary matrix is intended to make a signal sparse in a specific domain. Owing to the inherent sparsity of the CIR, an identity matrix is usually adopted as the dictionary matrix for sparse channel estimation [21–23]. As for the measurement matrix design, it was shown in [24] that a small measurement matrix mutual coherence (MMMC) is desirable (i.e., to have a measurement matrix whose columns have small normalized mutual correlations). In the case of pilot subcarrier based channel estimation, the measurement matrix is fully determined by the pilot pattern (i.e., the position and the value of the pilots); hence, the measurement matrix design boils down to pilot pattern design. For the sake of simplicity and tractability, one often imposes that all pilots have the same amplitude. As the MMMC only depends on the amplitude of the pilots and not on their phases, the measurement matrix design then further reduces to a pure pilot subcarrier allocation (PA) problem, which is in itself a combinatorial optimization problem. PAs corresponding to cyclic difference sets (CDS) are known to yield minimum MMMC [25–29]; however, in most practical situations a CDS PA does not exist. Because an exhaustive search for the PA that minimizes the MMMC is usually too complex, several approximation algorithms have been put forward over the last five years [25–28,30–36]. The state-of-the-art PA design method is the stochastic sequential search (SSS) algorithm from Qi et al. [27]. SSS combines a limited complexity with a relatively good performance. Nevertheless, the search for more efficient PA design algorithms continues. In this paper, we propose three new PA algorithms, and we demonstrate their superiority with respect to SSS. The rest of the paper is organized as follows. The OFDM UWB communication system is described in Section 2. The novel PA methods are derived in Section 3. Numerical results are pro-

[*] Corresponding author at: Telecommunications and Information Processing Department of Ghent University/IMEC, Ghent, Belgium.
E-mail address: Taoyong.Li@ugent.be (T. Li).

https://doi.org/10.1016/j.sigpro.2020.107666
0165-1684/© 2020 Elsevier B.V. All rights reserved.

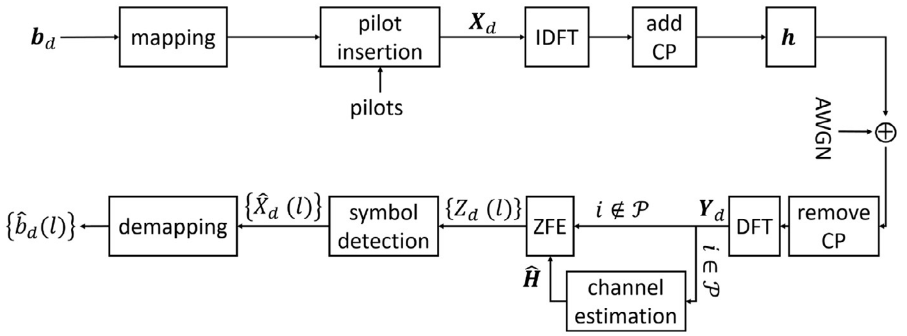


Fig. 1. A block diagram of the system.

vided in Section 4 and a summarizing conclusion is provided in Section 5.

2. System description

We consider a packet-based MB-OFDM implementation of an UWB transmission system. A MB-OFDM packet consists of concatenated OFDM symbols that form a known preamble/header section, followed by a payload. The preamble can be employed for initial packet synchronization. The payload is primarily intended to convey the data. It will be further assumed to consist of D OFDM symbols. Traditionally, pilots are added to the payload data in order to provide a-priori information to the channel estimator. In the following, we will assume that in each payload symbol N_p out of the N subcarriers are allocated to pilot transmission. Obviously, channel estimation accuracy can be expected to improve if the ratio $\frac{N_p}{N}$ increases, but for the sake of power and bandwidth efficiency, the ratio $\frac{N_p}{N}$ should be kept as small as possible. In [37], it is argued that in MB-OFDM UWB the coherence time of the channel is typically several times larger than the maximum transmission time of a packet. This means that the channel can be considered as constant over the packet and that the channel estimator can base its outcome on all the $D \cdot N_p$ pilots in the entire packet.

A block diagram of the considered communication system is shown in Fig. 1. The dth OFDM symbol in the packet payload (with $d = 1, 2, \ldots, D$) conveys $m\left(N - N_p\right)$ information bits $\boldsymbol{b}_d = \left[b_d(1), b_d(2), \ldots, b_d\left(m\left(N - N_p\right)\right)\right]$, with $b_d(i) \in \{0, 1\}$. At the transmitter, the sequence $\boldsymbol{b}_d$ is divided into $\left(N - N_p\right)$ groups of m bits that are bijectively mapped to the 2^m elements of a complex-valued constellation $\Omega = \{\omega_1, \omega_2, \ldots, \omega_{2^m}\}$ with zero mean and average symbol energy $E_s = \frac{1}{2^m}\sum_{i=1}^{2^m} |\omega_i|^2$. The resulting $\left(N - N_p\right)$ data symbols are subsequently interweaved with N_p pilots with amplitude $\sqrt{E_s}$ and known phase. The resulting sequence $\boldsymbol{X}_d = [X_d(1), X_d(2), \ldots, X_d(N)]^T$ is used to modulate the N subcarriers of the dth OFDM symbol. If i is an element of $\mathcal{P}_d = \{p_{d,1}, p_{d,2}, \ldots, p_{d,N_p}\}$, $X_d(i)$ is a pilot, otherwise, it is a data symbol. To construct the dth OFDM symbol, the vector $\boldsymbol{X}_d$ is first applied to an N-point inverse discrete Fourier transform (IDFT) unit, after which a cyclic prefix (CP) is added.

The UWB CIR is represented by a sparse random vector $\boldsymbol{h} = [h(1), h(2), \ldots, h(L_d)]^T$, with $K \ll L$ non-zero elements. It is assumed that $\boldsymbol{h}$ remains fixed over the duration of a packet. The receiver removes the CP and performs the inverse N-point discrete Fourier transform (DFT) operation. Assuming that the CP is longer than L, the vector $\boldsymbol{Y}_d = [Y_d(1), Y_d(2), \ldots, Y_d(N)]^T$ at the DFT output can be modeled as:

$$\boldsymbol{Y}_d = \mathrm{diag}(\boldsymbol{X}_d)\boldsymbol{H} + \boldsymbol{W}_d, \tag{1}$$

where $\boldsymbol{W}_d = [W_d(1), W_d(2), \ldots, W_d(N)]^T \sim \mathcal{CN}\left(\boldsymbol{0}, \sigma^2\boldsymbol{I}_N\right)$ is additive white Gaussian noise (AWGN) at dth OFDM symbol time, and $\boldsymbol{H} = [H(1), H(2), \ldots, H(N)]^T$ is defined as:

$$\boldsymbol{H} = \boldsymbol{F}\boldsymbol{h}, \tag{2}$$

with $\boldsymbol{F}$ the $N \times L$ matrix with elements $F_{u,v} = \frac{1}{\sqrt{N}}e^{-j2\pi\frac{(u-1)(v-1)}{N}}$, $u = 1, 2, \ldots, N$ and $v = 1, 2, \ldots, L$.

The DFT output $\boldsymbol{Y}_d$ passes through a zero-forcing equalizer (ZFE) to neutralize the effect of the channel. First, the pilot symbol observations $Y_d(i)$, with $i \in \mathcal{P}_d$ and $d \in \{1, 2, \ldots, D\}$, are used to generate an estimate $\hat{\boldsymbol{h}}$ of $\boldsymbol{h}$. Then, the information symbol observations $Y_d(i)$, with $i \notin \mathcal{P}_d$ are divided by $\hat{H}(l)$, with $\hat{H}(l)$ denoting the lth element of the vector $\hat{\boldsymbol{H}} = \boldsymbol{F}\hat{\boldsymbol{h}}$. Next, for $l \notin \mathcal{P}_d$, the symbol decision unit recovers $X_d(l)$ from $Z_d(l) = \frac{Y_d(l)}{\hat{H}_d(l)}$, using a decision rule that is optimum for $\hat{H}(l)$ equal to $H(l)$. Finally, estimates $\left\{\hat{b}_d(i)\right\}$ of the conveyed information bits $\{b_d(i)\}$ are obtained by demapping of the resulting symbol estimates $\left\{\hat{X}_d(l) = \arg\min_{\omega\in\Omega} \|Z_d(l) - \omega\|_2\right\}$.

A good measure for the overall system performance is the average BER, i.e.,

$$\mathrm{BER} = \mathbb{E}\left[\mathrm{BER}|_{\boldsymbol{H},\boldsymbol{W},\boldsymbol{b}}\right], \tag{3}$$

where $\mathbb{E}[\cdot]$ denotes the statistical expectation with respect to the joint distribution of $\boldsymbol{H}$, $\boldsymbol{W} = (\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_D)$ and $\boldsymbol{b} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_D)$, and

$$\mathrm{BER}|_{\boldsymbol{H},\boldsymbol{W},\boldsymbol{b}} = \frac{1}{m(N - N_p)D}\sum_{d=1}^{D}\sum_{i=1}^{m(N-N_p)}\left|\hat{b}_d(i) - b_d(i)\right|. \tag{4}$$

The BER of a practical system is lower bounded by the BER of a virtual system that has perfect channel knowledge at the receiver. The use of inaccurate channel estimates in the equalization stage of the receiver may significantly degrade the BER.

