Signal Processing 196 (2022) 108509

Contents lists available at ScienceDirect

Signal Processing

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

Low-complexity soft ML detection for generalized spatial modulation

M. Ángeles Simarro [a,*], Víctor M. García-Mollá [b], F.J. Martínez-Zaldívar [a], Alberto Gonzalez [a,1]

[a] ITEAM, Universitat Politécnica de Valéncia, Spain
[b] DSIC, Universitat Politécnica de Valéncia, Spain

ARTICLE INFO

Article history:
Received 8 July 2021
Revised 11 November 2021
Accepted 13 February 2022
Available online 16 February 2022

Keywords:
Generalized spatial modulation (GSM)
Multiple-Input multiple-Output (MIMO)
Low-complexity
Soft-output

ABSTRACT

Generalized Spatial Modulation (GSM) is a recent Multiple-Input Multiple-Output (MIMO) scheme, which achieves high spectral and energy efficiencies. Specifically, soft-output detectors have a key role in achieving the highest coding gain when an error-correcting code (ECC) is used. Nowadays, soft-output Maximum Likelihood (ML) detection in MIMO-GSM systems leads to a computational complexity that is unfeasible for real applications; however, it is important to develop low-complexity decoding algorithms that provide a reasonable computational simulation time in order to make a performance benchmark available in MIMO-GSM systems. This paper presents three algorithms that achieve ML performance. In the first algorithm, different strategies are implemented, such as a preprocessing sorting step in order to avoid an exhaustive search. In addition, clipping of the extrinsic log-likelihood ratios (LLRs) can be incorporating to this algorithm to give a lower cost version. The other two proposed algorithms can only be used with clipping and the results show a significant saving in computational cost. Furthermore clipping allows a wide-trade-off between performance and complexity by only adjusting the clipping parameter.

1. Introduction

Multiple-Input Multiple-Output (MIMO) systems have been widely used because of their remarkable capacity to improve the transmission rate and reliability of wireless communications [1,2]. More recently, a concept known as Spatial Modulation (SM) [3,4] has emerged, which is a promising digital modulation technology to achieve an attractive tradeoff between spectral efficiency and energy efficiency. The SM technique uses the spatial domain to transmit information in addition to classical signal constellations, activating only one transmit antenna per channel usage. Based on SM, many different works have been proposed [5–9]. SM can be further generalized as the Generalized Spatial Modulation (GSM) [10–12], which is capable of achieving higher spectral efficiency compared to the conventional SM. GSM activates more than one transmit antenna in each time slot, achieving higher transmission rates that grow considerably faster with the increase in the number of transmit antennas.

GSM uses a subset of Transmit Antennas (TAs) in each time slot, so the information bits are split into two parts: the first part selects the subset of activated TAs, and the second part generates the modulation symbols that are transmitted by the active antennas. Thus, the detection process on the receiver side becomes complicated because both the subset of the TAs as well as the symbol vector have to be estimated.

Several hard-output detectors have been proposed for MIMO-GSM systems [13–17]. Some of them use different strategies that are aided by successive sphere decoders that achieve fast ML detection. Examples of detectors of this type are the Sorting Assisted Successive Sphere Decoding Algorithm (SA-SSDA) [14] and the Box Optimization Hard Decoder aided by Box-Optimization bound (BOHD) [16]. In practice, almost all wireless communications systems use an error-correcting code (ECC) to enhance data reliability, which requires soft-output demodulation algorithms to achieve the highest coding gain. To this end, several soft-output detectors have been proposed for MIMO-GSM systems. The optimal Maximum Likelihood (ML) soft-output decoder for MIMO-GSM systems requires an exhaustive search over all active antenna and modulated symbol combinations, making it impractical for most applications. To reduce the computational cost, several low-complexity soft-output ML implementations have been reported for MIMO-GSM systems where different meaningful strategies have been employed [18–21].

As discussed above, the soft-output ML detection becomes impractical for MIMO-GSM, especially when the number of possible TA combinations or the number of transmit antennas is large. On the other hand, since the computation complexity of soft-output ML detectors for MIMO-GSM is too high, the use of clip-

* Corresponding author.
E-mail addresses: mdesiha@iteam.upv.es (M. Ángeles Simarro), vmgarcia@dsic.upv.es (V.M. García-Mollá), fjmartin@dcom.upv.es (F.J. Martínez-Zaldívar), agonzal@iteam.upv.es (A. Gonzalez).
[1] EURASIP Member.

https://doi.org/10.1016/j.sigpro.2022.108509
0165-1684/

ping [22] allows the detection performance and complexity to be conveniently adjusted, making the algorithms suitable for practical applications. Hence, one of the main challenges is to provide low-complexity detectors with negligible performance loss. Even though this task has been widely discussed in the literature, it is also necessary to provide fast decoders to perform soft-output ML detection in MIMO-GSM systems. Among other uses, researchers must have a benchmark bit error rate (BER) with a reasonable computational cost in order to reduce the simulation time when investigating their MIMO-GSM proposals. To the best of the authors' knowledge, none of the soft-output detectors already proposed address the issue of optimal ML detection.

In this context, this work aims to provide meaningful contributions to the design of soft-output ML detectors for MIMO-GSM systems, presenting three algorithms that achieve the ML performance. The first one can work with and without clipping, whereas the other two can be used only for the clipped case. Throughout this work, we reformulate the MIMO-GSM problem for soft-output detection, and we apply several strategies that are used in hard-output detection [14,16] in order to provide soft-output ML detection with manageable computational cost.

In the first proposed algorithm, a sorting step is carried out first to sort the TA combinations. Also, the log-likelihood ratios (LLRs) are suboptimally calculated at this stage to reduce the computational complexity. In the second stage, the LLRs are updated to their optimal values through recursive tree searches. A modified version of the Single Tree Search (STS) algorithm [23] is used for this purpose. Note that the target of the proposed algorithm is to achieve the soft-output ML solution with an manageable computational cost. In addition, the clipping of the LLRs in the tree search for optimal solutions has been incorporated in this algorithm.

The two alternative proposals have been developed using clipping. Both algorithms are based on the Double Tree Search (DTS) detector proposed in [24] for MIMO systems. This detector achieves the clipped soft-output ML solution at a reduced computational cost because it takes advantage of the fast Box Optimization Hard Detector (BOHD) algorithm [25]. The results show that these two different proposals are very effective in reducing the computational complexity.

The remainder of this paper is organized as follows. Section 2 presents the system model and the soft-output ML detection for MIMO-GSM. Section 3 explains the method proposed for reducing the computational complexity of the soft-output ML detection without clipping. Section 4 describes the proposed method when clipping is used. Two additional methods are also outlined for the clipping case. Section 5 presents the experimental results of applying the proposed algorithms for different simulation setups. Finally, the conclusions are presented in Section 6.

2. MIMO-GSM with soft-output ML detection

Consider a MIMO-GSM system equipped with N_t and N_r transmit and receive antennas, respectively. At each slot time, only N_a transmit antennas are activated, where $N_r \geq N_a$ and $N_t > N_a > 1$ are assumed without loss of generality. Therefore, the total number of possible TA combinations is $\binom{N_t}{N_a}$. Nevertheless, not all of the possible combinations are valid.

The block diagrams of the MIMO-GSM transmitter and receiver are shown in Fig. 1. A data frame of source information bits $\mathbf{b} = [b_1, b_2, \ldots, b_B]$ is encoded by a channel code to generate a codeword $\mathbf{c} = [c_1, c_2, \ldots, c_N]$. Then, the codeword is interleaved with a random bit interleaver. Afterwards, $n = n_1 + n_2$ interleaved coded bits are used for each transmission and divided into two parts. The first part (composed by n_1 bits) is used to activate certain N_a antennas of N_t. Thus, of the total number of possible configurations only $N_c = 2^{n_1}$ TA combinations are permitted. Let $\Gamma = (\zeta_1, \zeta_2, \ldots, \zeta_{N_c})$ be the set of valid TA combinations. Each valid ζ_k TA configuration, for $k = 1, \ldots, N_c$, can be described as a set of indices, $\zeta_k = \{i_{k_1}, i_{k_2} \ldots, i_{k_{N_a}}\}$ with $1 \leq i_{k_j} \leq N_t$ and $j = 1, \ldots, N_a$. The second part (composed of $n_2 = N_a \log_2(M)$ bits) is used to modulate the N_a modulation symbols to be transmitted by the activated antennas. The symbols are taken from a M-ary Quadrature Amplitude Modulation (M-QAM) denoted as Ω.

Let $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ and $\mathbf{v} \in \mathbb{C}^{N_r \times 1}$ be the MIMO channel matrix and noise vector, whose elements follow the complex Gaussian distribution with $\mathcal{CN}(0,1)$ and $\mathcal{CN}(0,\sigma^2)$, respectively. If the transmission is carried out through the ζ_k^{th} valid TA combination, the corresponding channel submatrix can be defined as $\mathbf{H}_{\zeta_\mathbf{k}} \in \mathbb{C}^{N_r \times N_a}$. Therefore, the receiver signal $\mathbf{y} \in \mathbb{C}^{N_r \times 1}$ is given by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} = \mathbf{H}_{\zeta_\mathbf{k}}\mathbf{s} + \mathbf{v}, \tag{1}$$

where $\mathbf{x} = [\ldots, 0, s_1, \ldots, 0, s_2, \ldots, 0, s_{N_a}, 0, \ldots]^T$ is the transmitted symbol vector and $\mathbf{s} = [s_1, \ldots, s_{N_a}]^T$ is the transmitted M-QAM symbol vector, which corresponds to the ζ_k antenna configuration indices.

