



**HAL**  
open science

# When Should We Use Geometrical-Based MIMO Detection Instead of Tree-Based Techniques? A Pareto Analysis

Bastien Trotobas, Adrien Llave, Amor Nafkha, Yves Louët

► **To cite this version:**

Bastien Trotobas, Adrien Llave, Amor Nafkha, Yves Louët. When Should We Use Geometrical-Based MIMO Detection Instead of Tree-Based Techniques? A Pareto Analysis. 2020. hal-02963699v1

**HAL Id: hal-02963699**

**<https://hal.science/hal-02963699v1>**

Preprint submitted on 11 Oct 2020 (v1), last revised 4 Nov 2020 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# When Should We Use Geometrical-Based MIMO Detection Instead of Tree-Based Techniques? A Pareto Analysis

Bastien Trotobas<sup>1,2</sup>, Adrien Llave<sup>1,3</sup>, Amor Nafkha<sup>1,2</sup> and Yves Louët<sup>1,2</sup>.

<sup>1</sup>IETR - Institut d'électronique et des technologies du numérique UMR CNRS 6164

<sup>2</sup>SCEE research group - Signal, Communication and Embedded Electronics

<sup>3</sup>FAST research group - Facial Analysis Synthesis and Tracking  
CentraleSupélec, Avenue de la boulaie, 35576 Cesson Sévigné, France

This paper is currently under review for publication in IEEE Access.

## Abstract

The soft-output multiple-input multiple-output (MIMO) detection problem has been extensively studied, and a large number of heuristics and metaheuristics have been proposed to solve it. Unlike classical tree-search based detectors, geometrical heuristic algorithms involved two consecutive steps: **(i)** an exploration step based on the geometry of the channel matrix singular vectors; **(ii)** a local exploitation step is performed in order to obtain better final solution. In this paper, new enhancements for geometrical heuristics are introduced to significantly reduce the complexity in quadrature phase-shift keying (QPSK) and allow 16 quadrature amplitude modulation (QAM) capability through new exploration techniques. The performance-complexity trade-off between the new detector and two tree-based algorithms is investigated through Pareto efficiency. The Pareto framework also allows us to select the most efficient tuning parameters based on an exhaustive search. The proposed detector can be customized on the fly using only one or two parameters to balance the trade-off between computational complexity and bit error rate performances. Moreover, the Pareto fronts demonstrate that the new geometrical heuristic is especially efficient with QPSK since it provides a significant reduction in regards to the computational complexity while preserving good bit error rate (BER) performance and ensuring high flexibility.

# 1 Introduction

In the last decades, the increase in the quantity of data sent over wireless channels has led to a shortage of available frequency bands. This scarcity has driven researchers and operators to improve the spectrum efficiency of wireless communications systems and, more recently, seek new frequency bands with THz technologies. Specifically, spectrum efficiency improvement increases the data throughput and the link quality without using new frequency bands.

In this context, MIMO systems have been widely adopted for their capability to multiplex transmitted data streams over time-frequency-space dimensions. The spatial multiplexing MIMO technique is mainly used to increase the data transmission rate or spectral efficiency. The space-division multiplexing (SDM) MIMO technology can transmit several data streams in the same time-frequency slot and separate them according to spatial considerations. This multiplexing technique increases the spectrum efficiency through the addition of antennas. However, the more antennas there are, the more complex the receiver design is. Therefore, these systems require new algorithms to exploit the spatial information to separate data streams efficiently.

WiFi standards like IEEE 802.11n/ac, long-term evolution (LTE), WiMAX, and 5G, among other modern standards, rely on MIMO technologies. All of these standards use known pilot signals in order to estimate the channel state information (CSI). It is common to assume that the receiver gets a perfect CSI whereas the transmitter is CSI-agnostic. This operating regime is easier to set up as it does not require each CSI to be sent back to each emitting antenna.

The separation of received streams has been widely studied, and many algorithms have already been proposed in the literature. This detection problem is known to be NP-hard [1], which implies that an optimal solution cannot be computed in polynomial time (unless under the unattainable assumption of  $P=NP$ ). Such optimal algorithms include the naive resolution with the maximum likelihood (ML) detector or the optimal tree-path search using sphere decoding (SD) as initiated by [2–4]. However, their exponential complexity does not make them suitable for hardware implementation, especially at low signal-to-noise ratio (SNR) regimes. Still, it should be noted that some SD implementations can compete with a polynomial algorithm in terms of complexity when the number of antennas and the constellation size are rather small and when the SNR is high enough [5].

Several approaches have been considered to offer heuristics and metaheuristics that provide good performance in polynomial time. Under ideal circumstances where the amount of

information available is significant, linear detectors provide an acceptable result. In more difficult cases, more advanced algorithms are necessary. The earliest heuristics are based on the interference cancellation between the different received data streams. Two versions of this heuristic coexist. The successive interference cancellation (SIC) detection scheme suppress the interference by iteratively maximizing the signal-to-interference-plus-noise ratio (SNR). This approach is well suited when the received signals present different individual quality metric corresponding to each of the received data streams [6, 7]. In the opposite scenario, when data streams have similar quality parallel interference cancellation (PIC) detectors are preferred [8, 9].

The detection problem turns out to be a complicated combinatorial optimization problem. Thus, to circumvent and simply solve this problem, tree-based heuristics algorithms can be used. These algorithms are classified according to the method of searching the tree: depth-first algorithms look for the best possible leaf through a descent, prune and backtrack process; breadth-first algorithms keep only a fixed number of paths at each step [10–13]; and best-first algorithms exploiting metrics to determine how to explore the tree [14, 15].

Alternative solutions have been proposed thanks to a shift in the problem perspective. For instance, detectors based on Markov chain Monte Carlo (MCMC) algorithms have been developed by addressing the problem through a probabilistic approach [16, 17]. The emergence of deep networks trained to tackle the detection problem is also studied in recent work [18]. Finally, bio-inspired metaheuristics based on ant colony optimization (ACO) [19] or on the firefly algorithm (FA) [20, 21] have also been proposed.