The accuracy of the channel estimator itself is usually measured in terms of the mean squared error (MSE), i.e.,

$$\mathrm{MSE} = \mathbb{E}\left[\mathrm{MSE}|_{\boldsymbol{H},\boldsymbol{W}}\right], \tag{5}$$

with

$$\mathrm{MSE}|_{\boldsymbol{H},\boldsymbol{W}} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{H}(i) - H(i)\right|^2 \tag{6}$$

and where $\mathbb{E}[\cdot]$ denotes averaging over the statistics of the channel and the noise.

The sparsity of the channel at hand allows to adopt a CS based channel estimation methods [14–20]. This in turn makes it possible to accurately estimate $\boldsymbol{h}$ from a very small number of pilots per packet, or equivalently, by using a very small ratio $\frac{N_p}{N}$) [20–23]. The rationale behind CS based channel estimation is to find $\hat{\boldsymbol{h}}$ so that

$$\hat{\boldsymbol{h}} = \arg\min \|\boldsymbol{h}\|_1 \text{ s.t. } \boldsymbol{Y}^{(\text{pilot})} = \boldsymbol{\Phi}\boldsymbol{h} + \boldsymbol{W}^{(\text{pilot})}, \tag{7}$$

where $\boldsymbol{Y}^{(\text{pilot})} = [Y_1(p_{1,1}), \ldots, Y_1(p_{1,N_p}), \ldots, Y_d(p_{d,1}), \ldots, Y_D(p_{D,N_p})]^T$, $\boldsymbol{W}^{(\text{pilot})} = [W_1(p_{1,1}), \ldots, W_1(p_{1,N_p}), \ldots, W_d(p_{d,1}), \ldots, W_D(p_{D,N_p})]^T$, and

$$\boldsymbol{\Phi} = \text{diag}\left(\boldsymbol{X}^{(\text{pilot})}\right)\boldsymbol{F}^{(\text{pilot})}, \tag{8}$$

with $\boldsymbol{X}^{(\text{pilot})} = [X_1(p_{1,1}), \ldots, X_1(p_{1,N_p}), \ldots, X_d(p_{d,1}), \ldots, X_D(p_{D,N_p})]^T$ and $\boldsymbol{F}^{(\text{pilot})}$ the $DN_p \times L$ matrix with elements $F_{u,v}^{(\text{pilot})} = \frac{1}{\sqrt{N}} e^{-j2\pi \frac{(p_u-1)(v-1)}{N}}$, where $v = 1, 2, \ldots, L$ and

$$p_u = p_{\left\lceil \frac{p_u}{N_p} \right\rceil, \text{mod}(p_u-1, N_p)+1}, \tag{9}$$

for $u = 1, 2, \ldots, DN_p$. In CS theory, $\boldsymbol{\Phi}$ is termed the 'measurement matrix' because it specifies the relationship between the measurements $\boldsymbol{Y}^{(\text{pilot})}$ and the sparse signal $\boldsymbol{h}$ to be recovered from $\boldsymbol{Y}^{(\text{pilot})}$. For (7) to be accurate, it is imperative that $\boldsymbol{\Phi}$ is an almost orthogonal matrix [9,10]. A suitable measure for the orthogonality of $\boldsymbol{\Phi}$ is its mutual coherence $\mu(\boldsymbol{\Phi})$, which is defined as [24]:

$$\mu\{\boldsymbol{\Phi}\} = \max_{1 \le m < n \le L} \frac{|\langle \boldsymbol{\phi}_m, \boldsymbol{\phi}_n \rangle|}{\|\boldsymbol{\phi}_m\|_2 \|\boldsymbol{\phi}_n\|_2}, \tag{10}$$

where $\boldsymbol{\phi}_m$ denotes the mth column of $\boldsymbol{\Phi}$ and $\langle \cdot \rangle$ represents the scalar product operation. The lower the measurement matrix mutual coherence (MMMC) is, the better. Using (8) and taking into account that $\left|X_d(p_{d,i})\right|^2 = E_s$ for $i = 1, 2, \ldots, N_p$, we have

$$\begin{aligned} \mu\{\boldsymbol{\Phi}\} &= \max_{1 \le m < n \le L} \frac{1}{DN_p} \left| \sum_{i=1}^{DN_p} e^{-j2\pi \frac{(p_i-1)(m-n)}{N}} \right| \\ &= \max_{1 \le q \le L-1} \frac{1}{DN_p} \left| \sum_{i=1}^{DN_p} e^{-j2\pi \frac{(p_i-1)q}{N}} \right|. \end{aligned} \tag{11}$$

It follows from (11) that, for given L, given N and given DN_p, $\mu\{\boldsymbol{\Phi}\}$ is fully determined by $\mathcal{P} = \{p_1, p_2, \ldots, p_{DN_p}\}$. The set $\mathcal{P}$ indicates which subcarriers are allocated to pilot symbols and is further simply referred to as the pilot allocation (PA). Using the Cauchy-Schwarz inequality, it is easily seen that (11) is upper bounded by 1, with equality if and only if either column $\boldsymbol{\phi}_m$ or $\boldsymbol{\phi}_n$ is a multiple of the other (which includes the special case that either of them is the zero vector). It immediately follows that it is not a good idea to have $p_u = p_{u'}$ for any $u \neq u'$, in $\mathcal{P}$. Hence, without loss of optimality, we will further assume that $\mathcal{P}$ consists of DN_p different subcarrier indices ranging from 1 to N.

The remainder of this paper focuses on designing a PA with a low MMMC. As the MMMC is known to be lower bounded by the Welch bound [25,29,38], we will always have:

$$\sqrt{\frac{N - DN_p}{DN_p(N-1)}} \le \mu\{\boldsymbol{\Phi}\} \le 1. \tag{12}$$

3. Novel PA methods

Finding the PA $\mathcal{P}$ that minimizes the MMMC $\mu\{\boldsymbol{\Phi}\}$ is essentially a combinatorial optimization problem:

$$\mathcal{P}_{opt} = \underset{\mathcal{P}}{\text{argmin}}\, \mu\{\boldsymbol{\Phi}\} \text{ s.t. } \mathcal{P} \subset \mathcal{N} = \{1, 2, \ldots, N\}. \tag{13}$$

In most practical situations (for the majority of (L, N, DN_p) combinations), no closed-form solution of (13) exists, so that approximate solution methods have to be used. Otherwise, if we want to find the optimal PA, we need to calculate the MMMC of all possible PAs, which quickly becomes impractical (even for offline computation). For example, more than 10^{40} PAs need to be tested for every L, if $N = 256$ and $DN_p = 32$. The state-of-the-art PA method is the stochastic sequential search (SSS) algorithm proposed in [27]. It is an iterative method consisting of two loops, i.e., the outer loop (run for T_{out} iterations) and the inner loop (run for T_{in} iterations). At the start of the nth outer loop iteration, a novel random PA, denoted as $\mathcal{P}^{(n)}$, is generated. This initial PA is then further optimized by running T_{in} inner loop iterations. During each inner loop iteration, every element $p_i^{(n)}$ of $\mathcal{P}^{(n)}$ is sequentially updated, keeping the other elements of $\mathcal{P}^{(n)}$ (i.e., the elements in $\mathcal{P}_{\sim i}^{(n)} = \mathcal{P}^{(n)} \backslash \left\{ p_i^{(n)} \right\}$) fixed. Updating $p_i^{(n)}$, consists in replacing the current value of $p_i^{(n)}$ with the value in $\{1, 2, \ldots, N\} \backslash \mathcal{P}_{\sim i}^{(n)}$ yielding the lowest MMMC. The PA $\mathcal{P}^{(n)}$ that is obtained after completing T_{in} inner loop iterations, as well as the corresponding MMMC $\boldsymbol{\Phi}^{(n)}$, are stored for further processing. The final outcome of SSS is the PA with the lowest MMMC among all stored PAs $\{\mathcal{P}^{(1)}, \mathcal{P}^{(2)}, \ldots, \mathcal{P}^{(T_{out})}\}$. Since SSS is essentially a random search, it will theoretically always converge to the optimal MMMC (for T_{out} sufficiently large). However, the main disadvantage of SSS is its complexity. Measuring the computational complexity in terms of the number of elementary operations on real quantities (NOR)[1] involved, the computational complexity of SSS is easily calculated as:

$$\text{NOR}_{\text{SSS}} = T_{out} T_{in} 2 (DN_p)^2 (N - DN_p)(L-1). \tag{14}$$

The value of T_{out} and T_{in} can be used to leverage between performance and complexity. For N much larger than DN_p and for L large as compared to 1, the following approximation applies:

$$\text{NOR}_{\text{SSS}} \approx 2 T_{out} T_{in} N (DN_p)^2 L. \tag{15}$$

PA algorithms are often run offline: an appropriate PA for every possible combination of N, L and DN_p is pre-computed and the resulting PA dictionary is stored. In such a scenario, one can argue that the complexity of the employed PA algorithm is not so relevant. Nevertheless, since it is such an impossible job to find the true optimal PA (this would require an exhaustive search), designers keep continuously looking for novel ways to compute better PAs in less time. This will be particularly true when the number of required dictionary entries is large. By means of example, we consider an OFDM system with $N = 1024$ subcarriers and a cyclic prefix length $L_{cp} = \frac{N}{4}$. In this case, L can take all (integer) values from 1 to 256 and DN_p can take all (integer) values from 1 to $L - 1$. (When $DN_p > L$, accurate channel estimation can be obtained with conventional methods and this case is not considered in our paper.) This means that the number of entries in the dictionary gets as large as

$$N_{dic} = \sum_{l=1}^{256} (l-1) = 32640. \tag{16}$$

As the example shows, dictionaries with several thousands of entries are not an exception. It should also be noted that devices with limited memory may have not enough space to store a PA dictionary of that size. In such a scenario, online PA computation is the only option and the use of low complexity PA algorithms becomes even more important.