On the receiver side, soft information in the form of LLRs is sent from the soft-output demodulator to the channel decoder, as can be observed in Fig. 1. In this work, we focus on the design of this demodulator. Given the received signal vector, the soft-output demodulator can compute the LLRs of the encoded bits denoted as $L_e(c_u)$ for $u = 1, \ldots, n$ as

$$L_e(c_u) = \ln\left[\frac{\sum_{\chi \in \chi_1^u} \exp(-\frac{||\mathbf{y}-\mathbf{H}\mathbf{x}||^2}{\sigma^2} - \sum_{j \neq u} c_j L_a(c_j))}{\sum_{\chi \in \chi_0^u} \exp(-\frac{||\mathbf{y}-\mathbf{H}\mathbf{x}||^2}{\sigma^2} - \sum_{j \neq u} c_j L_a(c_j))}\right] \tag{2}$$

where χ_1^u and χ_0^u are the subsets of the MIMO-GSM signal χ of size Ω^{N_t}, fulfilling $\chi_1^u = \{\mathbf{x} \in \chi : c_u = 1\}$ and $\chi_0^u = \{\mathbf{x} \in \chi : c_u = 0\}$, respectively. $L_a(c_j)$ represents the a priori information. Since we assume that no prior information is available in the detection process, all valid TAs and all constellation points are considered to be equally likely. Thus, the term $\sum_{j \neq u} c_j L_a(c_j)$ can be omitted from Eq. (2). Omitting this term and denoting the metric corresponding to the $\mathbf{x}$ MIMO-GSM vector as $d^u = \frac{||\mathbf{y}-\mathbf{H}\mathbf{x}||^2}{\sigma^2}$, Eq. (2) can be rewritten as

$$L_e(c_u) = \ln\left[\frac{\sum_{\chi \in \chi_1^u} \exp(-d^u)}{\sum_{\chi \in \chi_0^u} \exp(-d^u)}\right] \tag{3}$$

Furthermore, the well-known Max-Log MAP detection rule [26] can be applied and (3) can be simplified as

$$L_e(c_u) = \min_{\sum_{\chi \in \chi_1^u}} (d^u) - \min_{\sum_{\chi \in \chi_0^u}} (d^u). \tag{4}$$

For each bit c_u, one of the two minima in (4) corresponds to the hard-output MAP detection problem expressed by $\hat{\mathbf{x}}^{ML} = \arg\min_{\mathbf{x} \in \chi} \frac{||\mathbf{y}-\mathbf{H}\mathbf{x}||^2}{\sigma^2}$ with the associated distance $d^{ML} = \frac{||\mathbf{y}-\mathbf{H}\hat{\mathbf{x}}^{ML}||^2}{\sigma^2}$. Thus, the second minimum in (4) can be obtained as

$$\bar{d}^u = \min_{\mathbf{x} \in \chi^u_{\bar{c}_u^{ML}}} \frac{||\mathbf{y} - \mathbf{H}\mathbf{x}||^2}{\sigma^2}, \tag{5}$$

where $\chi^u_{\bar{c}_u^{ML}}$ is the set of the MIMO-GSM signal χ, fulfilling $\chi^u_{\bar{c}_u^{ML}} = \{\mathbf{x} \in \chi : c_u = \bar{c}_u^{ML}\}$, with $\bar{c}_u^{ML}$ being the binary complement of the c_u bit of the ML solution. We call these other minimum distances as counter-hypothesis distances $\bar{d}^u$ [23]. Therefore, the LLR in (4) can be computed as

$$L_e(c_u) = (d^{ML} - \bar{d}^u)(1 - 2c_u^{ML}). \tag{6}$$

A trivial approach for solving (6) would be to use an extended constellation, using "0" for the detection of inactive antennas and using a standard soft-output ML detector such as the Single Tree Search (STS) [23] or the Box Optimization Repeated Tree Search (BORTS) [24]. However, the search complexity of these algorithms

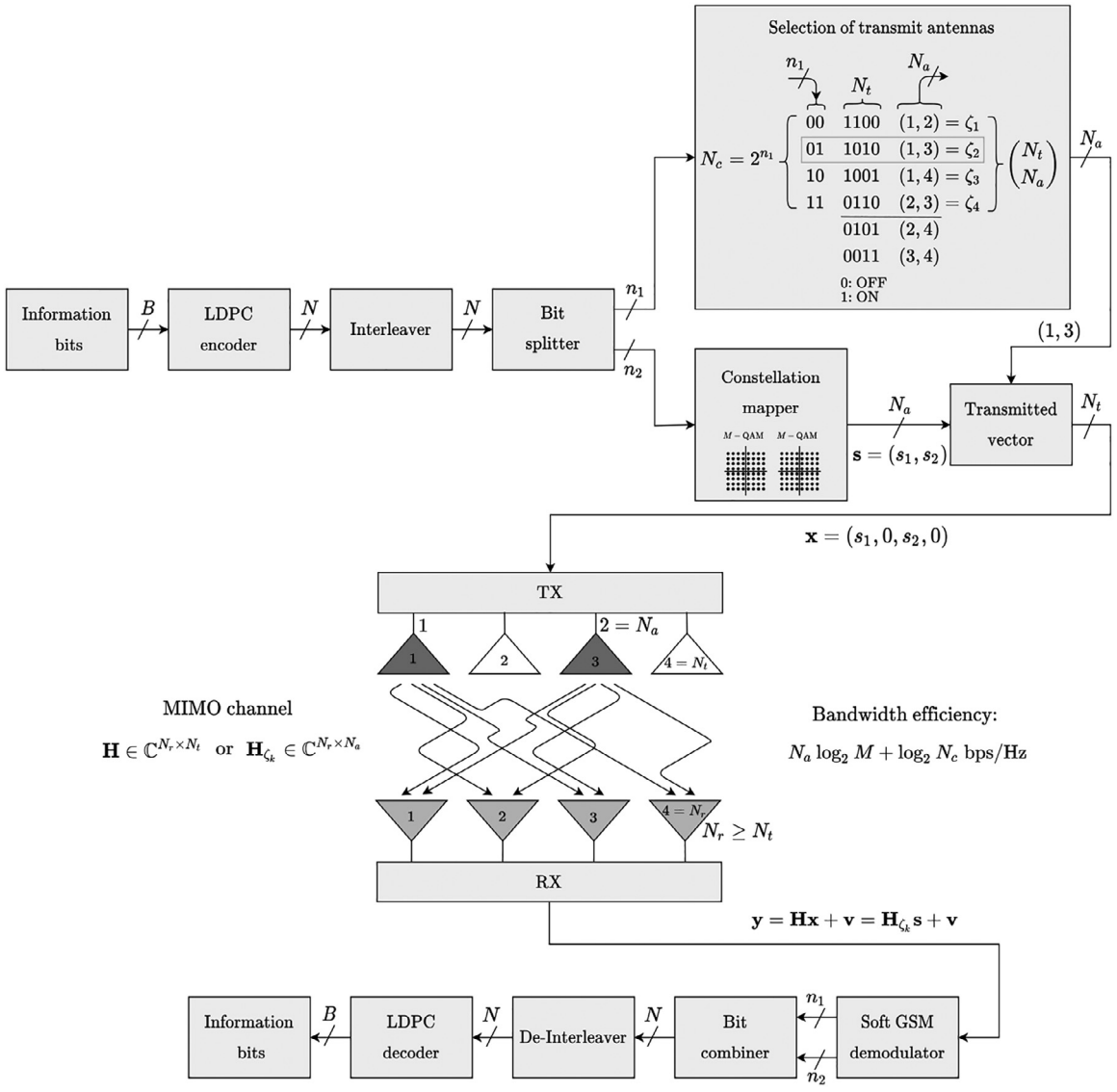

Fig. 1. LDPC coded MIMO-GSM system.

will become prohibitively high when N_t or M increases. Furthermore, these methods cannot be easily applied when $N_t > N_r$ because it is not possible to compute the required triangular factorization of the channel matrix which is needed for the tree detection.

3. The proposed ML strategy for soft-output detection without clipping

The excessive complexity of the MIMO-GSM soft-output ML detection limits its application. To reduce the complexity while maintaining optimal performance, we propose a new two-stage algorithm. The main idea, which is also used by some hard-output detection algorithms such as SA-SSDA [14] or BOHD [16], is to split the problem into N_c subproblems. Thus, for each valid TA configuration, we can obtain a smaller problem to solve. To do this, we must reformulate the soft-output MIMO-GSM detection problem.

According to (1) and assuming equally likely a priori values, the metrics d^u in (4) depend on the ζ_k combination and can also be calculated as

$$d^u_{\zeta_k} = \frac{||\mathbf{y} - \mathbf{H}_{\zeta_k}\mathbf{s}||^2}{\sigma^2}. \tag{7}$$

Then, the LLRs of the bits that select the TA combination $L_e(c_t)$ with $t = (1, \ldots, n_1)$ can be computed as

$$L_e(c_t) = \min_{\zeta_k \in \Gamma_1^k, \mathbf{s} \in \mathbb{S}} (d^t_{\zeta_k}) - \min_{\zeta_k \in \Gamma_0^k, \mathbf{s} \in \mathbb{S}} (d^t_{\zeta_k}) \tag{8}$$

where Γ_1^k and Γ_0^k represent a subspace of the total TA combinations Γ of size N_c, fulfilling that $\Gamma_1^k = \{\zeta_k \in \Gamma : c_t = 1\}$ and $\Gamma_0^k = \{\zeta_k \in \Gamma : c_t = 0\}$, respectively.

Likewise, the LLRs of the n_2 modulated bits, $L_e(c_r)$ with $r = (n_1 + 1, \ldots, n)$ can be computed as

$$L_e(c_r) = \min_{\zeta_k \in \Gamma, \mathbf{s} \in \mathbb{S}_1^r} (d^r_{\zeta_k}) - \min_{\zeta_k \in \Gamma, \mathbf{s} \in \mathbb{S}_0^r} (d^r_{\zeta_k}) \tag{9}$$

where $\mathbb{S}_1^r$ and $\mathbb{S}_0^r$ represent a subspace of the set signalling $\mathbb{S}$ of size Ω^{N_a}, fulfilling that $\mathbb{S}_1^r = \{\mathbf{s} \in \mathbb{S} : c_r = 1\}$ and $\mathbb{S}_0^r = \{\mathbf{s} \in \mathbb{S} : c_r = 0\}$, respectively.