We previously investigated the interest of geometrical heuristics to solve the MIMO detection problem [22]. We compared the geometrical approach and the tree-based one for a bit interleaved coded modulation (BICM) scheme with QPSK modulation. The criteria of BER and complexity revealed that the geometrical method was close but not yet as good as the state-of-the-art detectors.

The geometrical approach provided in [22] was restricted to lower-order constellations (i.e. binary phase-shift keying (QPSK) and QPSK) whereas higher-order modulations were not investigated at all. Moreover, the geometrical heuristic was compared with a tree-based reference on both BER performance and complexity but these two criteria were not considered simultaneously within a trade-off perspective. This paper presents two new contributions that help to overcome the highlighted limitations present in [22]:

- Enhancements to improve geometric heuristics on QPSK and extend its use cases to high-order modulations (e.g.,

16-QAM)

- Performance-complexity trade-off study using the Pareto efficiency.

The paper is organized as follows. In Section 2, the MIMO system model under consideration is presented. In Section 3, a detailed description of the soft-output computation is given, while in Section 4, we provide the geometrical-based MIMO detection framework. The proposed enhanced geometrical-based detection algorithm is presented in Section 5. A detailed performance-complexity tradeoff analysis of the proposed algorithm through Pareto-front curves is given in Section 6 for both QPSK and 16-QAM. Section 7 concludes the paper.

## Notations

In the following, bold uppercases (*resp.* lowercases) denote matrices (*resp.* vectors), whereas the other letters refer to scalars. All sets are noted with calligraphic uppercases, and the corresponding lowercase refers to their cardinality. The vector  $\mathbf{h}_j$  denotes the  $j^{\text{th}}$  column of matrix  $\mathbf{H}$  and  $\mathbf{H}^T$ , denotes the transpose  $\mathbf{H}$ . The natural logarithm is denoted by  $\ln$ . The real and imaginary part of a complex number  $a \in \mathbb{C}$  are denoted by  $\Re(a)$  and  $\Im(a)$ , respectively. We denote the  $l_2$ -norm of vector  $\mathbf{y}$  as  $\|\mathbf{y}\|$ .

## 2 Transmission Model

Let the  $N \times N$  Rayleigh-fading MIMO system transmitting  $N$  data streams to  $N$  receiving antennas. We assume a perfect CSI at the receiver (ideal channel estimator), whereas the transceiver is CSI-agnostic. From a model perspective, the receiver knows the channel matrix  $\mathbf{H}_c \in \mathbb{C}^{N \times N}$  where  $h_c(i, j)$  is the complex channel gain from antenna  $j$  to antenna  $i$ . The MIMO channel is modeled as quasi-static block fading where the channel path gains can be considered constant during a large block comprising hundreds of transmitted vectors [23]. The channel gains change according to a statistical model given by an independent Rayleigh-distributed envelope. Therefore, the computational complexity of any pre-processing phases, such as singular value decomposition (SVD) or QR decomposition, pseudo-inverse calculation, can be negligible. Indeed, they are performed once for several transmitted vectors.

Let  $\mathcal{Q}_c$  be the set of all constellation symbols from a square QAM. We denote by  $\mathbf{y}_c \in \mathbb{C}^N$  the signals received on each antenna after the propagation of the symbols  $\mathbf{x}_c \in \mathcal{Q}_c^N$  through

the channel and after the addition of the complex gaussian noise  $\mathbf{w}_c \sim \mathcal{CN}(0, \sigma^2)$ . With this notations, the system model is expressed as

$$\mathbf{y}_c = \mathbf{H}_c \mathbf{x}_c + \mathbf{w}_c. \quad (1)$$

This model can be rewritten as an equivalent real-valued expression such that

$$\mathbf{y} \triangleq \begin{bmatrix} \Re(\mathbf{y}_c) \\ \Im(\mathbf{y}_c) \end{bmatrix} \quad (2)$$

$$\mathbf{H} \triangleq \begin{bmatrix} \Re(\mathbf{H}_c) & -\Im(\mathbf{H}_c) \\ \Im(\mathbf{H}_c) & \Re(\mathbf{H}_c) \end{bmatrix} \quad (3)$$

$$\mathbf{x} \triangleq \begin{bmatrix} \Re(\mathbf{x}_c) \\ \Im(\mathbf{x}_c) \end{bmatrix} \quad (4)$$

$$\mathbf{w} \triangleq \begin{bmatrix} \Re(\mathbf{w}_c) \\ \Im(\mathbf{w}_c) \end{bmatrix} \quad (5)$$

$$\mathcal{Q} \triangleq \Re(\mathcal{Q}_c) \quad (6)$$

$$n \triangleq 2N \quad (7)$$

where the new parameter  $n$  defines the size of the real-valued matrices and vectors. It can be interpreted as the number of real-valued data streams. Indeed, switching from a complex-valued to a real-valued perspective is equivalent to process real and imaginary parts independently. In this real-valued model, (1) becomes the following equivalent system model:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}. \quad (8)$$

The BICM transceiver is composed of an encoder, an assumed-perfect interleaver, and a modulator. The receiver uses the corresponding components in a reversed order: demodulator, deinterleaver, and then decoder.

## 3 Soft-output Computations

Let  $b_{ij}$  be the  $i^{\text{th}}$  bit encoded in the  $j^{\text{th}}$  symbol of  $\mathbf{x}$ . Basic hard-output detectors provide an estimate of the vector of transmitted symbols. To improve the performance, soft-output detectors search for the log-likelihood ratio (LLR) for each bit defined as

$$L_{ij} = \ln \frac{P(b_{ij} = 1 | (\mathbf{H}, \mathbf{y}))}{P(b_{ij} = 0 | (\mathbf{H}, \mathbf{y}))} \quad (9)$$

with  $P(b_{ij} | (\mathbf{H}, \mathbf{y}))$  the probability mass function of  $b_{ij}$  given the channel state and the received vector.