In the following, we propose three efficient PA methods with a lower complexity than SSS, i.e., the deterministic greedy algorithm (DGA), the simplified deterministic greedy algorithm (SDGA) and a simulated annealing (SA) based method. The respective performance of the proposed methods will be contrasted to that of the state-of-the-art in Section 4. Two of the proposed methods, namely DGA and SA, outperform the state-of-the-art SSS in terms of convergence speed by several orders of magnitude. This has important consequences: as our simulations demonstrate, it is not possible for SSS to achieve the same accuracy as SA and DGA within a reasonable amount of computation time. The third PA method that

[1] The elementary operations on real quantities include addition, subtraction, multiplication and division of real numbers.

we propose, SDGA, focuses entirely on complexity; it is therefore of interest only for scenarios with online PA.

3.1. DGA

The first algorithm that we propose is a deterministic greedy algorithm (DGA) that sequentially finds the pilot positions p_t, $t = 1, 2, \ldots, DN_p$. DGA takes p_1 equal to 1. Subsequently, the values of p_t for $t = 2, 3, \ldots$. are obtained, using the following recursive equation:

$$p_t = \underset{p_t \in \mathcal{N} \backslash \mathcal{P}_{t-1}}{\operatorname{argmin}} \left(\max_{1 \le q \le L-1} \frac{1}{t} \alpha_q(t) \right), \tag{17}$$

where $\mathcal{P}_t = \{p_1, p_2, \ldots, p_t\}$ collects the first t pilot positions, $\mathcal{N} \backslash \mathcal{P}_{t-1}$ is the set of all subcarrier indices in $\{1, 2, \ldots N\}$ but not in $\mathcal{P}_{t-1}$ and $\alpha_q(t)$ is defined as:

$$\alpha_q(t) = \left| \sum_{i=1}^{t} e^{-j2\pi \frac{(p_i - 1)q}{N}} \right|. \tag{18}$$

The procedure is summarized in Algorithm 1.

Algorithm 1 DGA and SDGA.

```
Set $p_1 = 1$ and $\mathcal{P}_1 = \{p_1\}$.
for $t = 2, 3, \ldots, DN_p$ do
    Find $p_t$ from (17) for DGA, or from (26) for SDGA
    Construct $\mathcal{P}_t$ as the union of $\mathcal{P}_{t-1}$and $p_t$
end for
Output $\mathcal{P} = \mathcal{P}_{DN_p}$.
```

The NOR involved in the DGA is:

$$\begin{aligned} \mathrm{NOR}_{\mathrm{DGA}} &= \sum_{t=2}^{DN_p} (N - t + 1) 2t(L-1), \\ &= \frac{DN_p(DN_p + 1)(3N - DN_p + 1) - 6N}{3}(L-1), \end{aligned} \tag{19}$$

which, for N much larger than DN_p and for L large as compared to 1, can be approximated as:

$$\mathrm{NOR}_{\mathrm{DGA}} \approx N(DN_p)^2 L. \tag{20}$$

Comparing (20) with (15), it follows that DGA is about half as complex as SSS with $T_{out} = T_{in} = 1$. This can be attributed to the absence of iterations and to the fact that the number of positions to be tested decreases on every recursion.

For completeness, we mention that a greedy approach similar to DGA was proposed in [26]; however, instead of minimizing the MMMC directly, the algorithm in [26] minimizes the variance of pilot location difference. The computational complexity of the algorithm in [26] is of order $\mathcal{O}(N!)$. Hence, DGA has a far lower complexity. Furthermore, numerical results (not reported in this paper) show that DGA in general yields a lower MMMC than the method from Pakrooh et al. [26], which in its turn results in a lower channel estimation MSE and system BER.[2]

3.2. SDGA

Although DGA has a significant lower complexity than SSS and than the method from Pakrooh et al. [26], the computational complexity still increases proportionally with N and L. To derive a lower complexity algorithm, we first rewrite (11) as:

$$\mu\{\mathbf{\Phi}\} = \max_q \frac{1}{DN_p} \alpha_q(DN_p). \tag{21}$$

with $\alpha_q(\cdot)$ defined as in (18). The MMMC $\mu\{\mathbf{\Phi}\}$ is the maximum among $\alpha_1(DN_p), \ldots, \alpha_{L-1}(DN_p)$ and therefore $\alpha_q(DN_p)$ should be low for all q in $\{1, \ldots, L-1\}$ in order to have a low MMMC. Keeping this in mind, we propose to solve the simpler problem of finding a PA yielding a small $\alpha_q(DN_p)$ for a given value of q. Adopting a similar greedy approach as before, such a PA can be found using $p_1 = 1$ and the recursive relationship

$$p_t = \underset{p_t \in \mathcal{N} \backslash \mathcal{P}_{t-1}}{\operatorname{argmin}} \left| \alpha_q(t-1) + e^{-j2\pi \frac{(p_t - 1)q}{N}} \right|, \tag{22}$$

which allows to compute p_t, for $t = 2, 3, \ldots, DN_p$, from $\{p_1, p_2, \ldots, p_{t-1}\}$, using the definition of $\alpha_q(t-1)$ from (18). For given $\alpha_q(t-1)$, (22) is equivalent to

$$p_t = \underset{p_t \in \mathcal{N} \backslash \mathcal{P}_{t-1}}{\operatorname{argmin}} f_q(p_t; \mathcal{P}_{t-1}), \tag{23}$$

with

$$f_q(p_t; \mathcal{P}_{t-1}) = \left| \left| \arg(\alpha_q(t-1)) + 2\pi \frac{(p_t - 1)q}{N} \right|_{2\pi} - \pi \right|, \tag{24}$$

where $|x|_{2\pi}$ denotes the value of x modulo-2π with values in [0, 2π] and arg(x) is the argument of x with values in $[-\pi, \pi]$. The function $f_q(\lambda; \mathcal{P}_{t-1})$ is a saw-tooth function of λ with period N/q, so that the interval $[\frac{1}{2}, N + \frac{1}{2}]$ corresponds to q entire periods of $f_q(\lambda; \mathcal{P}_{t-1})$. The complexity of (23) can be significantly reduced by noting that the search can be limited to the $2q$ values of p_t in $\mathcal{N} \backslash \mathcal{P}_{t-1}$ that are closest to the q zero-crossings $(\lambda_1, \lambda_2, \ldots, \lambda_q)$ of $f_q(\lambda; \mathcal{P}_{t-1})$ from (24) in $[\frac{1}{2}, N + \frac{1}{2}]$. The lowest complexity is obtained with $q = 1$, in which case only 2 values of p_t need to be tested to find the solution to (23) and we obtain

$$p_t = \underset{\{p^*_{t,l}, p^*_{t,s}\}}{\operatorname{argmin}} f_1(p_t; \mathcal{P}_{t-1}), \tag{25}$$

where $p^*_{t,s}$ is the largest element in $\mathcal{N} \backslash \mathcal{P}_{t-1}$ smaller than or equal to λ_1, and $p^*_{t,l}$ is the smallest element in $\mathcal{N} \backslash \mathcal{P}_{t-1}$ larger than or equal to λ_1, with λ_1 the unique solution to $f_1(\lambda, \mathcal{P}_{t-1}) = 0$.

As our goal is to derive a low complexity PA algorithm, a simple ad-hoc approach could be to use $p_1 = 1$ and (25) to derive $\mathcal{P}_{DN_p}$. Unfortunately, it turns out that this approach often results in a rather large MMMC. This does not come as a complete surprise. In many cases, a PA that yields a very small $\alpha_1(DN_p)$ yields a very large $\mu\{\mathbf{\Phi}\}$. An obvious example of such a PA is an equidistant PA in which the pilot subcarriers are evenly spread over the entire bandwidth. To allow (25) to escape from this undesired PA without jeopardizing the minimization of $\alpha_1(DN_p)$ too much, we propose to replace (25) with

$$p_t = \underset{\{p'_{t,l}, p'_{t,s}\}}{\operatorname{argmin}} f_1(p_t; \mathcal{P}_{t-1}), \tag{26}$$

where $p'_{t,s}$ is the largest element in $\mathcal{N} \backslash \mathcal{P}_{t-1}$ smaller than or equal to $\lambda_1 + 1$, and $p'_{t,l}$ is the smallest element in $\mathcal{N} \backslash \mathcal{P}_{t-1}$ larger than $\lambda_1 - 1$. The PA algorithm that uses $p_1 = 1$ and (26) is further referred to as simplified deterministic greedy algorithm (SDGA). The procedure of SDGA is similar to that of DGA, and is also summarized by Algorithm 1. The computational complexity of SDGA is of the order $\mathcal{O}(DN_p)$ only.