As in (4), for each bit c_t and c_r, one of the two minima in (9) and (8) corresponds to the hard-output MAP detection problem.

$$(\hat{\zeta}_k^{ML}, \hat{\mathbf{s}}^{ML}) = \arg \min_{\zeta_k \in \Gamma, \mathbf{s} \in \mathbb{S}} \frac{||\mathbf{y} - \mathbf{H}_{\zeta_k}\mathbf{s}||^2}{\sigma^2} \tag{10}$$

$$d^{ML} = \frac{||\mathbf{y} - \mathbf{H}_{\hat{\zeta}_k^{ML}}\hat{\mathbf{s}}^{ML}||^2}{\sigma^2}. \tag{11}$$

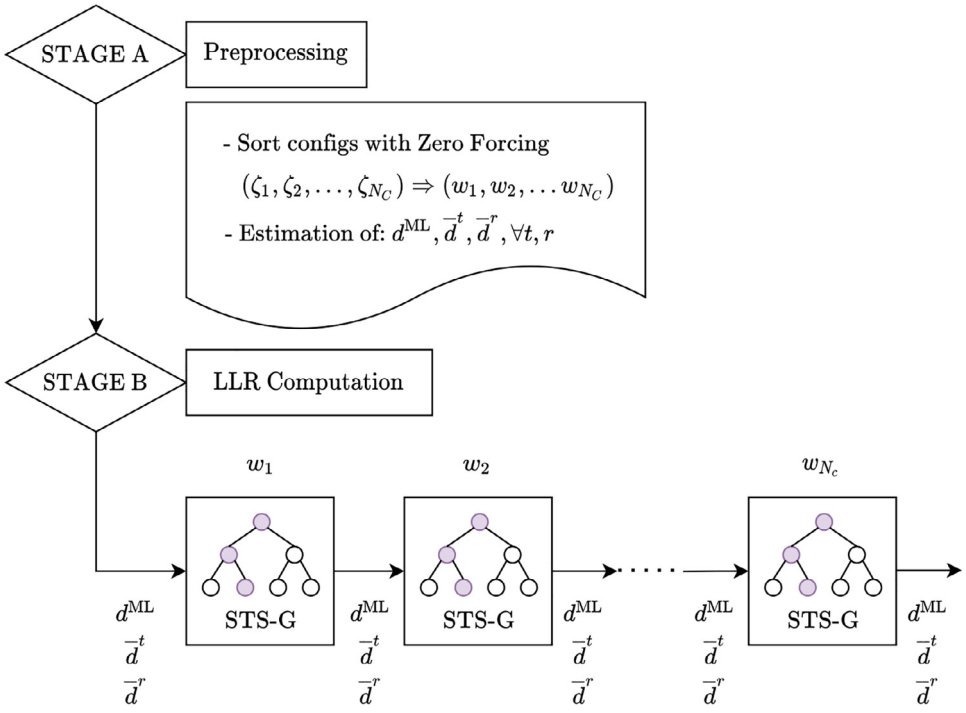

Fig. 2. Illustration of the procedure of the OSTS algorithm.

Thus, by denoting c_t^{ML} as the tth bit used to select the $\hat{\zeta}_k^{\,ML}$ TA combination, (8) can be computed as

$$L_e(c_t) = (d^{ML} - \bar{d}^t)(1 - 2c_t^{ML}). \tag{12}$$

Likewise, by denoting c_r^{ML} as rth bit in the $\hat{\mathbf{s}}^{ML}$ solution, (9) can be computed as

$$L_e(c_r) = (d^{ML} - \bar{d}^r)(1 - 2c_r^{ML}), \tag{13}$$

With $\bar{d}^t$ and $\bar{d}^r$ in (12) and (13) being the counter-hypothesis distances associated with the TA combination and the transmitted symbol, respectively. These counter-hypothesis distances are calculated as:

$$\bar{d}^t = \min_{\zeta_k \in \Gamma_{\bar{c}_t^{ML}}^t, \mathbf{s} \in \mathbb{S}} \frac{||\mathbf{y} - \mathbf{H}_{\zeta_\mathbf{k}}\mathbf{s}||^2}{\sigma^2} \tag{14}$$

$$\bar{d}^r = \min_{\zeta_k \in \Gamma, \mathbf{s} \in \mathbb{S}_{\bar{c}_r^{ML}}^r} \frac{||\mathbf{y} - \mathbf{H}_{\zeta_\mathbf{k}}\mathbf{s}||^2}{\sigma^2}. \tag{15}$$

In the counter-hypothesis distances, $\bar{c}_t^{ML}$ and $\bar{c}_r^{ML}$ denote the binary complement of the c_t and c_r bit in the label of $\hat{\zeta}_k^{ML}$ and $\hat{\mathbf{s}}^{ML}$, respectively.

Thus, the detection process computes the values of d^{ML}, $\hat{\zeta}_k^{\,ML}$, $\hat{\mathbf{s}}^{ML}$, $\bar{d}^t$ $\forall t$, and $\bar{d}^r$ $\forall r$. These values could be solved for each antenna configuration ζ_k. Then, by comparing the solutions of all TA configurations, we can calculate Eqs. (12) and (13). To solve each subproblem, the STS algorithm can be considered a suitable detector. It ensures that each node in the tree search is visited only once when searching for the ML solution and all the distances of the counter-hypotheses simultaneously. The number of calculated nodes of the STS compared to other algorithms (such as the RTS) is significantly lower. However, applying the STS algorithm for each subproblem ζ_k can be computationally expensive because N_c different subproblems must be totally solved. To further reduce the detection complexity, the proposed algorithm (called Ordered Single Tree Search (OSTS)) applies different strategies in two stages. The procedure of the algorithm is illustrated in Fig. 2, and its pseudocode implementation is given in Algorithm 1.

Algorithm 1: The proposed Ordered Single Tree Search for GSM Algorithm (OSTS).

Input: $\mathbf{y}$, $\mathbf{H}_{\zeta_\mathbf{k}}$, $\mathbf{Q}_{\zeta_\mathbf{k}}$, $\mathbf{R}_{\zeta_\mathbf{k}}$, $\forall k$, Ω
Output: $L_e(c_u)$ $\forall n$
/* Stage A */
for $k = 1 : N_c$ **do**
  $\mathbf{z}_{\zeta_\mathbf{k}} = \mathbf{Q}_{\zeta_k}\mathbf{y}$
  $\hat{\mathbf{z}}_{\zeta_\mathbf{k}} = \mathcal{Q}\left(\mathbf{R}_{\zeta_\mathbf{k}}{}^{-1}\mathbf{z}_{\zeta_\mathbf{k}}\right)$
  $d_{\zeta_k} = \|\mathbf{y} - \mathbf{H}_{\zeta_\mathbf{k}}\hat{\mathbf{z}}_{\zeta_\mathbf{k}}\|^2$
$\mathbf{d} = [d_{\zeta_1}, d_{\zeta_2}, \ldots, d_{\zeta_{N_c}}]$
$[\omega_1, \omega_2, \ldots, \omega_{N_c}] = \arg\mathrm{sort}(\mathbf{d})$;
$\hat{\zeta}_k^{\,\mathrm{ML}} = \omega_1$; $d^{\mathrm{ML}} = \mathbf{d}(\omega_1)$; $\hat{\mathbf{s}}^{\mathrm{ML}} = \hat{\mathbf{z}}_{\hat{\zeta}_\mathbf{k}^{\,\mathrm{ML}}}$
$\bar{d}^t = \min_{\bar{c}_t^{\mathrm{ML}}}(\mathbf{d})$ $\forall t$
$\bar{d}^r = \min_{\bar{c}_r^{\mathrm{ML}}}(\mathbf{d})$ $\forall r$
/* Stage B */
for $i = 1 : N_c$ **do**
  $[d^{\mathrm{ML}}, \hat{\zeta}_k^{\,\mathrm{ML}}, \hat{\mathbf{s}}^{\mathrm{ML}}, \bar{d}^t, \bar{d}^r]$= STS_G($\mathbf{z}_{\omega_i}$, $\mathbf{R}_{\omega_i}$, d^{ML}, $\hat{\zeta}_k^{\,\mathrm{ML}}$, $\hat{\mathbf{s}}^{\mathrm{ML}}$, $\bar{d}^t$, $\bar{d}^r$, ζ_k)
$L_e(c_t) = (d^{\mathrm{ML}} - \bar{d}^t)(1 - 2c_t^{\mathrm{ML}})$ $\forall t$
$L_e(c_r) = (d^{\mathrm{ML}} - \bar{d}^r)(1 - 2c_r^{\mathrm{ML}})$ $\forall r$

3.1. Description of the OSTS algorithm

3.1.1. Stage A: Sorting and initializing step

In hard-output detection [14,16], the use of an adjustable radius across all of the subproblems and the reordering of the different ζ_k configurations achieves a large reduction in complexity. A similar strategy is implemented in the proposed soft-output detector.

First, the configurations must be sorted so that the correct configuration is detected in the first positions. The sorting method proposed in [14] uses the matrix $Q_{\zeta_{k2}}$ of each configuration ζ_k

which is obtained after the QR-decomposition of the channel matrix $\mathbf{H}_{\zeta_k}$. The QR decomposition gives a unitary matrix $\mathbf{Q}_{\zeta_\mathbf{k}} \in \mathbb{C}^{N_r x N_r}$ and an upper triangular matrix $\mathbf{R}_{\zeta_\mathbf{k}} \in \mathbb{C}^{N_r x N_a}$. Given that the last $N_r - N_a$ rows of $\mathbf{R}_{\zeta_\mathbf{k}}$ are zeros, the QR decomposition is usually rewritten as

$$\mathbf{H}_{\zeta_\mathbf{k}} = \mathbf{Q}_{\zeta_\mathbf{k}}\mathbf{R}_{\zeta_\mathbf{k}} = \mathbf{Q}_{\zeta_\mathbf{k}}\begin{pmatrix}\mathbf{R}_{\zeta_\mathbf{k1}}\\ 0\end{pmatrix} = \left(\mathbf{Q}_{\zeta_\mathbf{k1}}\mathbf{Q}_{\zeta_\mathbf{k2}}\right)\begin{pmatrix}\mathbf{R}_{\zeta_\mathbf{k}1}\\ 0\end{pmatrix} \tag{16}$$

where $\mathbf{R}_{\zeta_\mathbf{k1}} \in \mathbb{C}^{N_a x N_a}$, $\mathbf{Q}_{\zeta_\mathbf{k1}} \in \mathbb{C}^{N_r x N_a}$ and $\mathbf{Q}_{\zeta_\mathbf{k2}} \in \mathbb{C}^{N_r x (N_r - N_a)}$. Thus,

$$\begin{aligned}||\mathbf{y} - \mathbf{H}_{\zeta_\mathbf{k}}\mathbf{s}||^2 &= ||\mathbf{Q}^\mathbf{H}_{\zeta_\mathbf{k}} \cdot (\mathbf{y} - \mathbf{Q}_{\zeta_\mathbf{k}}\mathbf{R}_{\zeta_\mathbf{k}}\mathbf{s})||^2 = ||\begin{pmatrix}\mathbf{Q}^\mathbf{H}_{\zeta_\mathbf{k1}}\\ \mathbf{Q}^\mathbf{H}_{\zeta_\mathbf{k2}}\end{pmatrix}\mathbf{y} - \begin{pmatrix}\mathbf{R}_{\zeta_\mathbf{k1}}\\ 0\end{pmatrix}\mathbf{s}||^2\\ &= ||\mathbf{Q}^\mathbf{H}_{\zeta_\mathbf{k1}}\mathbf{y} - \mathbf{R}_{\zeta_\mathbf{k1}}\mathbf{s}||^2 + ||\mathbf{Q}^\mathbf{H}_{\zeta_\mathbf{k2}}\mathbf{y}||^2.\end{aligned} \tag{17}$$

However, this method cannot be used if $N_a = N_r$ because the term used for the sorting ($||\mathbf{Q}^\mathbf{H}_{\zeta_\mathbf{k2}}\mathbf{y}||^2$) does not exist. On the other hand, [16] proposes a simple but efficient sorting method that can be used without any limitation. This ordering strategy, which is based on the Zero Forcing (ZF) estimator of each subproblem is used in the proposed algorithm:

$$\mathbf{z}_{\zeta_\mathbf{k}} = \mathbf{Q}_{\zeta\,k}\mathbf{y} \tag{18}$$

$$\hat{\mathbf{z}}_{\zeta_\mathbf{k}} = \mathcal{Q}\left(\mathbf{R}_{\zeta_\mathbf{k}}{}^{-1}\mathbf{z}_{\zeta_\mathbf{k}}\right), \tag{19}$$

where $\mathcal{Q}(\cdot)$ gives the nearest constellation symbol. Then, the Euclidean distance associated at each estimator $d_{\hat{z}_{\zeta_k}}$ is computed and stored in the vector of distances $\mathbf{d} = [d_{\hat{z}_{\zeta_1}}, d_{\hat{z}_{\zeta_2}}, \ldots, d_{\hat{z}_{\zeta_{N_c}}}]$.