Expression (9) is not suitable for practical use as it has been shown that its computations is exponentially complex [24]. A common solution is to use the max-log approximation:

$$L_{ij} \approx \frac{1}{2\sigma^2} \left( \min_{\mathbf{x} \in \mathcal{X}_{ij}^0} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 - \min_{\mathbf{x} \in \mathcal{X}_{ij}^1} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \right), \quad (10)$$

where  $\mathcal{X}_{ij}^k = \{\mathbf{x} \in \mathcal{Q}^n : b_{ij} = k\}$  is the set of all symbols with  $b_{ij}$  equals to  $k$  [25–27].

This new expression (10) is still exponentially complex as the norm must be computed for each point in the constellation. Indeed, we clearly have  $\mathcal{X}_{ij}^0 \cup \mathcal{X}_{ij}^1 = \mathcal{Q}^n$  which contains  $2^n$  points. Therefore, we introduce a new subset  $\mathcal{S} \subset \mathcal{Q}^n$  with a lower cardinality and approximate (10) on it. The new expression becomes

$$L_{ij} \approx \frac{1}{2\sigma^2} \left( \min_{\mathbf{x} \in \mathcal{S} \cap \mathcal{X}_{ij}^0} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 - \min_{\mathbf{x} \in \mathcal{S} \cap \mathcal{X}_{ij}^1} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \right). \quad (11)$$

The reduced subset  $\mathcal{S}$  may have no points from  $\mathcal{X}_{ij}^k$  (i.e.,  $\mathcal{S} \cap \mathcal{X}_{ij}^k = \emptyset$ ). In such a configuration, we assign  $k$  to the bit  $b_{i,j}$  by the fact that the probability  $P(b_{i,j} = k | (\mathbf{H}, \mathbf{y}))$  is considered equal to one. Therefore,  $L_{ij}$  is set to its maximum (if  $b_{i,j} = 1$ ) or minimum (if  $b_{i,j} = 0$ ) value to reflect this certainty.

In the remainder of this paper, the objective function denotes the squared norm involved in the LLRs, and a point is considered better than another if it has a smaller objective function.

## 4 Geometrical-Based Detection

Soft-output geometrical heuristics are based on three main steps: exploration, exploitation, and the LLRs computation. This section reviews the first two-step as the last phase has already been discussed in the previous section.

Exploration and exploitation steps are both designed to search for feasible solutions into  $\mathcal{Q}^n$ . They can be viewed as coarse and fine search methods. Coarse search step (exploration) is performed over whole solution set  $\mathcal{Q}^n$  in order to find pertinent solutions, whereas exploitation step (fine search) step refines the quality of the solutions through local searches.

### 4.1 SVD-Based Exploration

The exploration step produces a set  $\mathcal{P}$  of promising points to be exploited in the next step. Let  $\mathbf{H} = \mathbf{U}\mathbf{A}\mathbf{V}$  be the SVD of

$\mathbf{H}$  with  $(\mathbf{U}, \mathbf{V})$  two orthogonal matrices and  $\mathbf{A}$  a diagonal one containing the singular values. Without loss of generality, we assume that the singular values are sorted in ascending order:

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n. \quad (12)$$

Let  $\mathbf{x}^* \in \mathbb{R}^n$  be the real vector minimizing the objective function. This point can be obtained using the Moore-Penrose inverse  $\mathbf{H}^+$  through  $\mathbf{x}^* = \mathbf{H}^+\mathbf{y}$ . The regular inverse can be use if the channel matrix is well conditioned, if not, the Moore-Penrose invert is required. We can rewrite the objective function as

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \|\mathbf{U}\mathbf{A}\mathbf{V}(\mathbf{x}^* - \mathbf{x})\|^2 \quad (13)$$

$$= (\mathbf{V}(\mathbf{x}^* - \mathbf{x}))^T \mathbf{A}\mathbf{U}^T\mathbf{U}\mathbf{A}(\mathbf{V}(\mathbf{x}^* - \mathbf{x})). \quad (14)$$

Given that  $\mathbf{U}$  is orthogonal, the previous equation gives

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = (\mathbf{V}(\mathbf{x}^* - \mathbf{x}))^T \mathbf{A}^2 (\mathbf{V}(\mathbf{x}^* - \mathbf{x})). \quad (15)$$

As  $\mathbf{V}$  is orthogonal, its columns  $\{\mathbf{v}_i\}_{i=1\dots n}$  constitute a basis and we can introduce  $\alpha_i$  the coordinates of  $\mathbf{x}^* - \mathbf{x}$  on this basis. The matrix  $\mathbf{A}$  is diagonal, then the objective function can be expressed as

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \sum_{i=1}^n \alpha_i^2 \lambda_i^2 \quad (16)$$

which provides hints on the objective function evolution. The ascending order of the singular values induces that the first  $\alpha_i$  are less impacting than the last ones. Therefore, points with similar coordinates except for the first ones, can be considered equivalent with respect to their objective function. Exploration step aims for building the promising set  $\mathcal{P}$  from points as different as possible but with equivalently low objective function. The promising set  $\mathcal{P}$  is also used to initialize  $\mathcal{S}$ .

### 4.2 Iterative Local Exploitation

The exploitation step gets the most of the previously selected points  $\mathcal{P}$  by exploring each promising feasible solution belonging the subset  $\mathcal{P}$ . For each promising point in the subset, an iterative local search is perform to find better feasible solutions around the starting points. One iteration can be decomposed into three steps :

1. Generate at most  $2n$  new points equal to the candidate but for one coordinate. This modified coordinate is once the previous and once the next symbol in the real-valued

constellation. If the coordinate to be modified is already at an extreme value (ie. the first or last symbol in the constellation), then a single point is generated.

2. Compute the objective function for each generated feasible solution and add them to  $\mathcal{S}$ .
3. Select the best feasible solution between the initial candidate and the newly generated points. The best point becomes the new starting candidate for the next iteration. If there is a tie, prefer the initial point.