3.3. SA

Because DGA and SDGA are deterministic greedy algorithms, they guarantee to produce a result in a limited amount of steps. However, as they never reconsider their choices, they mostly fail to find an approximation of the globally optimal solution. To enable the search algorithm to escape from a local optimum while still limiting the computational effort, simulated annealing can be adopted [39]. Algorithm 2 details how SA can be applied to solve

[2] For example, for $N = 256$, $L = 64$ and $DN_p = 32$, the MMMC of the method from Pakrooh et al. [26] is 0.2402, as opposed to 0.1896 for DGA.

Algorithm 2 SA.

```
1:  Set $T_{init}$, $T_{stop}$, $T_{rate}$ and $T_{iter}$.
2:  Select an initial PA $\mathcal{P}$.
3:  Set $T = T_{init}$.
4:  while $T > T_{stop}$ do%outerloop
5:      for $l = 1, 2, \ldots, T_{iter}$ do %innerloop
6:          Randomly pick a pilot index $k$ from $\{1, 2, \ldots, DN_p\}$.
7:          Exchange $p_k$ with a randomly selected value from $\mathcal{N}\backslash\mathcal{P}$to form $\mathcal{P}_{tmp}$.
8:          Calculate $\mu\{\boldsymbol{\Phi}\}_{tmp}$ correspondingto $\mathcal{P}_{tmp}$
9:          if $\mu\{\boldsymbol{\Phi}\}_{tmp} - \mu\{\boldsymbol{\Phi}\} < 0$ or $e^{-\frac{\mu\{\boldsymbol{\Phi}\}_{tmp}-\mu\{\boldsymbol{\Phi}\}}{T}} > rand(\cdot)$ then
10:             Set $\mathcal{P} = \mathcal{P}_{tmp}$.
11:         end if
12:     end for
13:     $T = T \cdot T_{rate}$.
14: end while
15: Output $\mathcal{P}$.
```

(13). Basically, we randomly alter the value of a randomly picked pilot position p_k, and compute the MMMC $\mu\{\boldsymbol{\Phi}\}_{tmp}$ for the new PA. Theoretically, it is possible that the new PA has already been tested in one of the previous steps. However, the probability that this happens is low. For example, if the total number of subcarriers N and the total number of pilot subcarriers DN_p equal 256 and 32, respectively, the probability of repeated testing is in order of $\frac{1}{DN_p(N-DN_p)} = 1.4 \cdot 10^{-4}$. If $\mu\{\boldsymbol{\Phi}\}_{tmp}$ of the new PA is better than the MMMC $\mu\{\boldsymbol{\Phi}\}$, we update the PA. However, as we are changing only one pilot symbol at a same time, using this condition is equivalent to searching for a local optimum in the neighborhood of the initial PA, implying we risk to get trapped in a local optimum. To solve this issue, SA allows a new PA to be selected, even if the MMMC is not better than the old PA. This is done by adding the condition

$$\exp\left(-\frac{\mu\{\boldsymbol{\Phi}\}_{tmp} - \mu\{\boldsymbol{\Phi}\}}{T}\right) > rand(\cdot), \tag{27}$$

i.e., the new PA is selected, even when $\mu\{\boldsymbol{\Phi}\}_{tmp} - \mu\{\boldsymbol{\Phi}\} > 0$, if the exponential of (27) is larger than a random value generated with a continuous uniform distribution from 0 to 1, i.e., $rand(\cdot) \sim U[0, 1]$. The exponential in (27) depends on the parameter T, which is called the temperature of the SA algorithm. If T is large, the probability that the condition (27) is fulfilled is large, implying that the algorithm can easily escape a local optimum. However, for large T, this easiness to escape an optimum also hinders the convergence if we are close to the global optimum, as the probability that the algorithm jumps away from the global optimum is large. Therefore, we start with a high initial temperature, T_{init}, and gradually reduce the temperature by scaling the temperature with a factor $T_{rate} < 1$. The algorithm ends when the temperature reaches a (low) stop temperature T_{stop} (outer loop). For each value of T, the algorithm tries to converge to an optimum by randomly selecting T_{iter} new pilot symbol positions p_k, and checking one by one if at least one of the conditions in line 9 of the algorithm is satisfied (inner loop). The set of rules presented in [40] provide useful guidelines to select the parameters T_{init}, T_{stop}, T_{rate} and T_{iter}.

The computational complexity of SA is dominated by the MMMC calculations involved in Algorithm 2. The evaluation of a single MMMC has complexity $2DN_p(L-1)$, and the number of MMMC calculations is $T_{iter}\left\lceil \log_{T_{rate}}\left(\frac{T_{stop}}{T_{init}}\right)\right\rceil$. This results in a total complexity number of

$$\text{NOR}_{\text{SA}} = \beta_N 2DN_p(L-1) \approx \beta_N 2DN_p L, \tag{28}$$

with $\beta_N = T_{iter}\left\lceil \log_{T_{rate}}\left(\frac{T_{stop}}{T_{init}}\right)\right\rceil$.

4. Numerical results and discussion

In this section, we demonstrate the performance of the proposed DGA, SDGA and SA methods for PA design in terms of NOR, MMMC, MSE and BER. Unless specified differently, SDGA is employed to select the initial PA in the case of SA. Alternatively, random (RAND) or equidistant PA initialization can be used. Following SSS, we also consider the option to extend the SA algorithm with an outer loop, running for T_{out} iterations. Then, in each iteration t_{out} = 1, 2,..., T_{out}, a new initial PA (further represented as $\text{PA}_{t_{out}-1,0}$) is selected and further optimized according to the SA procedure from Algorithm 2, after which the MMMC of all T_{out} obtained PAs is compared and only the PA with the lowest MMMC is retained. Compressive sampling matching pursuit (CoSaMP) [15] was chosen as the reconstruction algorithm. CoSaMP is a variant of the well-known orthogonal matching pursuit algorithm [15] and is known for its good performance and low complexity. Preliminary results, not presented here, indicate that very similar results are obtained using other CS reconstruction methods [17–20]. As the MMMC depends on the product DN_p but not on D and N_p separately, we will further focus on the single OFDM symbol case only; hence, $D = 1$ and the number of elements in $\mathcal{P}$ is N_p in our simulations. Preliminary results, not reported in this paper, confirm that a system with $D = x$ and $N_p = y$ behaves virtually the same as a system with $D = 1$ and $N_p = xy$, if the channel remains constant for at least D OFDM symbols. This means that, assuming that the required number of pilot subcarriers is known (mainly determined by the sparsity of the channel), increasing D can improve the bandwidth and power efficiency of the transmission. The number of subcarriers, N, is a power of 2. We assume that $L = \frac{N}{4}$ [41,42]. Further, $N_p = \frac{N}{8}$ is considered as a typical number of pilot symbols per OFDM symbol [41]. To estimate the BER and the MSE, the expectation in (3) and (5) is replaced by an arithmetic mean over a large amount of random realizations of $\boldsymbol{b}$, $\boldsymbol{h}$ and $\boldsymbol{W}$. Hereby, all possible sequences $\boldsymbol{b}$ are picked with equal probability $2^{-m(N-N_p)}$. Quadrature phase shift keying (QPSK) and Gray code bit-to-symbol mapping is applied. Furthermore, each possible way to select K out of L channel tap indices is picked with equal probability $(K!(L-K)!)/L!$. The values of the non-zero channel taps are generated independently according to $\mathcal{CN}\left(0, \frac{1}{K}\right)$. Finally, the noise samples vector $\boldsymbol{W}$ is generated according to $\mathcal{CN}\left(0, \sigma^2\boldsymbol{I}_N\right)$. Results are presented as a function of the average signal-to-noise ratio (SNR), i.e., $\text{SNR} = \mathbb{E}[\frac{|X_d(l)|^2|H(l)|^2}{\sigma^2}] = \frac{E_s}{\sigma^2}$.

4.1. Evaluation of SDGA

We first discuss the performance of SDGA. Considering three cases N_p = 26, 28 and 32 for N = 256 and L = 64, the achieved MMMCs of SDGA are 0.3476, 0.3575 and 0.3337, respectively. Note that randomly selecting a PA might also result in a satisfactory MMMC. To demonstrate the effectiveness of SDGA, we determine the average amount of PAs that needs to be randomly generated before finding one that has a lower MMMC than SDGA. To this end, we randomly generate 10^6 PAs for each of the abovementioned cases and calculate their corresponding MMMC. To simplify the analysis, we split these MMMCs into $10^6/N_g$ groups of size N_g, with N_g = 1, 2, 5, 10, 20, 25, 40 or 50. Finally, we count in how many groups the minimum MMMC is lower than the MMMC obtained with SDGA, and calculate the probability $\Pr[\text{MMMC}_{\text{SDGA}} < \min_{i=1,2,\ldots,N_g}\{\text{MMMC}_{\text{rand}}^{(i)}\}]$ that the MMMC of SDGA is smaller than the lowest of each group. The value of this probability is shown in Fig. 2 as a function of N_g. For $N_g = 1$, in over 67% of the cases (over 87% for $N_p = 26$), SDGA yields a lower MMMC than a randomly generated PA. Further, it can be observed that in order to outperform SDGA with a certainty of 99%, one should randomly

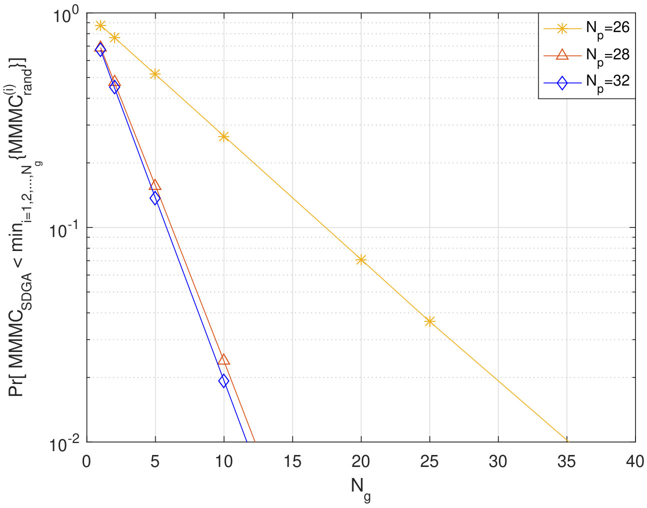


Fig. 2. Probability that SDGA yields a lower MMMC than the best out of a group of N_g randomly generated PAs.