The valid TA combinations are then sorted according to the Euclidean distance, from smallest to largest,

$$[\omega_1, \omega_2, \ldots, \omega_{N_c}] = \arg\operatorname{sort}(\mathbf{d}) \tag{20}$$

where $\operatorname{sort}(\dot{})$ defines an ordering function for reordering the elements of the input vector in ascending order, and ω_1 and ω_{N_c} are the indices of the maximum and minimum value in $\mathbf{d}$, respectively.

In addition, to find the ordered TA combinations, the initial and suboptimal values of d^{ML}, $\hat{\zeta}_k^{\,ML}$, $\hat{\mathbf{s}}^{ML}$, $\bar{d}^t$ and $\bar{d}^r$ are computed in this stage. The ZF estimators of the different subproblems are used to compute these approximations according to

$$\hat{\zeta}_k^{\,ML} = \omega_1; \quad d^{ML} = \mathbf{d}(\omega_1); \quad \hat{\mathbf{s}}^{ML} = \hat{\mathbf{z}}_{\hat{\zeta}_\mathbf{k}^{\,\mathbf{ML}}} \tag{21}$$

$$\bar{d}^t = \min_{\bar{c}_t^{ML}}(\mathbf{d}) \quad \forall t \tag{22}$$

$$\bar{d}^r = \min_{\bar{c}_r^{ML}}(\mathbf{d}) \quad \forall r. \tag{23}$$

Thus, in this step, the different configurations have been sorted and the main parameters needed by the STS algorithm have been initialised to suboptimal values. By means of these two strategies, the computational cost of the second stage can be reduced.

3.1.2. Stage B: Exact LLR computation

After the preprocessing stage, a modified STS algorithm is used to solve each ML ζ_k subproblem. The STS_G function in Algorithm 1 denotes the STS algorithm proposed in [23] by adding the following adjustments.

- *List administration:* The original STS uses a list administration to update the counter-hypothesis distances of the transmitted bits $\bar{d}^r$ in order to find the ML solution and all of the counter-hypothesis distances simultaneously. In our case, it is necessary to use two list administrations: one for the transmitted bits, and one for the bits that select the TA combination. Therefore, when a leaf node has been reached, the decoder must also update the second list administration with one of two options:
  - If a new ML hypothesis is found, all $\bar{d}^t$ such that $c_t = \bar{c}_t^{ML}$ are set to d^{ML}.
  - Otherwise, all $\bar{d}^t$ with $c_t = \bar{c}_t^{ML}$ and $d(\mathbf{s}) < \bar{d}^t$ are set to $d(\mathbf{s})$, with $d(\mathbf{s})$ being the Euclidean distance of the leaf node.
- *Pruning criterion:* Another important key to complexity reduction in the STS algorithm is the pruning criterion. This criterion is based on the fact that a subtree that originated from a given node is only visited if the ML distance or the counter-hypothesis distances of the transmitted bits are going to be updated. Therefore, we must modify this pruning criterion to also take into account the counter-hypothesis distances of the bits that select the TA combination. Thus, in addition to the conditions described in [23], the pruning radius r must be updated according to

$$r = \max\{r, \bar{d}^t | (t = 1, \ldots, n_1) \wedge (c_t = \bar{c}_t^{ML})\}. \tag{24}$$

In summary, for each ζ_k configuration, the STS_G algorithm is executed, and the order of execution of ζ_k is the one given in Stage A by ω_i. In addition, in order to reduce the computational complexity, the main approach is not to initialise the values of d^{ML}, $\hat{\zeta}_k^{\,ML}$, $\hat{\mathbf{s}}^{ML}$, $\bar{d}^t$ and $\bar{d}^r$, as in the original STS algorithm. For each configuration, these initial values are taken from the output of the previous solved configuration ω_{i-1} or from the values of Stage A if it is the first subproblem to be solved. Therefore, for each configuration, the distances used are lower bounded, reducing the computational complexity of the following subproblems. For this reason, it is important to place the correct configuration in the first positions.

4. The proposed ML strategy for soft-output detection with clipping

In practical applications, the complexity of the max-log LLR can be reduced by the use of clipping [22]. Therefore, given a clipping parameter L_{clip}, the dynamic range of LLRs is bounded according to

$$|L_e(c_u)| \leq L_{clip}, \quad \forall u. \tag{25}$$

The key issue is to analyse that (25) in conjunction with (12) and (13) results in an upper bound of the counter-hypothesis distances $\bar{d}^t$ and $\bar{d}^r$. Therefore, the counter-hypothesis distances that are larger than $d^{ML} + L_{clip}$ do not need to be computed exactly and can be set to the value $d^{ML} + L_{clip}$. Thus, the computational complexity can be reduced and becomes the case with clipping more relevant from a practical point of view. Obviously, this leads to performance degradation, so the value of L_{clip} can be used to suitably adjust the detection complexity/performance trade-off. Three different strategies for the case with clipping are presented in this work.

4.1. Description of the clipped OSTS

The proposed OSTS described above is easily adapted for the case with clipping. For this purpose, Stage A can be easily modified by replacing Eqs. (22) and (23) by the following:

$$\bar{d}^t = \min_{\bar{c}_t^{ML}}\{\mathbf{d}, d^{ML} + L_{clip}\} \quad \forall t \tag{26}$$

$$\bar{d}^r = \min_{\bar{c}_r^{ML}}\{\mathbf{d}, d^{ML} + L_{clip}\} \quad \forall r. \tag{27}$$

Stage B, the exact LLR computation is changed in order to compute clipped LLR values. This procedure can be inserted into the STS_G algorithm, as in the original STS algorithm [27] by just applying the next adjustment when a leaf node has been reached and a new ML hypothesis has been found:

$$\bar{d}^t = min\{\bar{d}^t, d^{ML} + L_{clip}\} \quad \forall t \tag{28}$$

$$\bar{d}^r = min\{\bar{d}^r, d^{ML} + L_{clip}\} \quad \forall r. \tag{29}$$

The remaining steps of the STS_G algorithm are not affected.

4.2. Description of the ODTS algorithm

The Double Tree Search (DTS) proposed in [24] modifies the original STS algorithm when the clipping strategy is applied. The DTS algorithm takes advantage of the fast BOHD algorithm [25] to first calculate the d^{ML} distance. Then, the STS is modified since it only needs to find the counter-hypothesis distances. Therefore, the computational cost is considerably reduced with respect to the original STS. We denote this new modified version of the STS_G algorithm as STS_GM.

To implement DTS for MIMO-GSM systems (ODTS), we first execute Stage A as in the clipped OSTS algorithm, in order to reduce the computational complexity of Stage B. After the preprocessing stage, the DTS algorithm proposed in [24] is used to solve each ML ζ_k subproblem, adding the following adjustments. The initial radius of the BOHD algorithm is the ML distance of the previous configuration since we use an adjustable and propagated radius to improve efficiency. Consequently, when the initial radius in the BOHD is smaller than the distances of all possible solutions, the detection ends very fast and no solution is returned. Thus, if the $d^{ML}_{\omega_i}$ solution given by the BOHD algorithm is lower than the propagated distance d^{ML}, the propagated distance and the counter-hypothesis distances must be updated according to

$$\bar{d}^t = \min\{\bar{d}^t, d^{ML}_{\omega_i} + L_{clip}\} \quad \forall t, \tag{30}$$

$$\bar{d}^r = \min\{\bar{d}^r, d^{ML}_{\omega_i} + L_{clip}\} \quad \forall r, \tag{31}$$

$$d^{ML} = d^{ML}_{\omega_i}. \tag{32}$$

Then, the STS_GM is used, which is the STS_G algorithm without the search of the ML solution since the solution has already been computed by the BOHD algorithm. Therefore, the STS_GM computes the counter-hypothesis distances that are lower than those already computed. The pseudocode is the same as Algorithm 1 modifying STS_G by DTS. The performance of the algorithm is shown in Fig. 3.

4.3. Description of the BO-DTS algorithm

Box optimization (BO) has been proposed as an aid to the MIMO detector in different works [24,25,28]. The MIMO detection problem is given by:

$$\mathbf{x}^{\mathrm{ML}} = \underset{\mathbf{x}\in\Omega^{N_t}\subset\mathbb{C}^{N_t}}{\arg\min} \|\mathbf{H}\cdot\mathbf{x}-\mathbf{y}\|^2. \tag{33}$$

Then, the auxiliary box optimization problem can be stated as:

$$\hat{\mathbf{xr}} = \underset{\substack{\mathbf{x}\in\mathbb{C}^{N_t} \\ \min(Re(\Omega))\le Re(x_i)\le\max(Re(\Omega)),1\le i\le m \\ \min(Im(\Omega))\le Im(x_i)\le\max(Im(\Omega)),1\le i\le m}}{\arg\min} \|\mathbf{H}\cdot\mathbf{x}-\mathbf{y}\|^2, \tag{34}$$

where $x_i, 1 \le i \le N_t$ are the components of the vector $\mathbf{x}$. This problem is derived from (33), discarding the condition that the components of the solution belong to the constellation Ω. The problem (34) is a continuous optimization problem and can be solved much faster than (33). The solution of this auxiliary problem provides extremely tight bounds that can be used to speed up standard sphere Decoders by radius shortening and branches pruning as shown in [24,25]. This technique can be also used for GSM problem detection. In hard-detection for GSM [16], it was used in order to achieve a lower bound of the minimum Euclidean distance for each configuration. This strategy can also be used for soft-detection to reduce the number of configurations that are totally analysed. Thus, in Stage A we can compute the solution of the continuous least squares problem for each configuration ζ_k as

$$\hat{\mathbf{sr}}_{\zeta_k} = \underset{\substack{\mathbf{s}\in\mathbb{S} \\ \min(Re(\Omega))\le Re(s_i)\le\max(Re(\Omega)),1\le i\le N_a \\ \min(Im(\Omega))\le Im(s_i)\le\max(Im(\Omega)),1\le i\le N_a}}{\arg\min} \left\|\mathbf{y}-\mathbf{H}_{\zeta_k}\mathbf{s}\right\|^2. \tag{35}$$