It is easy to prove that this algorithm reaches a stable point. First, each iteration decreases the objective function over the finite set  $\mathcal{Q}^n$  of possibilities. Moreover, oscillations between two equally good points are prevented by the tie rule. Therefore, this process ends on a stable point that is not guaranteed to be the global optimum. The performance-complexity trade-off can be tuned by stopping the algorithm after a predefined number of iterations rather than waiting for a stable point.

## 5 Enhancements to Geometrical Detection

In this section, we present some enhancements to the geometrical based detection framework that was presented in the previous section. Section 5.1 describes the previous exploitation technique and highlights its limitations, then Section 5.2 provides the new exploration techniques to build the set of promising feasible solutions. Section 5.3 introduces a lossless method to reduce the complexity and Section 5.4 summarizes the proposed geometrical detector and provides the algorithmic computational costs of the detection process.

### 5.1 Previous Exploration Technique

The SVD-based exploration is the core step of the geometrical heuristic since it provides a good set  $\mathcal{P}$  of promising point is the key to the success of the following steps. In the previous works,  $\mathcal{P}$  was build using an intersection-based process [22, 28].

Fig. 1 illustrates this exploration technique with  $n = 2$  and a 16-QAM modulation scheme. The process starts by computing the line passing through  $\mathbf{x}^*$  and directed by the singular vector  $\mathbf{v}_1$ . The dashed blue line represents this straight line. Then, each intersection of this line with the basis axis is projected on the 16-QAM constellation to build the subset  $\mathcal{P}$ . This process is illustrated by the purple arrows. Fig. 1

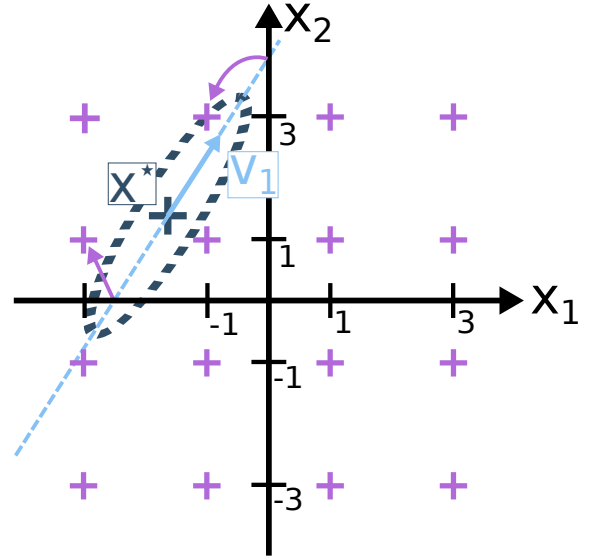


Figure 1: Previous exploration process with  $n = 2$  and a 16-QAM.

shows the process with one direction for readability's sake, but real detectors apply this method for several directions. As an example, we used 3 directions for a 4x4 channel in [22] or 4 directions for a 30x30 channel in [28].

The exploration technique is able to find promising points as the elements of  $\mathcal{P}$  should have similar objective function values but it has two major drawbacks. First, the construction of  $\mathcal{P}$  implies that it contains only points that are near the basis hyperplanes. For instance, the point  $(-3, -3)$  cannot be reached by this exploration process, regardless of its quality, since no intersection could be projected to this point. That is why, this method performs well with QPSK, where all points are accessible but that it leads to poor results with higher-order modulation schemes. Besides, this exploration technique is very complex, with large channels. Indeed, building  $\mathcal{P}$  requires to compute the intersection of several straight lines with the  $n$  basis hyperplanes. Therefore, the more  $n$  growth, the more intersections are needed. That is why the new exploration technique should rely on steps that do not increase in complexity with additional antennas.

### 5.2 Proposed Exploration Techniques

As exposed in the previous section, the exploration technique presented in [22] has two significant drawbacks. In this section, we propose new exploration techniques that solve these issues. Indeed, the new methods are able to select, with no

preferences, each point of the constellation so that it could perform well on higher-order modulation schemes. Moreover, they are composed of steps that are independent of the number of transmitted data streams  $n$ .

### 5.2.1 Intersection-Less (IL) Exploration

As stated in Section 4.1, the exploration builds a subset of promising points  $\mathcal{P} \subset \mathcal{Q}^n$  as different as possible with equivalently low objective function. On the first hand, (16) shows that equivalent points differ only on their first coordinates when expressed on the basis  $\{\mathbf{v}_i : 1 \leq i \leq n\}$ . On the other hand,  $\mathbf{x}^*$  is obviously a good point since it is the global optimum on  $\mathbb{R}^n$ . Therefore, the sought equivalently good points can be constructed by adding to  $\mathbf{x}^*$  some linear combinations of the first  $\mathbf{v}_i$ . Let  $n_d \leq n$  be the number of  $\mathbf{v}_i$  used during this new exploration step.

To have equivalent points, we build them as

$$\mathbf{x}^* + \sum_{i=1}^{n_d} \frac{\pm f}{\lambda_i} \mathbf{v}_i \quad (17)$$

with  $f$  some scalar whose value will be discussed later. Expressed in the  $\mathbf{V}$  basis and after the translation to  $\mathbf{x}^*$ , these points' coordinates are

$$\left( \pm \frac{f}{\lambda_1}, \pm \frac{f}{\lambda_2}, \dots, \pm \frac{f}{\lambda_{n_d}}, 0, \dots, 0 \right)_{(\mathbf{x}^*, \mathbf{V})} \quad (18)$$

Equation (16) highlights that these points have exactly the same objective function being  $f^2 n_d$ . However, these points do not belong to  $\mathcal{Q}^n$  so that they cannot be used directly to build the subset  $\mathcal{P}$ . Therefore, the promising set is built by taking the nearest value in  $\mathcal{Q}$  for each coordinates. This last step is the same used in the zero-forcing (ZF) detector and will be referred as the “projection on a set” in the remainder of this paper.

Fig. 2 and 3 provide examples of the intersection-less exploration process for a 16-QAM,  $f = 1$  and  $n = 2$ . The exploration process starts at its center  $\mathbf{x}^*$ . Blue arrows represent the addition of the scaled singular vectors that appear in (17) and purple ones correspond to the projection on the constellation.