Table 1
List of SA inner loop parameter settings.

Sett_{in}	T_{init}	T_{stop}	T_{iter}	T_{rate}
1	10^{-1}	10^{-6}	50	0.95
2	10^{-2}	10^{-6}	50	0.95
3	1	10^{-6}	50	0.95
4	10^{-1}	10^{-6}	50	0.90
5	10^{-1}	10^{-6}	50	0.97
6	10^{-1}	10^{-6}	20	0.95
7	10^{-1}	10^{-6}	70	0.95
8	1	10^{-7}	70	0.97

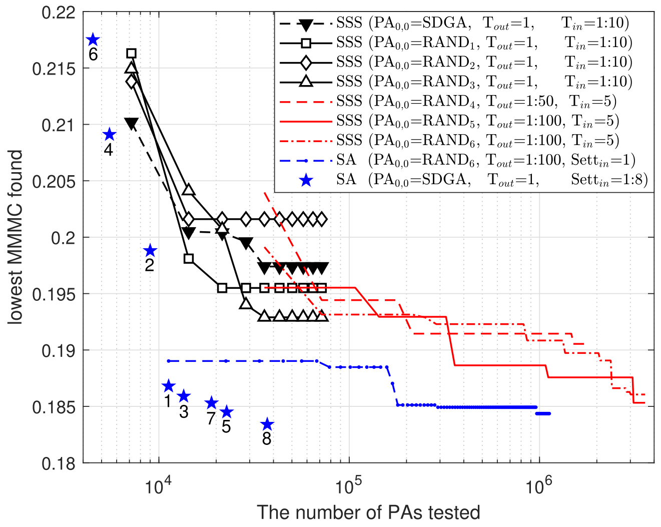


Fig. 3. Lowest MMMC found versus the number of PAs tested, for N=256, $N_p = 32$ and $L = 64$, using SSS and SA with different parameter settings; $\mathrm{PA}_{0,0}$ denotes the type of PA that is used to bootstrap the algorithms, if random initialization is employed, a different subindex i in RAND_i indicates that the random generator was initialized with a different seed; T_{out} is the number of outer SSS or SA iterations; T_{in} is the number of inner SSS iterations; and Sett_{in} is the index of the SA inner loop parameter settings from Table 1. The different curves result from varying the value of T_{in}, T_{out} or Sett_{in}, while keeping the value of the other parameters fixed.

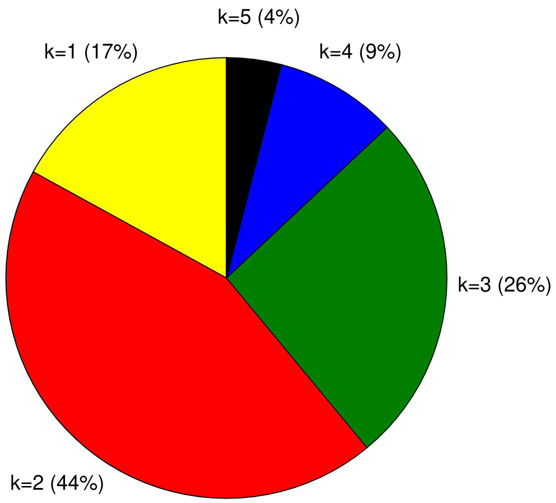


Fig. 4. Average number k of SSS inner loop iterations required to achieve convergence, for N=256, $N_p = 32$ and $L = 64$.

generate 12 (for $N_p = 32, 28$) or even 35 (for $N_p = 26$) PAs, evaluate all corresponding MMMCs and determine the PA with minimum MMMC. However, this results in a complexity of $\mathcal{O}(24N_p(L-1))$ or $\mathcal{O}(70N_p(L-1))$, which is far larger than that of SDGA. Owing to the acceptable performance and the extremely low complexity, SDGA can be considered as an effective method for situations with online PA computation where the PA needs to be redesigned frequently due to fast channel variations.

4.2. SSS and SA parameter selection

For a meaningful comparison, suitable values for T_{init}, T_{stop}, T_{rate}, T_{iter} and T_{out} for SA, and for T_{in} and T_{out} for SSS, need to be selected for each system setting. A short-list of 8 interesting SA inner loop parameter combinations (T_{init}, T_{stop}, T_{rate}, T_{iter}) is made by taking into account [40]; these settings, labeled with an integer Sett_{in} ranging from 1 to 8, are tabulated in Table 1. Fig. 3 shows the MMMC of PAs obtained for $N = 256$, $N_p = 32$ and $L = 64$, using SSS or SA with different types of initialization, different T_{out}, different T_{in} and different Sett_{in}. We make the following observations:

- The black curves show the evolution of the MMMC as a function of the number of inner SSS iterations. Independent of the initial PA, the MMMC barely decreases after the 5th inner loop iteration. This is confirmed by Fig. 4, showing (for a total of 100 experiments) the percentage of experiments where the MMMC does not change between the kth and the 10th inner loop iteration. We conclude that, for the considered set-up, the number of SSS inner loop iterations T_{in} can be safely limited to 5, without loosing performance.
- The red curves show the evolution of the MMMC as a function of the number of outer SSS iterations. The blue curve with circular markers does the same for SA with $\mathrm{Sett}_{in} = 1$. The blue star-shaped markers show the effect of the SA inner loop parameter values for T_{out} equal to 1. Both SSS and SA converge towards the PA with the smallest MMMC, but SA decreases the MMMC significantly faster than SSS. For example, the number of PAs that needs to be tested to obtain an MMMC of less than 0.19 is 30 to 200 times larger with SSS than with SA. Moreover, while SA achieves an MMMC lower than 0.184 after having tested only 37100 PAs, the lowest MMMC achieved with SSS after testing over $3 \cdot 10^6$ PAs is still larger than 0.185. In order to compare algorithms with a similar complexity, it is reasonable to take T_{out} equal to 1, for SSS and SA both. As for SA, $\mathrm{Sett}_{in} = 1$ can be considered to offer a close-to-optimum NOR-MMMC trade-off for the considered value set-up.

We also did simulations for other values of N and found similar results. Although both SSS and SA have the potential to find

Table 2
Parameter values for SSS and SA.

N	SSS	SA				
	T_{in}	T_{init}	T_{stop}	T_{iter}	T_{rate}	β_N
64	2	10^{-1}	10^{-6}	50	0.90	5500
128	3	10^{-1}	10^{-6}	50	0.95	11250
256	5	10^{-1}	10^{-6}	50	0.95	11250
512	5	1	10^{-6}	50	0.95	13500
1024	6	1	10^{-6}	50	0.97	22700

Table 3
NOR for SSS, DGA and SA when $L = \frac{N}{4}$ and $N_p = \frac{N}{8}$.

N	NOR		
	SSS	DGA	SA
64	$2.1504*10^5$	$6.4680*10^4$	$1.3200*10^6$
128	$5.3330*10^6$	$1.0292*10^6$	$1.1400*10^7$
256	$1.4451*10^8$	$1.6311*10^7$	$4.5360*10^7$
512	$2.3304*10^9$	$2.5928*10^8$	$2.1946*10^8$
1024	$4.4920*10^{10}$	$4.1328*10^9$	$1.4819*10^9$

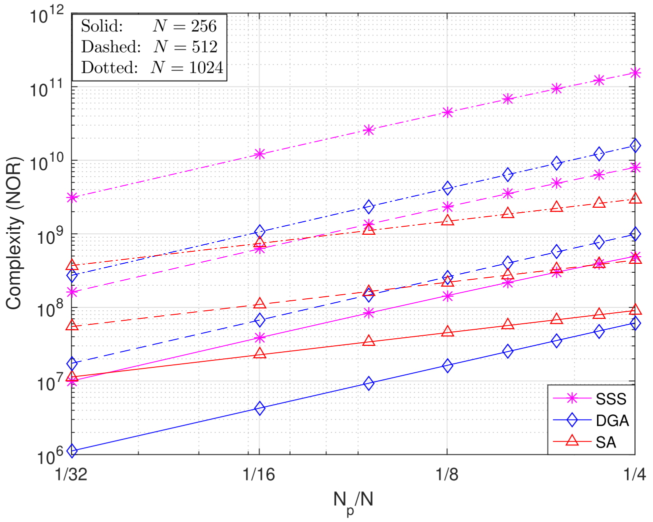


Fig. 5. Exact NOR of SSS, DGA and SA for $L = N/4$.