This problem is derived from (10) for each configuration ζ_k and discarding the condition that the components of the solution belong to the Ω constellation. In this case, the boundaries of the search area are box-shaped, hence the name Box Optimization

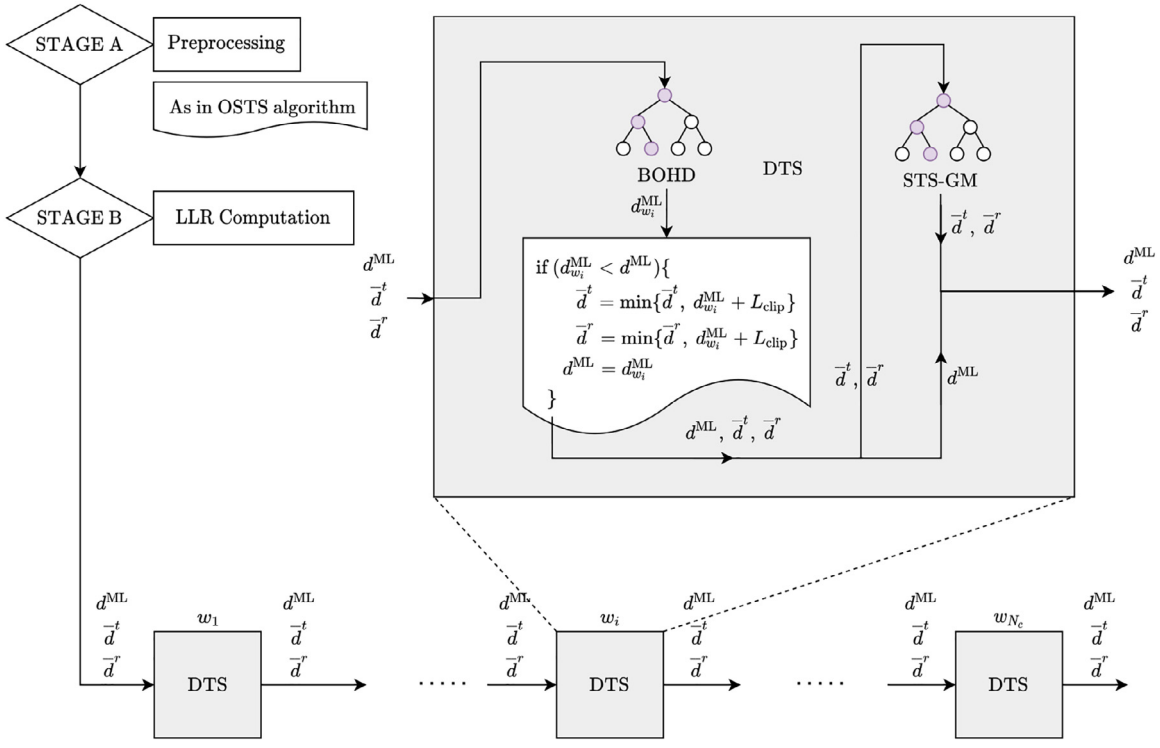

Fig. 3. Illustration of the procedure of the ODTS algorithm.

(BO). An efficient algorithm for solving this problem adapted to the MIMO problem was proposed in [25]. Therefore, for each configuration, a lower bound on the minimum Euclidean distance can be computed as $dr_{w_i} = \left\| \mathbf{y} - \mathbf{H}_{\zeta_\mathbf{k}} \hat{\mathbf{sr}}_{\zeta_k} \right\|^2$ and stored in the vector of distances $\mathbf{d}$. Thus, Stage A is modified and the different configurations are sorted according to the distance dr_{w_i} using the BO solver proposed in [25] to compute $\hat{\mathbf{sr}}_{\zeta_k}$ in line 2 of Algorithm 2.

Algorithm 2: The proposed Box-Optimization Assisted Double Tree Search for GSM Algorithm (BO-DTS).

Input: $\mathbf{y}$, $\mathbf{H}_{\zeta_\mathbf{k}}$, $k = 1, 2, \ldots, N_c$, Ω
Output: $L_e(c_u), u = 1, 2, \ldots, n$
/* Stage A */
1 **for** $k = 1 : N_c$ **do**
2 $\quad \hat{sr}_{\zeta_k} = BO(\mathbf{y}, \Omega, \mathbf{H}_{\zeta_\mathbf{k}})$
3 $\quad \hat{\mathbf{z}}_{\zeta_\mathbf{k}} = \mathcal{Q}(\hat{sr}_{\zeta_k})$
4 $\quad d_{\zeta_k} = \|\mathbf{y} - \mathbf{H}_{\zeta_\mathbf{k}} \hat{\mathbf{z}}_{\zeta_\mathbf{k}}\|^2$
5 $\quad dr_{\zeta_k} = \|\mathbf{y} - \mathbf{H}_{\zeta_\mathbf{k}} \hat{\mathbf{sr}}_{\zeta_\mathbf{k}}\|^2$
6 $\mathbf{d} = [dr_{\zeta_1}, dr_{\zeta_2}, \ldots, dr_{\zeta_{N_c}}]$
7 $[\omega_1, \omega_2, \ldots, \omega_{N_c}] = \arg\text{sort}(\mathbf{d})$;
8 $\hat{\zeta}_k^{\text{ML}} = \omega_1$; $d^{\text{ML}} = \mathbf{d}(\omega_1)$; $\hat{\mathbf{s}}^{\text{ML}} = \hat{\mathbf{z}}_{\hat{\zeta}_\mathbf{k}^{\text{ML}}}$
9 $\bar{d}^t = \min_{\bar{c}_t^{\text{ML}}}(\mathbf{d}) \quad \forall t$
10 $\bar{d}^r = \min_{\bar{c}_r^{\text{ML}}}(\mathbf{d}) \quad \forall r$
/* Stage B */
11 **for** $i = 1 : N_c$ **do**
/* Modified DTS Algorithm */
12 $\quad$ **if** $dr_{\omega_i} < d^{ML}$ **then**
13 $\quad\quad d_{\omega_i}^{\text{ML}}$= BOHD($\mathbf{y}$, $\mathbf{H}_{\zeta_\mathbf{k}}$, d^{ML})
14 $\quad\quad$ **if** $d_{\omega_i}^{ML} < d^{ML}$ **then**
15 $\quad\quad\quad d^{\text{ML}} = d_{\omega_i}^{\text{ML}}$
16 $\quad\quad\quad \hat{\zeta}_k^{\text{ML}} = \omega_i \bar{d}^r = \min(\bar{d}^r, d^{\text{ML}} + L_{clip}) \quad \forall r$
17 $\quad\quad\quad \bar{d}^t = \min(\bar{d}^t, d^{\text{ML}} + L_{clip}) \quad \forall t$
18 $\quad\quad$ **if** $d_{\omega_i}^{ML} < \min(\bar{d}^r, \bar{d}^t)$ **then**
19 $\quad\quad\quad [\bar{d}^t \ \bar{d}^r]$= STS_GM($\mathbf{y}$, $\mathbf{H}_{\zeta_\mathbf{k}}$, $\hat{\mathbf{s}}^{\text{ML}}$,$\bar{d}^t \ \bar{d}^r$, $\hat{\zeta}_k^{\text{ML}}$, ω_i)
20 $L_e(c_t) = (d^{\text{ML}} - \bar{d}^t)(1 - 2c_t^{\text{ML}}) \quad \forall t$
21 $L_e(c_r) = (d^{\text{ML}} - \bar{d}^r)(1 - 2c_r^{\text{ML}}) \quad \forall r$

Moreover, in Stage B, these distances can be used to discard configurations. In other words, when a new configuration ω_i is going to be explored, (i.e., a DTS algorithm is executed for ω_i), the distance d^{ML} has already been updated to the smallest distance found so far. Then, if the lower bound distance of this configuration dr_{w_i} is greater than the current d^{ML}, the w_i configuration can be safely ignored because the distance of any signal in this configuration will be greater than d^{ML} and consequently greater than $\bar{d}^t \ \forall t$ and $\bar{d}^r \ \forall r$. This procedure is graphically described in Fig. 4. The pseudocode implementation is given in Algorithm 2.

Table 2
Setups for Computed Simulations.

	N_t	N_a	N_r	M−QAM	N_c	bps/Hz
Setup 1	4	2	4	64	4	14
Setup 2	8	2	8	64	16	16
Setup 3	8	4	8	16	16	20
Setup 4	32	4	4	16	64	22

In addition, the dr_{w_i} distance can also be used within the DTS algorithm, to avoid the execution of STS_GM algorithm. This is due to the fact that, in some configurations, the d^{ML} distance will be larger than dr_{w_i}, and therefore we have to run the BOHD algorithm within the DTS block in Fig. 4. However, if there are no counter-hypothesis distances for $\bar{d}^t \ \forall t$ and for $\bar{d}^r \ \forall r$ lower than dr_{w_i}, the STS_GM algorithm in the DTS detector can be ignored since it will not find counter-hypothesis distances lower than those that have already been computed.

The main features of the proposed algorithms and the referenced algorithms are summarised in Table 1.

5. Simulation results

This section presents several numerical examples. We have chosen four different setups to test our proposals (see Table 2). The channels have been chosen independently and have been identically distributed Rayleigh fading. For the channel coding, an LDPC of rate 1/5 has been used as specified in TS 38.212 [29]. The results of the Monte Carlo simulations are presented throughout this section, where the E_b/N_0 is varying. The test were carried out running Matlab R2019 using a dual Intel Xeon CPU E5-2697 processor with the Ubuntu operating system.

It is important to underline that the proposed methods provide the optimal ML performance as long as the detection schemes used for computing the ML distance and the counter-hypothesis distances are ML. The Bit Error Rate (BER) performance of two ML soft-output detection algorithms without clipping is the same, which also happens when comparing two soft-output ML detectors with identical value of the clipping parameter. This has been verified by simulation, as illustrated by Fig. 5. Thus, any soft-output ML algorithm, as well as those proposed in this paper, provide the performance curves shown in Fig. 5.

For this reason, the main task of this section is to comparatively evaluate the complexity of the different algorithms. As mentioned above, the aim of the paper is to present different MIMO-GSM ML algorithms that provide an upper bound on the attainable detection accuracy with much lower computational complexity than exhaustive detection or other ML algorithms already proposed. In order to compare our proposals in terms of computational cost (since in terms of BER any ML algorithm gives the same result), the STS algorithm applied to the MIMO-GSM problem has been selected, since it is one of the most efficient algorithms to provide soft-output ML results. Although some suboptimal algorithms as [21] obtain near-optimal performance with low complexity for particular antenna and constellation values, none of them achieves

Table 1
Main features of different algorithms for ML detection.