Fig. 2 illustrates the case  $n_d = 1$  where two different points are obtained:  $(-3, -1)$  and  $(-1, 3)$ . Fig. 3 shows the same scenario with  $n_d = 2$ . In this situation, three points are generated:  $(-3, -1)$ ,  $(-3, 1)$  and  $(-1, 3)$ .

The points from (17) can be either within or outside the constellation regarding the magnitude of each singular value. That is why the projection can produce either good or bad

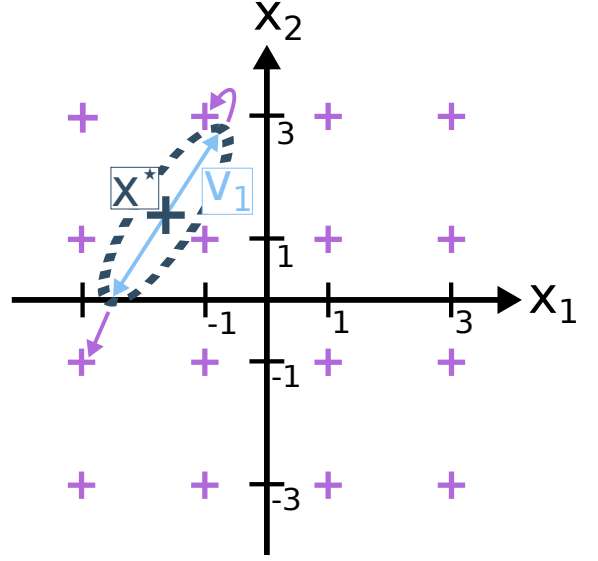


Figure 2: Well-conditioned intersection-less exploration with  $n = 2$ ,  $n_d = 1$  and a 16-QAM.

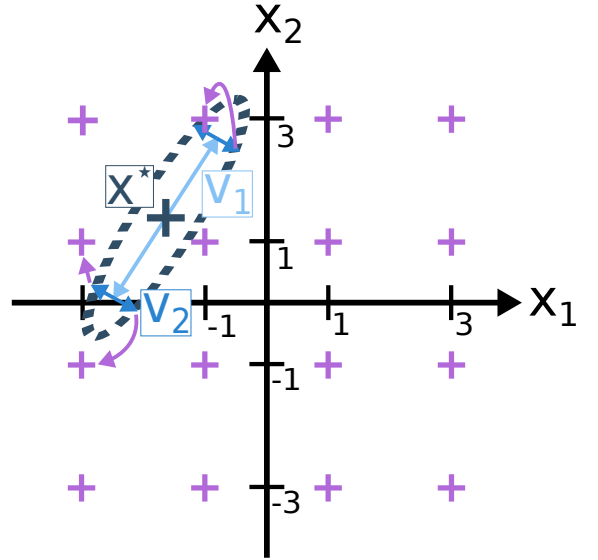


Figure 3: Well-conditioned intersection-less exploration with  $n = 2$ ,  $n_d = 2$  and a 16-QAM.

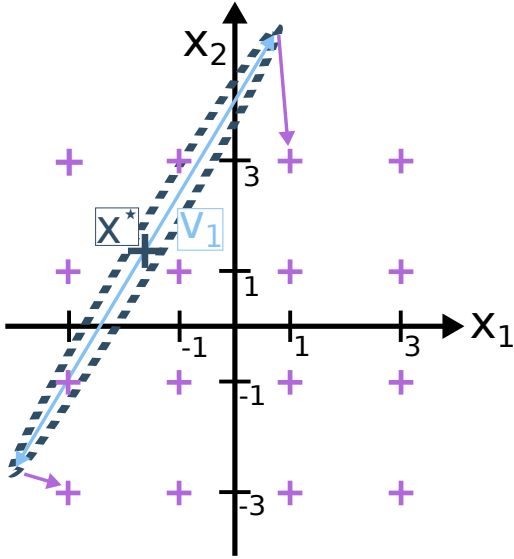


Figure 4: Poorly-conditioned intersection-less exploration with  $n = 2$ ,  $n_d = 1$  and a 16-QAM.

points. Fig. 2 and 3 show an ideal case whereas Fig. 4 presents a poorly conditioned case with  $\lambda_1 \ll \lambda_2$ . In the latter, the explored points are not at all promising, which compromises the construction of  $\mathcal{P}$ . For this reason, we introduce the scaling coefficient  $f$  in (17) which selects the sampled iso-value. A large  $f$  induces a large iso-value and thus widely spaced points whereas a small scaling coefficient produces closer points. Thus, the proposed exploration uses a few different scales to increase the opportunity that at least one of the iso-values will generate suitable points.

### 5.2.2 Sub-Constellation Projection

It is required that  $\mathcal{S}$  includes the points minimizing each of the two terms for (11) to be equivalent to reference (10). If not, the result of (11) is a close but inexact approximation of (10). The described exploration is not guaranteed to yield these two minima. Thus, the process can be modified to force the search for points in each sub-constellations  $\mathcal{X}_{ij}^k$ . This goal is easily achieved by altering the final projection. The point construction steps from  $x^*$  and singular vectors can be kept as it. Subsequently, the projection is no longer performed on  $\mathcal{Q}^n$  but on every set  $\mathcal{X}_{ij}^k$ . The overhead of this method is to move from 1 projection to a  $2nq$  projections.

## 5.3 Complexity Reduction

As for the majority of detectors, the overriding computational cost is the objective function evaluation. Thus, any refinement of this step greatly decreases the complexity. The efforts are focused on the number of products since this type of operation is considerably more complex than an addition.