Table 4
MMMC for SSS, SDGA, DGA and SA when $L = \frac{N}{4}$ and $N_p = \frac{N}{8}$.

N	MMMC			
	SSS	DGA	SDGA	SA
64	0.3062	0.3062	0.7304	0.3062
128	0.2791	0.2782	0.7716	0.2576
256	0.1974	0.1896	0.3337	0.1868
512	0.1606	0.1547	0.2916	0.1498
1024	0.1291	0.0976	0.2290	0.0895

the global optimum, implying they eventually will converge to the same MMMC for a large number of outer iterations, our simulation results indicate that the SA algorithm has a better convergence speed than SSS. As a consequence, a pilot allocation yielding a given MMMC that is obtained with SA within a few hours or days, would take weeks or months to compute using the SSS algorithm. This situation will occur when $N > 256$ and/or $T_{out} > 1$. It is clear that this is not practical, even though the pilot allocation can be computed offline. In Table 2, we enlist suitable values of T_{init}, T_{stop}, T_{rate}, T_{iter} for SA, and T_{in} for SSS, for N=64, 128, 512, 1024 (with $N_p = \frac{N}{8}$ and $L = \frac{N}{4}$). The number of outer iterations T_{out} is fixed to 1 throughout the rest of the paper.

4.3. NOR

We now compare the computational complexity of DGA, SDGA, SA and SSS for different values of N and N_p/N. The value of β_N in (28) obtained with the proposed values of T_{iter}, T_{stop}, T_{rate} and T_{init} is displayed in the seventh column of Table 2.

It is immediately clear from (15), (20) and (28) that, as compared to DGA, SA and SSS, SDGA has negligible complexity (only $\mathcal{O}(N_p)$). Table 3 shows numerical values for the NOR of DGA, SA and SSS, for a fixed ratio $\frac{N_p}{N} = \frac{1}{8}$. We observe that for large N ($N = 512$ and $N = 1024$), SA yields a lower NOR than DGA, which in turn yields a lower NOR than SSS. This is in correspondence with β_N increasing less than proportional to $\frac{NN_p}{2}$ (see (15), (20) and (28)). For smaller values of N ($N = 256$, 128, 64), DGA turns out to be less complex than SA. This is because, for small N, the prefactor β_N is large as compared to $\frac{NN_p}{2}$. For the considered values of N (≥ 256), SSS yields the highest NOR.

The impact of the pilot subcarrier ratio N_p/N on the computational complexity of the considered PA methods is shown in Fig. 5 where, on a log-log diagram, NOR_{DGA}, NOR_{SA} and NOR_{SSS} are plotted as a function of $\frac{N_p}{N} \in [\frac{1}{32}, \frac{1}{4}]$, for $N = 256$, $N = 512$ and $N = 1024$. The curves for SSS and SA were obtained with the values of the parameters fixed to the values proposed in Table 2 for $N = 256, 512, 1024$ and $N_p = N/8$. So, for given N, the parameters β_N and T_{in} do not vary with N_p/N. We make the following observations:

- For given N (and $L = N/4$), the NOR of SA increases proportionally to N_p, whereas the NOR of DGA and SSS increases proportionally to N_p^2. For given N and given N_p/N, DGA is $2T_{in} \approx 10$ times less complex than SSS. For given N and small N_p/N, DGA is also less complex than SA. For given N and very small N_p/N, even SSS becomes less complex than SA. However, for given N and large N_p/N, both SSS and DGA are more complex than SA.
- If N increases, the value of N_p/N above which SA becomes the least complex method decreases. The NOR values at $N_p/N = 1/8$, correspond to the ones reported in Table 3. For values of N_p/N in the vicinity of this typical value, DGA is the least complex method for $N = 256$ (and below) and SA is the least complex method for $N = 1024$ (and above), while SA and DGA have about the same complexity (but significantly lower than SSS) for $N = 512$.

4.4. MMMC

In Section 4.3, it was argued that all three proposed algorithms are significantly less complex than the state-of-the-art method. Interestingly, Table 4 and Fig. 6 show that the complexity reduction offered by the proposed algorithms, not necessarily comes at the expense of an increased MMMC. Table 4 contains MMMC results for several values of N and N_p fixed to $N/8$. Fig. 6 shows MMMC values for fixed N ($N = 256$) as a function of N_p, for N_p values in the vicinity of $N/8$.

The very simple SDGA method results in much higher MMMC values than SSS. However, as the obtained MMMC decreases with N, SDGA might still be considered an interesting option for systems with a large amount of subcarriers and very stringent complexity constraints. On the other hand, DGA and SA both yield a lower MMMC than SSS in all considered cases. In all scenarios, the lowest MMMC is obtained with SA, even though this method has a lower complexity than SSS and, for high N, also a lower complexity than DGA.

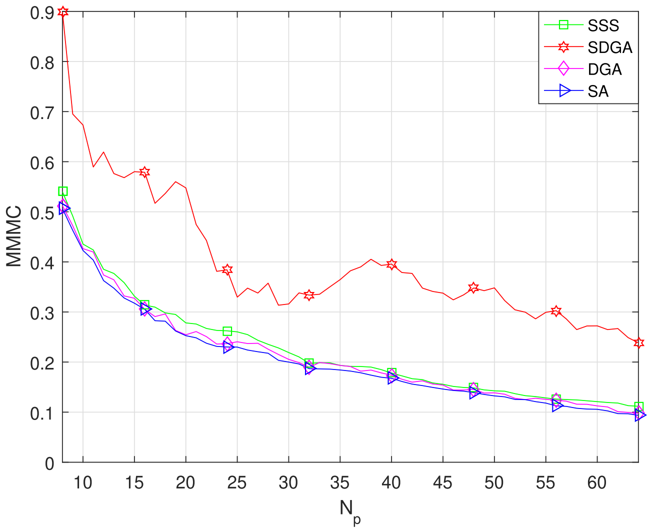


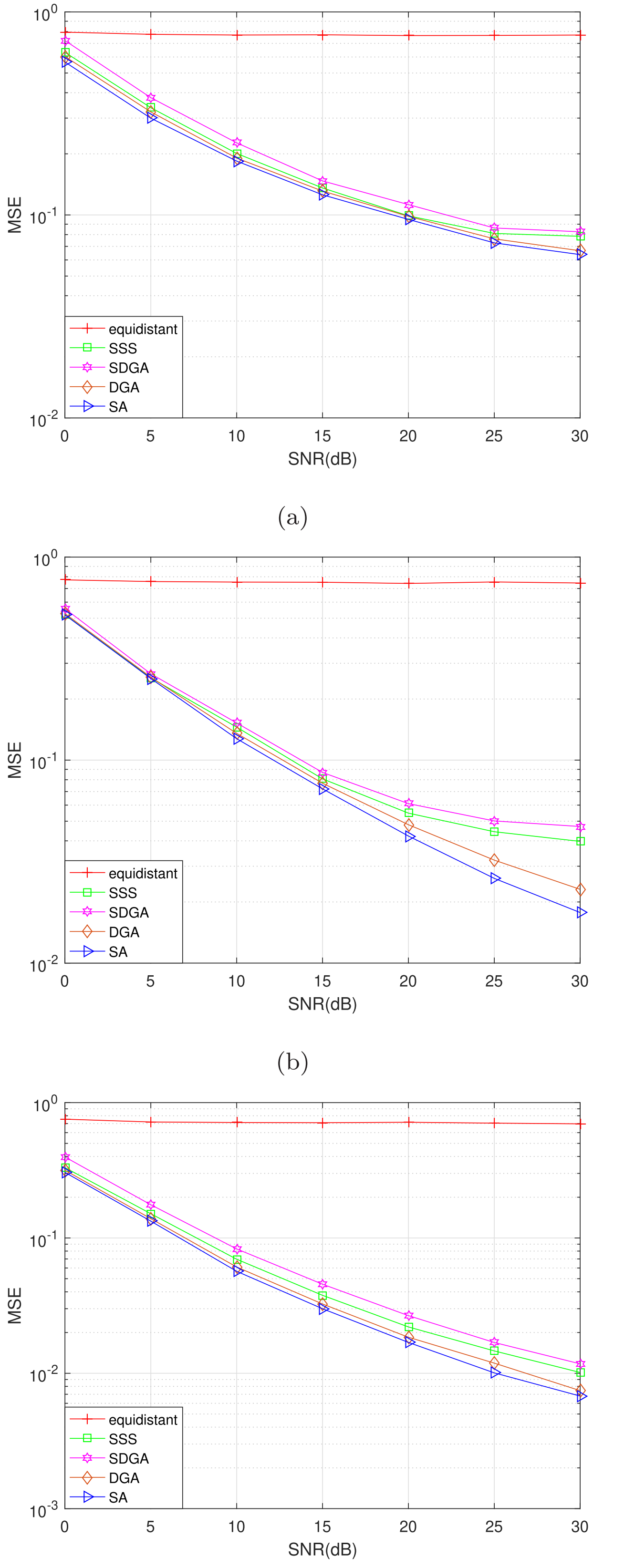
Fig. 6. MMMC values as a function of N_p for SSS, DGA, SDGA and SA.