Algorithm	GSM	Soft-Output	No clipping	Clipping	Performance	Ref.
STS	No	Yes	Yes	Yes	max-log ML	[27]
DTS	No	Yes	No	Yes	max-log ML	[24]
SA-SSDA	Yes	No	-	-	ML	[14]
BOHD	Yes	No	-	-	ML	[16]
OSTS	Yes	Yes	Yes	Yes	max-log ML	proposed
ODTS	Yes	Yes	No	Yes	clipped max-log ML	proposed
BO-DTS	Yes	Yes	No	Yes	clipped max-log ML	proposed

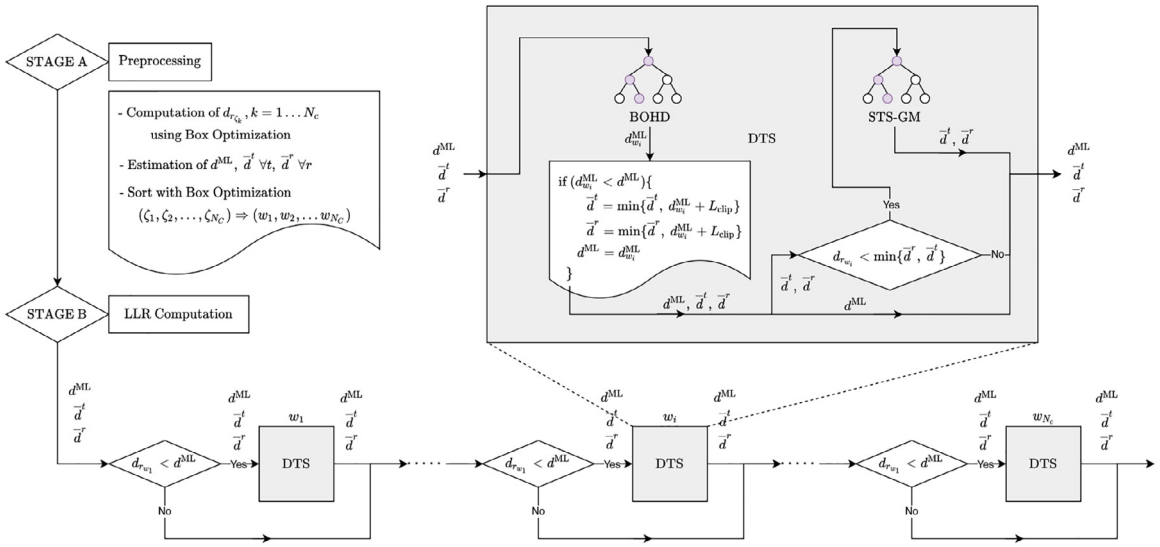

Fig. 4. Illustration of the procedure of the BO-DTS algorithm.

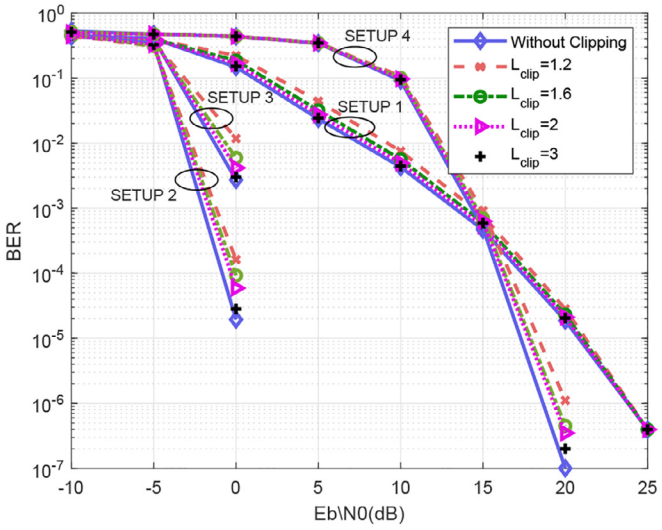

Fig. 5. BER performance comparison for the different setups with different values of clipping.

the soft-output ML benchmark. Thus, our proposals have not been compared with suboptimal algorithms in terms of complexity.

The computational complexity of MIMO tree search detectors is usually measured through the number of expanded nodes. However, since each algorithm has a different cost in the expansion of every node, this method cannot be the only parameter to compare the computational cost of different methods. Thus, the number of flops, another commonly used metric, has also been used for our proposals. We have also recorded the computing times since the final goal is to provide methods that can be executed faster. Nevertheless, it important to note that the computing times eventually depend on the computing platform and on the implementation program. In our case, we have also recorded the number of comparisons that each algorithm executes since rankings that are only based on the other parameters procedure misleading results because the STS algorithm involves many comparisons in the expansion of each node. Therefore, in order to have a comprehensive comparative evaluation of all of the algorithms, we have taken into account all of the parameters mentioned: the average number of flops, times, expanded nodes, and comparisons.

5.1. Results without clipping

In the case without clipping, we compare the proposed OSTS with the original STS algorithm. The STS algorithm has been applied to the MIMO-GSM problem in two ways. First, an STS algorithm is run to solve each subproblem of (10). After completing all of the subproblems, the detector takes the shortest distance as ML solution (this procedure has been denoted as STS). Second, the original STS is applied to solve (4) with the use of an extended constellation, that is, using ″0″ for the detection of inactive antennas. This way of applying STS algorithm to the MIMO-GSM problem is denoted as STS_EC (STS with extended constellation). It is important to note that the STS_EC cannot be used for Setup 4 since it is not possible to obtain the required triangular factorization of the channel matrix.

In terms of performance, as mentioned above, the three ML algorithms provide the same BER curves. In terms of complexity, the different complexity parameters were recorded for the different setups and the results are shown in Fig. 6. The results show that applying the STS algorithm to each subproblem and comparing the final results to obtain the ML solution reduces the complexity when compared to the algorithm using an extended constellation. Furthermore, if we apply the proposed strategies that give rise to the OSTS algorithm, all of the evaluated parameters are considerably reduced. This reduction in cost is independent of the system size, although it is considerably higher when the size of the problem increases, as is the case of Setup 4. The results for this setup are shown in Fig. 6 which show how the different parameters are reduced by 5 to 10 times by the OSTS algorithm compared to the STS method. This reduction depends on the working Eb/N0. Moreover, the reduction in the number of flops is particularly significant, as shown Fig. 6(b).

5.2. Results with clipping

In the case with clipping, we compare the OSTS (with clipping) with the ODTS and BO-DTS algorithms. The comparison with the STS algorithm (with clipping) has been omitted here since it was proved in the previous subsection than the OSTS algorithm provides a much lower computational cost than the STS algorithm. The experiments were performed with different values of the clipping parameter. As mentioned above, the BER performance for the same clipping value is the same when an ML algorithm is applied.

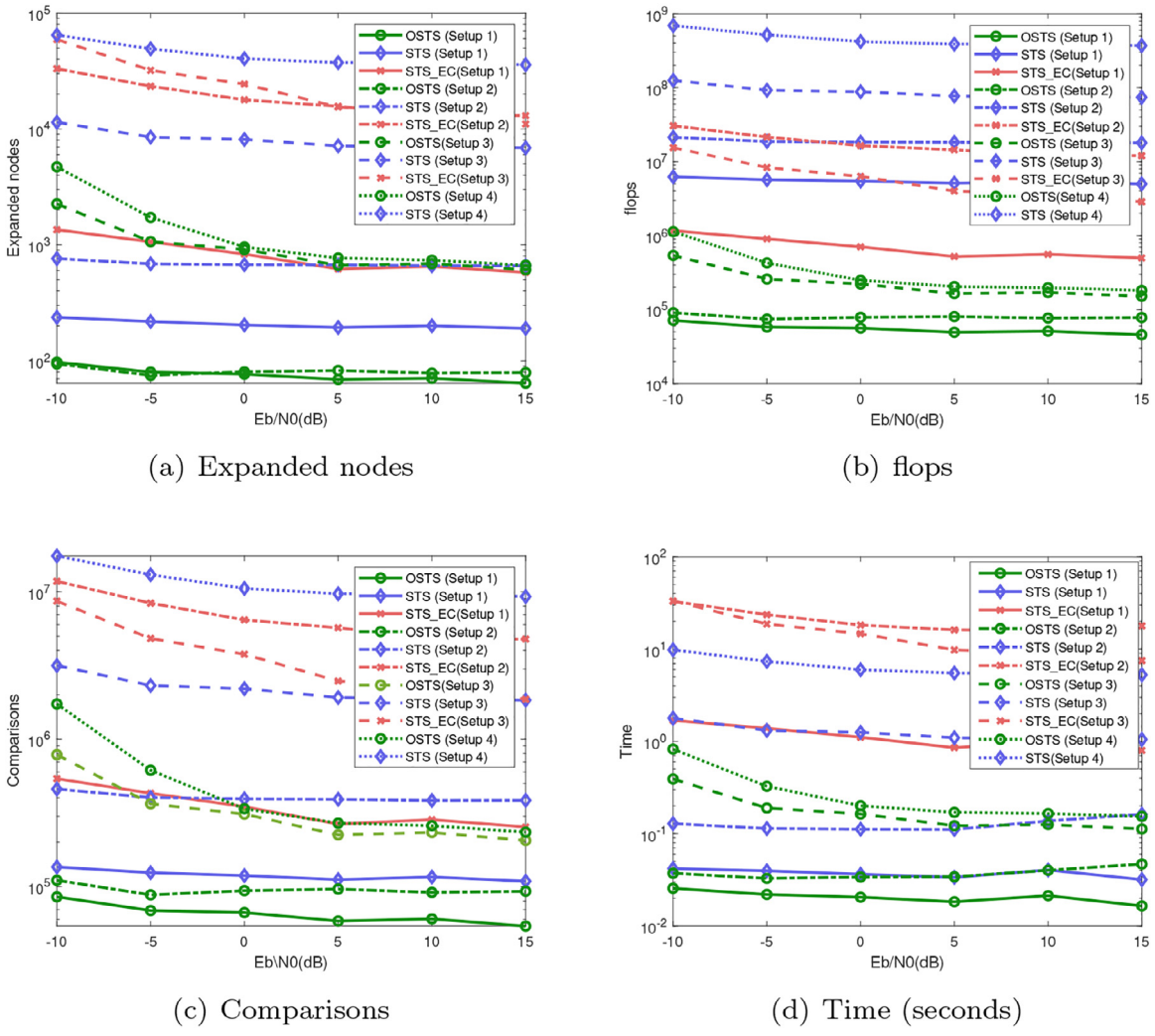

Fig. 6. Average number of expanded nodes, flops, comparisons and computing times (seconds) different setups.