The products involved during the objective function evaluation are due to the squares in the norm and to the computation of

$$\mathbf{H}\mathbf{x} = \sum_{j=1}^n \mathbf{h}_j x_j \quad (19)$$

with  $\mathbf{h}_j$  the  $j^{\text{th}}$  column of  $\mathbf{H}$ . A naive computation of (19) would require  $n$  products for each  $\mathbf{h}_j x_j$  and then  $n^2$  additions to add up all the vectors. However, it is noteworthy that the values  $x_j$  are in the finite set  $\mathcal{Q}$  so that there are only  $q$  possibilities for the product  $\mathbf{h}_j x_j$ . For instance, with  $\mathcal{Q} = \{-3, -1, 1, 3\}$  (i.e., using a 16-QAM), a product  $\mathbf{h}_j x_j$  can either be  $-3\mathbf{h}_j$ ,  $-\mathbf{h}_j$ ,  $\mathbf{h}_j$  or  $3\mathbf{h}_j$  but nothing else. Generally speaking, there are  $q$  different possibilities. Based on this property and remarking that  $\mathbf{H}$  is known for a whole block, we can preprocess and store the  $q$  possible products  $\mathbf{h}_j x_j$ . This is equivalent to preprocess and store the matrices  $-3\mathbf{H}$ ,  $-\mathbf{H}$ ,  $\mathbf{H}$  and  $3\mathbf{H}$  for 16-QAM and in the general case, to store and compute the set  $\{s\mathbf{H} : s \in \mathcal{Q}\}$ . Then, the computation of (19) become the sum of  $n$  known vectors and does not require any product at all.

This technique dramatically reduces the amount of product required to decode the symbols at the cost of  $qn^2$  coefficients storage. Indeed, one have to store the channel matrix  $\mathbf{H}$  multiplied by each possible symbol in  $\mathcal{Q}$ . This storage space and the number of precalculations can be halved in ordinary situations since the usual constellations allow to obtain the negative symbols by changing the sign. Continuing the previous 16-QAM example, storing  $\mathbf{H}$  and  $3\mathbf{H}$  is enough to compute (19) with only additions since  $-\mathbf{H}$  and  $-3\mathbf{H}$  can be obtained by subtracting rather than adding the preprocessed vectors.

## 5.4 Summary of the New Detector

Fig. 5 sums up the new geometrical algorithm presented in this paper. The upper part depicts the block-wise pre-processing. It includes the computation of the Moore-Penrose inverse  $\mathbf{H}^+$  as well as the pre-processing introduced in section 5.3. Since these calculations are only performed once per block of hundreds of symbol vectors, their complexity is negligible compared to the remaining part of the algorithm.



Table 1: Complexity evaluation for each step of the proposed geometrical detector.

Step	Nb prods	Nb adds
Compute $\mathbf{x}^* = \mathbf{H}^+ \mathbf{y}$	$n^2$	$(n-1)n$
Add prescaled $\mathbf{v}_i$ according to (17)	0	$pn$
Objective function evaluation	$pn$	$2pn$
One local search iteration	$2pn^2$	$3pn$
Max-log approximation.	$n \log_2(q)$	$n \log_2(q)$

The lower part of the figure illustrates the detection process for a symbol from the received vector and the estimated noise variance. During the exploration step, the optimal vector is first derived using the pre-calculated  $\mathbf{H}^+$ . It is then added to the sum  $\sum_{i=1}^{n_d} \frac{\pm f}{\lambda_i} \mathbf{v}_i$  that is also precomputed by the previous preprocessing. These points are then projected on the constellation and optionally on the sub-constellations, as described in Section 5.2. All these points are used to initialize the subset  $\mathcal{S}$ . During the exploitation step, some local search iterations are performed to add new promising points to  $\mathcal{S}$  (see Section 4.2). Eventually, the LLRs are estimated using the approximation of (11).

Table 1 details the complexity for each steps of the proposed geometrical detector with  $p = 2^{n_d n_f}$  the number of promising points in  $\mathcal{P}$  and  $n_f$  the number of scaling factor used. The preprocessed operations computed at a block level are neglected.

## 6 Comparisons With Tree-Based References

The proposed algorithm is to be compared with the reference detectors according to two criteria: BER performance and computational complexity. Indeed, highly complex detectors are required to provide the best performance in unfavorable use cases (low SNR regime). On the contrary, a simpler algorithm is preferable to increase the transmission rate under favorable conditions. Besides, the flexibility of the algorithm is an attractive feature for matching the performance-complexity trade-off to the use cases.

Section 6.1 described the tree-based reference to be simulated using the Monte-Carlo setup proposed in Section 6.2. Section 6.3 introduces the Pareto efficiency and Section 6.4 provides the comparisons using this framework.

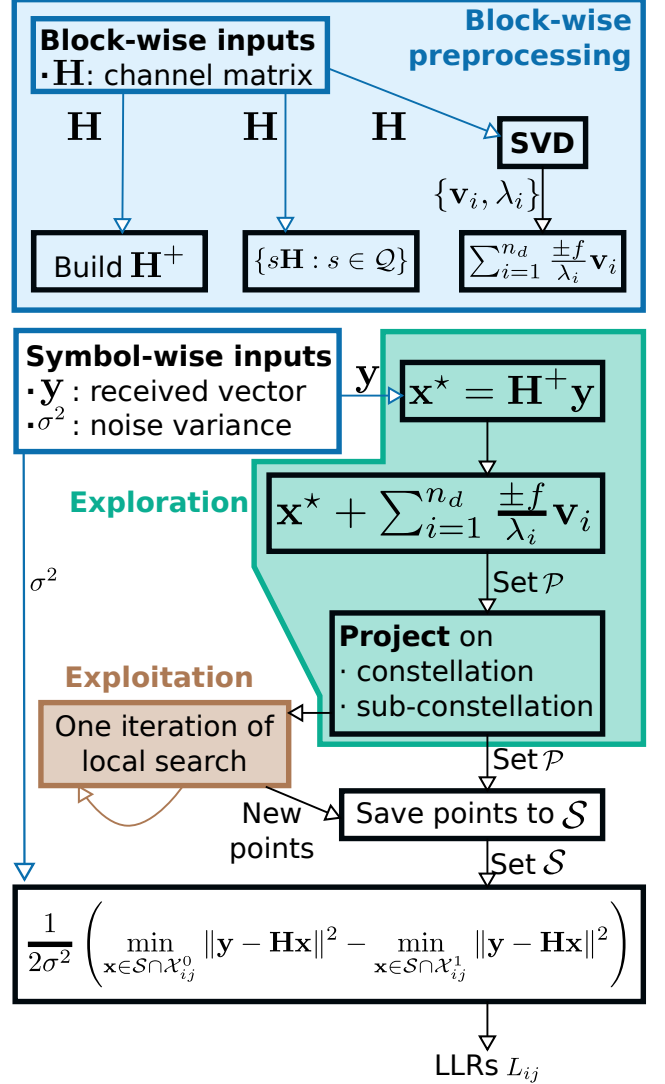


Figure 5: Summary of the proposed new geometrical detector.