4.5. MSE

The whole idea behind designing the PA with a view to minimizing the MMMC relies on the assumed relationship between the MMMC and the MSE performance of the considered channel estimator. Fig. 7 shows the obtained MSE as a function of the SNR for a system with $N = 256$, $N_p \in \left\{26, 28, 32 = \frac{N}{8}\right\}$, $L = \frac{N}{4} = 64$ and $K = 12$, if that system uses SSS, DGA, SDGA or SA PA design. At given SNR and for given N_p, SDGA yields a higher MSE than SSS, SSS yields a higher MSE than DGA and SA yields the lowest MSE. This order of improving MSE performance is in correspondence with the observation in Fig. 6 that, for the considered values of N_p, SDGA yields a higher MMMC than SSS, which yields a higher MMMC than DGA, which in turn yields a higher MMMC than SA. At given SNR and for a given PA method, the obtained MSE decreases if N_p increases. For $N_p = 26$, the difference between the MSE values obtained with SSS, SDGA, DGA and SA is small. Increasing N_p from 26 to 28 results in a significantly larger MSE reduction for SA and DGA than for SSS and SDGA, especially at large SNR. This can be explained as follows. $N_p = 26$ pilot symbols are insufficient to produce a fairly accurate channel estimate independent of the PA. When N_p is increased to 28, a relatively good channel estimate can be obtained only with an appropriate PA (DGA and SA): for $N_p = 28$, SA results in a 3 times lower MSE than SSS at an SNR of 30 dB. For $N_p = 32$, the number of pilots is sufficiently large to also obtain an accurate channel estimate with less good PAs. For comparison, we also applied equidistant PAs for sparse channel estimation, i.e., $\mathcal{P}_{equi,26} = \{1, 11, \ldots, 251\}$, $\mathcal{P}_{equi,28} = \{5, 14, \ldots, 248\}$ and $\mathcal{P}_{equi,32} = \{1, 9, \ldots, 249\}$, which are the most common choice for conventional (non-sparse) channel estimation [43–45]. The corresponding MMMC values are $\mu\{\boldsymbol{\Phi}\}_{equi,26} = 0.9336$, $\mu\{\boldsymbol{\Phi}\}_{equi,28} = 0.9805$ and $\mu\{\boldsymbol{\Phi}\}_{equi,32} = 1$. For all three considered values of N_p, the MSE corresponding to the equidistant PA is at least an order of magnitude larger (at a nominal SNR of 15dB) than when the PA is designed using SSS, DGA, SDGA or SA, indicating that equidistant PAs are not suitable for sparse channel estimation, as was also observed in [27,30,33].

4.6. BER

From the above it follows that the proposed DGA and SA algorithms are attractive alternatives for the state-of-the-art approach

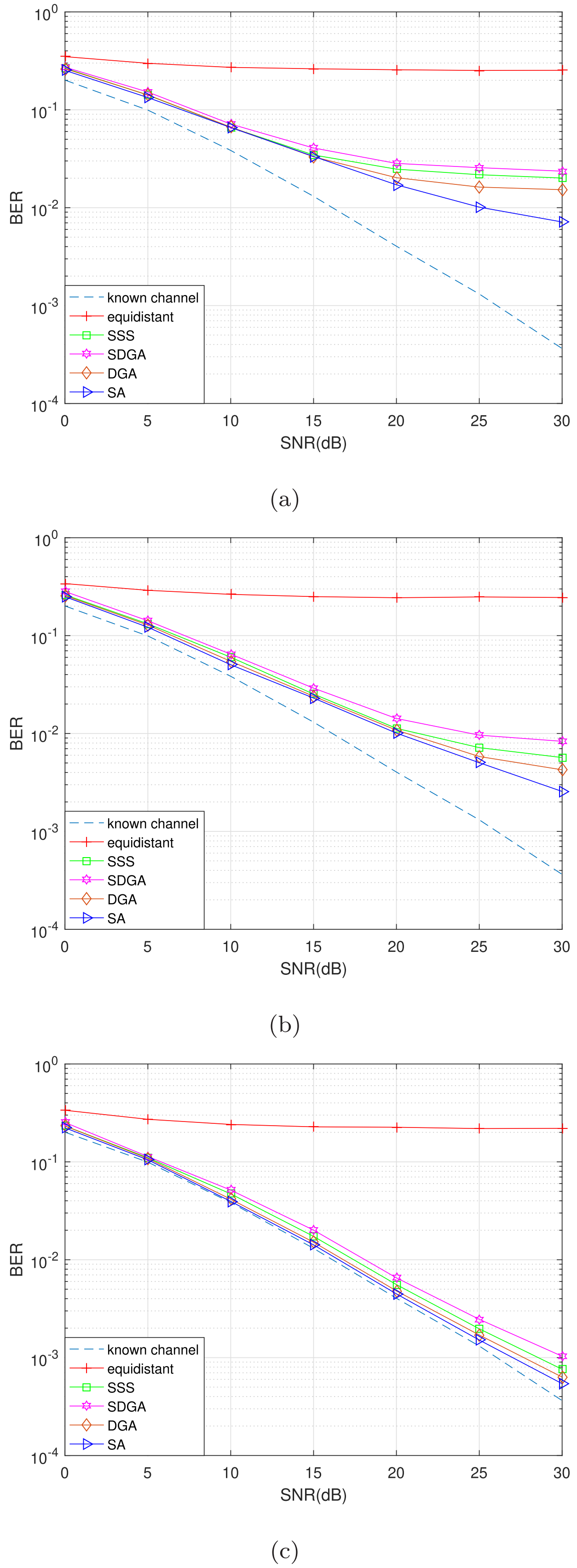


(a)

(b)

(c)

Fig. 7. MSE for a system with $N = 256$ and (a) $N_p = 26$, (b) $N_p = 28$, or (c) $N_p = 32$.



(a)

(b)

(c)

Fig. 8. BER for a system with $N = 256$ and (a) $N_p = 26$, (b) $N_p = 28$, or (c) $N_p = 32$.

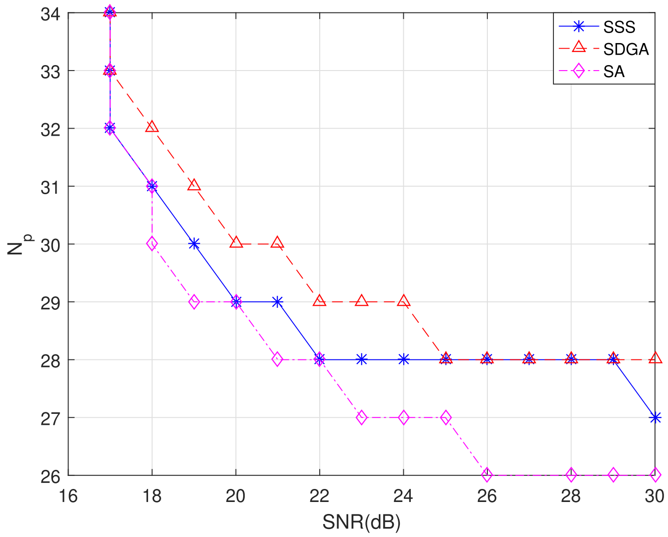


Fig. 9. Minimum N_p required to achieve and SNR for BER $= 10^{-2}$ at a given SNR.

(SSS) from Qi et al. [27]. For $N = 256$, $N_p \in \left\{26, 28, 32 = \frac{N}{8}\right\}$, $L = \frac{N}{4} = 64$ and $K = 12$, both SA and DGA yield a lower MSE and a lower NOR than SSS. While SA outperforms DGA in terms of MSE, the opposite is true with respect to the NOR. To conclude our analysis, in Fig. 8, we report the BER of the same system scenarios as previously considered in Fig. 7. Obviously, for given N_p and given PA, the BER decreases if the SNR increases. At given SNR and for a given PA method, the BER decreases if N_p increases. At a given SNR and for a given N_p, a PA method yielding a lower MMMC and MSE, also yields a lower BER. The lowest BER is achieved with SA. With this method, $N_p = 32 = \frac{N}{8}$ pilot symbols per OFDM symbol suffice to approach the BER of a system with perfect channel knowledge at the receiver. Using SA rather than SSS (or SDGA) may reduce the required amount of pilot subcarriers per OFDM symbol. This is illustrated more clearly in Fig. 9, which (for $N = 256$, $L = \frac{N}{4} = 64$ and $K = 12$) reports the minimum number of pilot subcarriers ($(N_p)_{\min}$) that is required to achieve a BER lower than 10^{-2} as a function of the SNR. A system with perfect channel knowledge at the receiver achieves a BER of 10^{-2} at an SNR equal to 17 dB. The same performance is achieved with SSS or SA and N_p larger than or equal to 32, and with SDGA and N_p larger than or equal to 33. For SNR larger than or equal to 17 dB, $(N_p)_{\min}$ decreases if the SNR increases. With less than $N_p = 26$ pilot subcarriers, it is virtually infeasible to reach a BER as low as 10^{-2}, independent of the SNR. This is because for such low values of N_p, all methods result in a BER floor above 10^{-2} (see Fig. 8(a)). We observe that using SA rather than SSS allows us to reduce the ratio N_p/N from almost 11% to about 10% for SNR in [26 dB, 29 dB]. Moreover, we notice that, for a given SNR, SDGA only needs about 2 (or 1) more pilot subcarriers in a total of 256 subcarriers to reach the same result as that of SA, meaning that the loss in pilot efficiency of SDGA is rather small. Hence, if there is no mandatory constraint on N_p, its extremely low complexity makes SDGA an attractive alternative to SA (because the same BER can be achieved with only a small increase of N_p/N).