Table 3
Average time (seconds), expanded nodes, flops, and comparisons of the OSTS, ODTS, BO-DTS methods for the soft-output detection for Setup 1.

		Eb/N0(dB)	-10	-5	0	5	10	15
colrule $L_{clip=1.2}$	Time	OSTS	1.9e-2	1.7e-2	1.4e-2	1.5e-2	1.0e-2	1.0e-2
		ODTS	7.7e-3	6.1e-3	4.9e-3	5.3e-3	4.0e-3	4.0e-3
		BO-DTS	5.0e-3	5.0e-3	4.6e-3	5.1e-3	3.9e-3	3.9e-3
	Nodes	OSTS	4.0e1	3.5e1	3.4e1	3.2e1	3.4e1	3.3e1
		ODTS	8.2e1	1.0e2	1.3e2	1.4e2	1.7e2	1.8e2
		BO-DTS	5.1e1	8.9e1	1.2e2	1.4e2	1.7e2	1.8e2
	Flops	OSTS	3.1e4	2.5e4	2.40e4	2.2e4	2.4e4	2.3e4
		ODTS	5.9e3	4.9e3	4.6e3	4.5e3	4.5e3	4.2e3
		BO-DTS	2.8e7	1.9e7	1.4e7	9.2e6	4.5e6	4e4
	Comps	OSTS	3.6e4	2.9e4	2.7e4	2.5e4	2.7e4	2.6e4
		ODTS	1.3e3	1.2e3	1.3e3	1.4e3	1.5e3	1.5e3
		BO-DTS	9.3e2	1.1e3	1.3e3	1.4e3	1.6e3	1.6e3
$L_{clip=3}$	Time	OSTS	1.5e-2	1.4e-2	1.4e-2	1.4e-2	1.4e-2	1.4e-2
		ODTS	6.5e-3	6.2e-3	6.7e-3	6.8e-3	6.9e-3	6.9e-3
		BO-DTS	4.6e-3	5.4e-3	6.4e-3	6.7e-3	6.8e-3	6.8e-3
	Nodes	OSTS	5.7e1	5.6e1	5.6e1	5.7e1	5.6e1	5.6e1
		ODTS	2.2e2	4.6e2	7.4e2	8.5e2	8.6e2	8.7e2
		BO-DTS	1.7e2	4.3e2	7.3e2	8.4e2	8.5e2	8.6e2
	Flops	OSTS	4.2e4	4.0e4	4.0e4	4.1e4	4.0e4	4.0e4
		ODTS	5.8e3	4.9e3	4.6e3	4.6e3	4.5e3	4.2e3
		BO-DTS	2.8e8	2.0e8	1.4e8	9.1e7	4.5e7	3.5e3
	Comps	OSTS	4.9e4	4.7e4	4.8e4	4.8e4	4.7e4	4.7e4
		ODTS	2.1e3	3.5e3	5.2e3	5.8e3	5.9e3	5.9e3
		BO-DTS	1.7e3	3.3e3	5.1e3	5.8e3	5.9e3	5.9e3

Table 4
Average time (seconds), expanded nodes, flops, and comparisons of the OSTS, ODTS, BO-DTS methods for the soft-output detection for Setup 2.

		Eb/N0(dB)	-10	-5	0	5	10	15
$L_{clip=1.2}$	Time	OSTS	2.1e-2	2.0e-2	2.0e-2	2.0e-2	1.9e-2	1.9e-2
		ODTS	1.3e-2	1.2e-2	1.2e-2	1.2e-2	1.1e-2	1.1e-2
		BO-DTS	1.1e-2	1.1e-2	1.1e-2	1.1e-2	1.1e-2	1.1e-2
	Nodes	OSTS	2.1e1	1.7e1	1.8e1	1.8e1	1.6e1	1.6e1
		ODTS	3.7e1	3.6e1	4.0e1	4.2e1	3.5e1	3.4e1
		BO-DTS	2.3e1	3.2e1	3.9e1	4.1e1	3.5e1	3.3e1
	Flops	OSTS	3.3e4	2.9e4	2.9e4	2.9e4	2.8e4	2.7e4
		ODTS	1.9e4	1.9e4	1.9e4	1.9e4	1.8e4	1.8e4
		BO-DTS	6.3e7	4.5e7	3.2e7	2.1e7	1.0e7	1.7e4
	Comps	OSTS	3.2e4	2.7e4	2.7e4	2.7e4	2.5e4	2.4e4
		ODTS	1.1e3	1.0e3	1.1e3	1.1e3	1.0e3	9.6e2
		BO-DTS	1.0e3	1.1e3	1.1e3	1.1e3	1.0e3	9.9e2
$L_{clip=3}$	Time	OSTS	2.85e-2	3.1e-2	3.3e-2	3.2e-2	3.1e-2	3.1e-2
		ODTS	1.5e-2	1.6e-2	1.6e-2	1.6e-2	1.5e-2	1.5e-2
		BO-DTS	1.2e-2	1.4e-2	1.5e-2	1.5e-2	1.5e-2	1.5e-2
	Nodes	OSTS	3.4e1	3.4e1	3.7e1	3.8e1	3.5e1	3.5e1
		ODTS	9.e1	1.7e2	2.2e2	2.3e2	2.0e2	2.0e2
		BO-DTS	7.4e1	1.6e2	2.2e2	2.3e2	2.0e2	2.0e2
	Flops	OSTS	4.2e4	4.1e4	4.3e4	4.3e4	4.1e4	4.2e4
		ODTS	1.9e4	1.9e4	1.9e4	1.9e4	1.8e4	1.8e4
		BO-DTS	6.3e7	4.5e7	3.2e7	2.1e7	1.0e7	1.7e4
	Comps	OSTS	4.4e4	4.3e4	4.6e4	4.6e4	4.3e4	4.4e4
		ODTS	1.5e3	2.1e3	2.6e3	2.7e3	2.3e3	2.4e3
		BO-DTS	1.5e3	2.2e3	2.6e3	2.7e3	2.4e3	2.4e3

Table 5
Average time (seconds), expanded nodes, flops, and comparisons of the OSTS, ODTS, BO-DTS methods for the soft-output detection for Setup 3.

		Eb/N0(dB)	-10	-5	0	5	10	15
$L_{clip=1.2}$	Time	OSTS	7.8e-2	3.0e-2	2.3e-2	1.7e-2	1.5e-2	1.4e-2
		ODTS	2.4e-2	1.6e-2	1.4e-2	1.3e-2	1.2e-2	1.2e-2
		BO-DTS	1.3e-2	1.2e-2	1.2e-2	1.1e-2	1.0e-2	1.0e-2
	Nodes	OSTS	5.2e2	1.4e2	8.4e1	3.7e1	1.6e1	8.1593
		ODTS	3.7e2	1.2e2	7.9e1	4.0e1	1.9e1	1.2e1
		BO-DTS	1.1e2	7.1e1	6.6e1	3.7e1	1.8e1	1.2e1
	Flops	OSTS	1.3e5	4.5e4	3.0e4	2.0e4	1.5e4	1.4e4
		ODTS	3.6e4	2.2e4	1.8e4	1.5e4	1.3e4	1.3e4
		BO-DTS	6.9e8	5.3e8	3.9e8	2.6e8	1.2e8	1e4
	Comps	OSTS	1.7e5	4.8e4	2.7e4	1.3e4	7.6e3	5.3e3
		ODTS	6.7e3	2.6e3	1.8e3	1.1e3	8.2e2	7.3e2
		BO-DTS	3.2e3	2.3e3	2.1e3	1.5e3	1.2e3	1.2e3
$L_{clip=3}$	Time	OSTS	8.2e-2	3.8e-2	3.2e-2	2.4e-2	2.09e-2	1.86e-2
		ODTS	2.4e-2	1.6e-2	1.5e-2	1.3e-2	1.2e-2	1.2e-2
		BO-DTS	1.3e-2	1.2e-2	1.3e-2	1.1e-2	1.1e-2	1.0e-2
	Nodes	OSTS	6.4e2	2.4e2	1.9e2	1.1e2	7.8e1	5.6e1
		ODTS	4.8e2	2.4e2	2.4e2	1.5e2	1.0e2	7.2e1
		BO-DTS	1.9e2	1.7e2	2.1e2	1.4e2	1.0e2	7.1e1
	Flops	OSTS	1.6e5	6.701e4	5.4e4	3.6e4	2.8e4	2.3e4
		ODTS	3.6e4	2.2e4	1.8e4	1.5e4	1.3e4	1.3e4
		BO-DTS	6.9e8	5.3e8	3.9e8	2.6e8	1.2e8	1e4
	Comps	OSTS	2.1e5	0.7e5	6.1e4	3.6e4	2.5e4	1.8e4
		ODTS	7.8e3	3.8e3	3.3e3	2.2e3	1.6e3	1.3e3
		BO-DTS	3.9e3	3.3e3	3.5e3	2.6e3	2.0e3	1.7e3

It is important to recall that high clipping values improve performance in terms of BER, but increase the computational cost of the algorithms. Figure 5 shows the results for the different setups. Moreover, this figure shows that when the clipping value is 3 (or more), the performance degradation is almost negligible.

The complexity results are summarised in Tables 3, 4, 5, and 6, where the average times, expanded nodes, flops, and comparisons are presented for the different setups of Table 2 and clipping values of 1.2 and 3. It is important to highlight that, in most of the evaluated cases, the number of expanded nodes by the OSTS algorithm is lower than the expanded nodes by the ODTS and BO-DTS algorithms. However, the number of comparisons performed by the OSTS algorithm is very significant compared with those performed by the other two algorithms. Furthermore, the computation times is considerably reduced by the ODTS and BO-DTS. Clearly, it is evident that the computation time per node of the OSTS algorithm is much higher than that of the ODTS and BO-DTS algorithms. Therefore, even though the ODTS and BO-DTS expand more nodes, they are more efficient than the OSTS algorithm.Thus, we can conclude that, when clipping values are applied, the ODTS and BO-DTS algorithms are more efficient in terms of complexity than the OSTS, regardless of the size of the system.

The BO-DTS algorithm requires a previous run of the BO algorithm in the preprocesing step. This means an increase in computational cost, which is clearly reflected in the number of flops executed by the algorithm. However, when the problem size increases (Setup 3 and 4), the increase in computational cost at the preprocessing stage (which is reflected in the number of flops) is compensated by the complexity reduction in the second stage because many configurations can be pruned. Therefore, even though the total number of flops is high, the number of expanded nodes, time, and comparisons is considerably reduced, as Table 6 shows.