## 6.1 Reference Tree-Based Detectors

The tree-based paradigm represents the detection problem as the search for the best path in a weighted tree. Any path starting from the root and reaching a leaf represents a decoded vector  $\mathbf{x}$  whose total weight corresponds to the objective function. Each node has as many children as the number of symbols in  $\mathcal{Q}$ . Selecting a node at each level corresponds to detect the corresponding component of  $\mathbf{x}$ . In the following, we refer to “extend a node” the process of computing the children’s objective function of a particular node to extend the current path.

As discussed in the introduction, tree-based detectors can be classified according to the method of searching into the tree. In this section, we describe two tree-based references: a canonical breadth-first K-best with Schnorr-Eurner enumeration [11] and a state-of-the-art best-first detector [15].

### 6.1.1 Breadth-First Detection: K-Best algorithm

The breadth-first approach builds paths from the root to the leaf with no backward step. At each level of the tree, each surviving path is extended. The  $K$  best new paths are preserved to be expended at the next level while the others are pruned. The process ends when it reaches the leaf level.

Table 2 details the computational complexity of the K-best detector as described in [11] with two metrics: the number of products and the number of additions. The complexity grows linearly with  $K$  and quadratically with  $n$ . It is assumed that the QR decomposition is performed on a block basis and therefore, can be neglected. Moreover, some preprocessing can be performed on a block basis to divide by a factor of about five the required  $K$  to reach a specific BER vs. SNR performance. As in [11], we refer by *mode 1* the algorithm without any preprocessing and by *mode 3* the one with the best preprocessing.

### 6.1.2 Best-first Detection: Cross-Level Parallel Tree-Search

We describe in this section the algorithm from [15]. The best-first approach selects the path to extend based on the partial cumulative weight of a node rather than exploring straight to the leaf (as in depth-first paradigm) or rather to keep a specified amount of paths (as in breadth-first paradigm). The cross-level variant keeps track of the best nodes at each level through several stacks of finite length. At each iteration, the best node from each stack is popped out, extended, and the best siblings and the best child are inserted in the corresponding stack. If a stack reaches its maximal length,

Table 2: Complexity evaluation of breadth-first K-best detector. Step numbers refer to the one described in Section III-B from [11].

Step	Nb prods	Nb adds
Compute $\mathbf{Q}^t \mathbf{y}$ .	$n^2$	$(n-1)n$
For each level $l$ from $n$ to 1	repeated $n$ times	
$\hookrightarrow$ Step 2) Compute weight for each child.	$Kq$	$Kq$
$\hookrightarrow$ Step 2) Update cumulative weight for each child.	$Kq$	$Kq$
$\hookrightarrow$ Step 4) Update path history.	$K(l-1)$ .	$K(l-1)$
<b>Sub-total for the loop</b>	$K \left( 2qn + \frac{(n-1)n}{2} \right)$	
Max-log approximation.	$K + n \log_2(q)$	$n \log_2(q)$

Table 3: Complexity of each step of the cross-level best-first detector. The second step is performed several times, the other are only carried out once.

Step	Nb prods	Nb adds
Compute $\mathbf{Q}^t \mathbf{y}$	$n^2$	$(n-1)n$
Expand a node at level $l$	$(n-l+1)q$	$2q$
Compute LLRs from metrics	$nq$	$nq$

then the worst path is pruned. The process ends when all the stacks are empty. Moreover, the algorithm saved the cumulative weights of the best leaves so far and used them to prune paths that are already worst. This pruning criteria inspired by SD, avoids the extension of paths that are known to lead to worst solutions.

Unlike K-best detection, the best-first cross-level algorithm does not expand the same number of nodes at each run. Indeed, the number of paths pruned due to stack overflows or to poor cumulative weights depend on  $\mathbf{H}$  and  $\mathbf{y}$ . That is why it is not possible to find a closed-form for its complexity. Table 3 details the complexity for the three steps and neglects the QR decomposition for the same reason that for K-best. Reference [15] reports that the algorithm visits, on average, hundreds of nodes per detection with  $n = 4$  and a 64-QAM. In the following, we will use the average observed complexity among all the detections as the complexity of this algorithm. This metric is computed at run time using the data from Table 3.

## 6.2 Monte-Carlo Simulation Setup

BER performance of the two references and the geometrical detector are evaluated thanks to Monte-Carlo simulations. At each iteration, a 1440-bit random message is modulated, sent through a 4x4 MIMO channel, and received. After the reception, the number of binary errors is counted. For every SNRs, the simulation continues until it either gets 200 binary errors or reaches  $5 \cdot 10^5$  transmitted bits.

The message is encoded in blocks of 720 bits with an irregular, systematic low-density parity-check (LDPC) code of ratio 1/2. The parity check matrix is designed according to the WiMAX standard (IEEE 802.16e) [29]. All receivers exploit the generated LLRs to decode the message with 15 iterations of a belief-passing min-sum algorithm. All the simulations are run using the CommPy framework [30].