5. Summary and conclusion

In this paper, we considered PA for pilot-assisted sparse channel estimation. To improve the performance of channel estimation, we propose three novel PA methods, i.e., DGA, SDGA and SA. All three methods use the MMMC as a cost function. Among these methods, SDGA has the lowest computational complexity, but its achieved MMMC is far from minimum. The computational complexity of DGA or SA is higher than that of SDGA, yet smaller than that of the state-of-the-art method SSS. On the other hand, the MMMC obtained with DGA or SA is far lower than that obtained with SDGA and also considerably smaller than that obtained with SSS. Numerical evaluation of corresponding MSE and BER results indicate that both DGA and SA outperform SSS, while SDGA comes with a small performance degradation but offers a huge complexity reduction.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Taoyong Li: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Visualization. **Nele Noels:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Visualization. **Heidi Steendam:** Resources, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgment

This work is supported by a Belgian EOS grant with project grant number EOS30452698, the Flemish fund for Scientific Research (FWO), the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme and the National Natural Science Foundation of China with grant number 61801516, 61701530 and 61701531.

References

[1] I. Oppermann, M. Hämäläinen, J. Iinatti, UWB: Theory and Applications, John Wiley & Sons, 2005.
[2] J. Reed, An Introduction to Ultra Wideband Communication Systems, Prentice Hall Press, 2005.
[3] M.-G. Di Benedetto, UWB Communication Systems: A Comprehensive Overview, vol. 5, Hindawi Publishing Corporation, 2006.
[4] R.J. Fontana, Recent system applications of short-pulse ultra-wideband UWB technology, IEEE Trans. Microw. Theory Tech. 52 (9) (2004) 2087–2104.
[5] F. Troesch, C. Steiner, T. Zasowski, T. Burger, A. Wittneben, Hardware aware optimization of an ultra low power UWB communication system, in: 2007 IEEE International Conference on Ultra-Wideband, IEEE, 2007, pp. 174–179.
[6] C. Park, T.S. Rappaport, Short-range wireless communications for next-generation networks: UWB, 60 GHz millimeter-wave WPAN, and ZigBee, IEEE Wirel. Commun. 14 (4) (2007) 70–78.
[7] A. Batra, Multi-band OFDM Physical Layer Proposal for IEEE 802.15 Task Group 3a, IEEE P802. 15-03/268r3 (2004).
[8] H.-K. Song, et al., Performance improvement of cooperative MB-OFDM system based coming home network, IEEE Trans. Consum. Electron. 53 (2) (2007) 442–447.
[9] E.J. Candès, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Inf. Theory 52 (2) (2006) 489–509.
[10] D.L. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (4) (2006) 1289–1306.
[11] J.L. Paredes, G.R. Arce, Z. Wang, Ultra-wideband compressed sensing: channel estimation, IEEE J. Sel. Top. Signal Process. 1 (3) (2007) 383–395.
[12] A.H. Muqaibel, M.T. Alkhodary, Practical application of compressive sensing to ultra-wideband channels, IET Commun. 6 (16) (2012) 2534–2542.
[13] X. Cheng, M. Wang, Y.L. Guan, Ultrawideband channel estimation: a bayesian compressive sensing strategy based on statistical sparsity, IEEE Trans. Veh. Technol. 64 (5) (2015) 1819–1832.
[14] S. Ji, Y. Xue, L. Carin, Bayesian compressive sensing, IEEE Trans. Signal Process. 56 (6) (2008) 2346–2356.
[15] D. Needell, J.A. Tropp, CoSaMP: iterative signal recovery from incomplete and inaccurate samples, Appl. Comput. Harmon. Anal. 26 (3) (2009) 301–321.
[16] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (1) (2009) 183–202.
[17] J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Trans. Inf. Theory 53 (12) (2007) 4655–4666.
[18] W. Dai, O. Milenkovic, Subspace pursuit for compressive sensing signal reconstruction, IEEE Trans. Inf. Theory 55 (5) (2009) 2230–2249.
[19] Y. Zhang, R. Venkatesan, O.A. Dobre, C. Li, An adaptive matching pursuit algorithm for sparse channel estimation, in: 2015 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2015, pp. 626–630.
[20] Y. Zhang, R. Venkatesan, C. Li, O.A. Dobre, Compressed sensing-based time–varying channel estimation in UWA-OFDM networks, in: 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE, 2015, pp. 1520–1525.
[21] W.U. Bajwa, J. Haupt, A.M. Sayeed, R. Nowak, Compressed channel sensing: a new approach to estimating sparse multipath channels, Proc. IEEE 98 (6) (2010) 1058–1076.
[22] J. Meng, W. Yin, Y. Li, N.T. Nguyen, Z. Han, Compressive sensing based high-resolution channel estimation for OFDM system, IEEE J. Sel. Top. Signal Process. 6 (1) (2011) 15–25.
[23] W. Ding, F. Yang, C. Pan, L. Dai, J. Song, Compressive sensing based channel estimation for OFDM systems under long delay channels, IEEE Trans. Broadcast. 60 (2) (2014) 313–321.
[24] D.L. Donoho, M. Elad, V.N. Temlyakov, Stable recovery of sparse overcomplete representations in the presence of noise, IEEE Trans. Inf. Theory 52 (1) (2006) 6–18.
[25] C. Qi, L. Wu, A study of deterministic pilot allocation for sparse channel estimation in OFDM systems, IEEE Commun. Lett. 16 (5) (2012) 742–744.
[26] P. Pakrooh, A. Amini, F. Marvasti, OFDM Pilot allocation for sparse channel estimation, EURASIP J. Adv. Signal Process. 2012 (1) (2012) 59.
[27] C. Qi, G. Yue, L. Wu, Y. Huang, A. Nallanathan, Pilot design schemes for sparse channel estimation in OFDM systems, IEEE Trans. Veh. Technol. 64 (4) (2015) 1493–1505.
[28] Y. Zhang, R. Venkatesan, O.A. Dobre, C. Li, Novel compressed sensing-based channel estimation algorithm and near-optimal pilot placement scheme, IEEE Trans. Wirel. Commun. 15 (4) (2015) 2590–2603.
[29] P. Xia, S. Zhou, G.B. Giannakis, Achieving the welch bound with difference sets, IEEE Trans. Inf. Theory 51 (5) (2005) 1900–1907.
[30] X. He, R. Song, W.-P. Zhu, Pilot allocation for distributed-compressed-sensing-based sparse channel estimation in MIMO-OFDM systems, IEEE Trans. Veh. Technol. 65 (5) (2016) 2990–3004.
[31] T. Ren, Y. Li, A novel joint pilot design scheme for sparse channel estimation in OFDM systems, Eng. Lett. 25 (1) (2017) 1–6.
[32] A.N. Uwaechia, N.M. Mahyuddin, New pilot allocation design schemes for sparse channel estimation in OFDM system, in: 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), IEEE, 2017, pp. 66–71.
[33] P. Vimala, G. Yamuna, Pilot design for sparse channel estimation in orthogonal frequency division multiplexing systems, J. Telecommun. Inf.Technol. (2018).
[34] X. Shenyang, J. Zhigang, S. Yishan, Practical pilot pattern design for sparse OFDM channel estimation, in: 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), IEEE, 2018, pp. 873–876.
[35] Y. Nie, X. Yu, Z. Yang, Deterministic pilot pattern allocation optimization for sparse channel estimation based on cs theory in OFDM system, EURASIP J. Wirel. Commun. Netw. 2019 (1) (2019) 7.
[36] T. Yaacoub, O.A. Dobre, R. Youssef, E. Radoi, Optimal selection of fourier coefficients for compressed sensing-based UWB channel estimation, IEEE Wirel. Commun. Lett. 6 (4) (2017) 466–469.
[37] M.A. Matin, Ultra Wideband Communications: Novel Trends-Antennas and Propagation, BoD–Books on Demand, 2011.
[38] L. Welch, Lower bounds on the maximum cross correlation of signals, IEEE Trans. Inf. Theory 20 (3) (1974) 397–399.
[39] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[40] M.-W. Park, Y.-D. Kim, A systematic procedure for setting parameters in simulated annealing algorithms, Comput. Oper. Res. 25 (3) (1998) 207–217.
[41] Y. Liu, W. Mei, H. Du, Two compressive sensing-based estimation schemes designed for rapidly time-varying channels in orthogonal frequency division multiplexing systems, IET Signal Proc. 8 (3) (2014) 291–299.
[42] S. Al-Shuhail, Combating Impairments in Multi-carrier Systems: A Compressed Sensing Approach, 2015 Ph.D. thesis.
[43] I. Barhumi, G. Leus, M. Moonen, Optimal training design for MIMO OFDM systems in mobile wireless channels, IEEE Trans. Signal Process. 51 (6) (2003) 1615–1624.
[44] X. Cai, G.B. Giannakis, Error probability minimizing pilots for OFDM with M-PSK modulation over rayleigh-fading channels, IEEE Trans. Veh. Technol. 53 (1) (2004) 146–155.
[45] W. Zhang, X.-G. Xia, P.-C. Ching, Optimal training and pilot pattern design for OFDM systems in rayleigh fading, IEEE Trans. Broadcast. 52 (4) (2006) 505–514.