Table 6
Average time (seconds), expanded nodes, flops, and comparisons of the OSTS, ODTS, BO-DTS methods for the soft-output detection for Setup 4.

		Eb/N0(dB)	-10	-5	0	5	10	15
$L_{clip=1.2}$	Time	OSTS	7.5e-1	2.9e-1	1.3e-1	1.1e-1	1.1e-1	1.0e-1
		ODTS	1.9e-1	1.3e-1	8.4e-2	7.9e-2	7.8e-2	6.4e-2
		BO-DTS	6.7e-2	7.0e-2	5.4e-2	5.6e-2	5.9e-2	5.1e-2
	Nodes	OSTS	3.8e3	1.2e3	7.5e2	6.1e2	5.3e2	4.8e2
		ODTS	2.6e3	1.5e3	1.7e3	1.9e3	1.8e3	1.5e3
		BO-DTS	3.6e2	6.2e2	1.0e3	1.3e3	1.3e3	1.1e3
	Flops	OSTS	9.2e5	3.2e5	2.0e5	1.6e5	1.5e5	1.3e5
		ODTS	2.4e5	1.3e5	1.0e5	9.1e4	7.5e4	5.1e4
		BO-DTS	3.4e9	2.7e9	2.0e9	1.3e9	6.7e8	2.71e4
	Comps	OSTS	1.4e6	4.5e5	2.6e5	2.1e5	1.8e5	1.6e5
		ODTS	4.9e4	2.7e4	2.6e4	2.6e4	2.4e4	1.9e4
		BO-DTS	1.2e4	1.4e4	1.8e4	2.0e4	2.0e4	1.7e4
$L_{clip=3}$	Time	OSTS	5.2e-1	2.1e-1	1.4e-1	1.2e-1	1.2e-1	1.1e-1
		ODTS	1.5e-1	1.2e-1	1.4e-1	1.5e-1	1.4e-1	1.7e-1
		BO-DTS	5.6e-2	7.0e-2	9.1e-2	1.0e-1	1.0e-1	9.97e-2
	Nodes	OSTS	4.2e3	1.5e3	8.7e2	7.1e2	6.5e2	6.2e2
		ODTS	4.4e3	6.2e3	1.0e4	1.2e4	1.3e4	1.2e4
		BO-DTS	1.2e3	3.4e3	6.8e3	8.7e3	9.3e3	9.0e3
	Flops	OSTS	1.0e6	3.8e5	2.2e5	1.9e5	1.7e5	1.7e5
		ODTS	2.4e5	1.3e5	1.0e5	9.1e4	7.5e4	5.1e4
		BO-DTS	3.4e9	2.7e9	2.0e9	1.3e9	6.7e8	2.7e4
	Comps	OSTS	1.5e6	5.4e5	3.0e5	2.5e5	2.2e5	2.1e5
		ODTS	6.8e4	7.4e4	1.1e5	1.3e5	1.40e5	1.3e5
		BO-DTS	2.0e4	4.2e4	7.6e4	9.4e4	1.0e5	9.6e4

Thus, in general terms, it is more efficient to use the BO-DTS algorithm for large systems and the ODTS algorithm is recommended for small systems.

6. Conclusions

In this work, three new algorithms for soft-output MIMO-GSM detection have been presented. One of them, the OSTS, can be applied with and without clipping, and the other two can only be used with clipping. The algorithms were tested in four setups with different system sizes. The results have provided some very clear conclusions. In the case without clipping, the proposed algorithm significantly reduces the computational simulation cost compared to other ML algorithms, which makes it relevant for use in these cases. From a practical point of view, the clipped option is more relevant. In the case with clipping, other options have been proposed (the ODTS and the BO-DTS) which have been found to be more efficient than the OSTS algorithm. Depending on the overall size of the system, it is more convenient to select either the ODTS or the BO-DTS. The ODTS provides a lower computational cost for small sizes. However, when the system size increases, the BO-DTS algorithm provides a cost that is considerably lower.

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

M. Ángeles Simarro: Conceptualization, Software, Investigation, Writing – original draft. **Víctor M. García-Mollá:** Conceptualization, Writing – review & editing. **F.J. Martínez-Zaldívar:** Writing – review & editing. **Alberto Gonzalez:** Writing – review & editing.

Acknowledgements

This work has been partially supported by Spanish Ministry of Science, Innovation and Universities and by European Union through grant RTI2018-098085-BC41 (MCUI/AEI/FEDER), by GVA through PROMETEO/2019/109 and by Spanish Ministry of Science, Innovation and Universities through grant RED2018-102668-T.

References

[1] D. Tse, P. Viswanath, Fundamentals of Wireless Communication, Cambridge University Press, 2005.
[2] E.G. Larsson, O. Edfors, F. Tufvesson, T.L. Marzetta, Massive MIMO for next generation wireless systems, IEEE Commun. Mag. 52 (2) (2014) 186–195.
[3] M. Wen, B. Zheng, K.J. Kim, M. Di Renzo, T.A. Tsiftsis, K.C. Chen, N. Al-Dhahir, A survey on spatial modulation in emerging wireless systems: research progresses and applications, IEEE J. Sel. Areas Commun. 37 (9) (2019) 1949–1972.
[4] R.Y. Mesleh, H. Haas, S. Sinanovic, C.W. Ahn, S. Yun, Spatial modulation, IEEE Trans. Veh. Technol. 57 (4) (2008) 2228–2241.
[5] L.L. Yang, Transmitter preprocessing aided spatial modulation for multiple-input multiple-output systems, in: 2011 IEEE 73rd Vehicular Technology Conference (VTC Spring), IEEE, 2011, pp. 1–5.
[6] R. Mesleh, S. Ikki, H.M. Aggoune, Quadrature spatial modulation, IEEE Trans. Veh. Technol. 64 (6) (2014) 2738–2742.
[7] J. Li, M. Wen, X. Cheng, Y. Yan, S. Song, M.H. Lee, Generalized precoding-aided quadrature spatial modulation, IEEE Trans. Veh. Technol. 66 (2) (2016) 1881–1886.
[8] J. Li, S. Dang, Y. Yan, Y. Peng, S. Al-Rubaye, A. Tsourdos, Generalized quadrature spatial modulation and its application to vehicular networks with NOMA, IEEE Trans. Intell. Transp. Syst. 22 (7) (2021) 4030–4039.
[9] I. Al-Nahhal, E. Basar, O.A. Dobre, S. Ikki, Optimum low-complexity decoder for spatial modulation, IEEE J. Sel. Areas Commun. 37 (9) (2019) 2001–2013.
[10] T.L. Narasimhan, P. Raviteja, A. Chockalingam, Generalized spatial modulation in large-scale multiuser MIMO systems, IEEE Trans. Wirel. Commun. 14 (7) (2015) 3764–3779.
[11] M. Wen, X. Cheng, L. Yang, Index Modulation for 5G Wireless Communications, Springer, 2017.
[12] M. Di Renzo, H. Haas, A. Ghrayeb, S. Sugiura, L. Hanzo, Spatial modulation for generalized MIMO: challenges, opportunities, and implementation, Proc. IEEE 102 (1) (2013) 56–103.
[13] Y. Wu, H. Ying, X.Q. Jiang, H. Hai, A joint data mapping and detection for high performance generalized spatial modulation, IEEE Commun. Lett. 23 (11) (2019) 2008–2011.
[14] T.H. Liu, C.E. Chen, C.H. Liu, Fast maximum likelihood detection of the generalized spatially modulated signals using successive sphere decoding algorithms, IEEE Commun. Lett. 23 (4) (2019) 656–659.
[15] S. Fan, Y. Xiao, P. Yang, L. Dan, W. Xiang, Approximate message passing detector based upon probability sorting for large-scale GSM systems, IEEE Trans. Veh. Technol. 68 (9) (2019) 9303–9307.
[16] V.M. Garcia-Molla, F.J. Martínez-Zaldívar, M.A. Simarro, A. Gonzalez, Maximum likelihood low-complexity GSM detection for large MIMO systems, Signal Process. (2020) 107661.
[17] H. Albinsaid, K. Singh, S. Biswas, C.P. Li, M.S. Alouini, Block deep neural network-based signal detector for generalized spatial modulation, IEEE Commun. Lett. (2020).

[18] B. Zheng, M. Wen, F. Chen, N. Huang, F. Ji, H. Yu, The K-best sphere decoding for soft detection of generalized spatial modulation, IEEE Trans. Commun. 65 (11) (2017) 4803–4816.
[19] S. Lin, B. Zheng, F. Chen, F. Ji, H. Yu, Soft demodulators based on deterministic SMC for single-carrier GSM in broadband channels, IEEE J. Sel. Areas Commun. 37 (9) (2019) 1973–1985.
[20] X.Q. Jiang, Y. Zheng, W. Chen, M. Wen, J. Li, Two-layer LDPC codes for low complexity ML detection in GSM systems, IEEE Wirel. Commun. Lett. 7 (3) (2017) 408–411.
[21] L. Xiao, P. Yang, Y. Xiao, J. Liu, S. Fan, B. Dong, S. Li, An improved soft-input soft-output detector for generalized spatial modulation, IEEE Signal Process. Lett. 23 (1) (2015) 30–34.
[22] M.S. Yee, Max-log-MAP sphere decoder, in: Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005., vol. 3, IEEE, 2005, pp. 1013–1016.
[23] C. Studer, A. Burg, H. Bolcskei, Soft-output sphere decoding: algorithms and VLSI implementation, IEEE J. Sel. Areas Commun. 26 (2) (2008) 290–300.
[24] V.M. Garcia-Molla, M.A. Simarro, F.-J. Martínez-Zaldívar, A. González, A.M. Vidal, Maximum likelihood soft-output detection through sphere decoding combined with box optimization, Signal Process. 125 (2016) 249–260.
[25] V.M. Garcia-Molla, A.M. Vidal, A. Gonzalez, S. Roger, Improved maximum likelihood detection through sphere decoding combined with box optimization, Signal Process. 98 (2014) 284–294.
[26] H. Wymeersch, Iterative Receiver Design, vol. 234, Cambridge University Press Cambridge, 2007.
[27] C. Studer, H. Bölcskei, Soft–input soft-output single tree-search sphere decoding, IEEE Trans. Inf. Theory 56 (10) (2010) 4827–4842.
[28] M. Stojnic, H. Vikalo, B. Hassibi, Speeding up the sphere decoder with H^{∞} and SDP inspired lower bounds, IEEE Trans. Signal Process. 56 (2) (2008) 712–726.
[29] G.T. 38.212, NR; Multiplexing and Channel Coding, 3rd Generation Partnership Project; Technical Specification Group Radio Access Network, 2018.