## 6.3 Pareto Fronts: a Tool to Study Trade-Offs

The detectors are compared based on the Pareto efficiency to study the performance-complexity trade-off objectively. A detector is said to be Pareto efficient if it is impossible to find an alternative detector that reduces either the complexity of the BER without losing on the other metric. The set of all Pareto-efficient detectors, called the Pareto front, represents the trade-off options. Indeed, switching from a Pareto-efficient detector to another means to favor complexity or BER in the trade-off. Conversely, a detector that is not Pareto efficient should not be selected because one can improve at least one of the metrics without losing on the other.

The computational complexity is assessed as the number of products alone and the number of operations (products plus additions). The first situation refers to implementations on application-specific integrated circuits (ASICs) where the overriding complexity is the number of products due to its difficulty. The second comparison is better adapted to field-programmable gate arrays (FPGAs) since embedded digital signal processors (DSPs) can compute a free addition when computing a product.

Table 4 lists all the parameters tested during the described Monte-Carlo simulations for both QPSK and 16-QAM. All combinations of the listed parameters are tested for the geometrical detector. To improve the readability of the figures, only a subset of these parameters are plotted in the following sections. In any case, all the detectors claimed to be Pareto efficient are Pareto efficient for all the parameters from Table 4.

Table 4: List of detector parameters tested during Monte-Carlo simulations.

Parameter	Values tested
K-best $K$	2, 4, 8, 12, 16, 24, 32, 48, 64, 128, 192, 256
Best-first stack lengths	(1, 12, 15), (1, 7, 10), (1, 3, 5)
Sub-constellation projection	Yes, No
Number of iterations for exploration step	0, 1, 2, 3, 4, 5, 6, 7, 8
Number of dimensions $n_d$	1, 2, 3, 4, 5, 6, 7, 8
Scaling factors $f$	From 1 to 5 factors among 0.125, 0.25, 0.5, 1, 2, 4

## 6.4 Simulation Results

The preprocessing described in Section 5.3 could be adapted to simplify the tree-based references. The results will be provided in two scenarios to permit a meaningful comparison. Firstly with the strict implementation of references and using preprocessing only for geometric detectors in Section 6.4.1 and 6.4.2. Then by extending the preprocessing to all the compared algorithms in Section 6.4.3.

### 6.4.1 Comparisons for QPSK

Fig. 6 and 7 present the Pareto efficient algorithms for a QPSK. The first one approximates the complexity by the number of products, whereas the second one estimates it by the number of operations (products plus additions). Table 5 details the parameters corresponding to the Pareto efficient detectors for the number of product/BER trade-off.

For readability reasons, only a subset of the geometric detectors tested is shown. The plots are restricted to detectors using the sub-constellation projection presented in Section 5.2.2 with two scale factors ( $f = 0.25, f = 4.0$ ) and no iterations. Indeed, these parameters include the geometric Pareto efficient detectors.

These parameters highlight that for a very small constellation, the geometrical detector should handle the performance-complexity trade-off using the number of dimensions  $n_d$ . Indeed, the receivers can select the number of dimensions based on the measured SNR, build the promising set  $\mathcal{P}$  using the

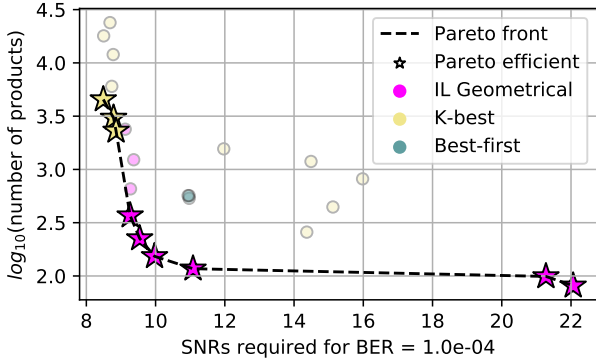


Figure 6: Pareto front for a QPSK: number of products/BER trade-off.

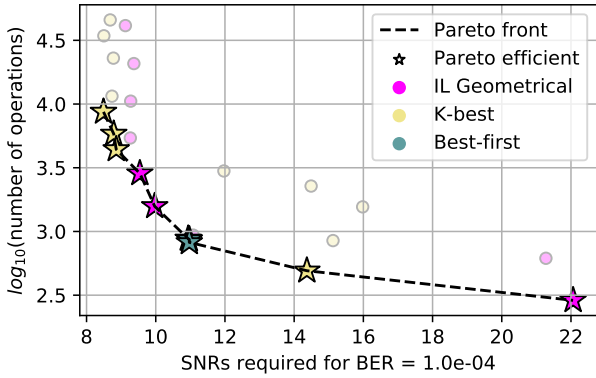


Figure 7: Pareto front for a QPSK: number of operations/BER trade-off.

sub-constellation variant, and then apply the max-log approximation without further exploitation.

Fig. 6 shows that the proposed algorithm requires ten times fewer products than the canonical K-best *mode 1* at the cost of 0.7 dB. Moreover, the geometrical allow for an on-the-fly configuration since changing  $n_d$  is enough to select the working point. Fig. 7 shows tighter spreads with the three detectors being efficient at some SNR range.

#### 6.4.2 Comparisons for 16-QAM

Fig. 8 and 9 provides the same trade-off analysis for a 16-QAM and Table 6 details the parameters of the Pareto efficient detectors. In this section, we only plot the geometrical detectors with  $n_d = 2$  and no sub-constellation projection.

Table 5: Pareto efficient detectors for number of products/BER trade-off with QPSK. Geometrical detectors reported in this table are all tuned with  $f \in \{0.25, 4\}$  and no iterations at all. The sub-constellation projection is enabled.

Detector	$\log_{10}(\text{prods})$	$\log_{10}(\text{ops})$	SNRs for $1.10^{-4}$
K-Best $K = 48$	3.66	3.94	8.5 dB
K-Best $K = 32$	3.48	3.77	8.8 dB
K-Best $K = 24$	3.36	3.64	8.9 dB
Geom $n_d = 5$	2.57	3.73	9.3 dB
Geom $n_d = 4$	2.35	3.46	9.6 dB
Geom $n_d = 3$	2.18	3.20	10.1 dB
Geom $n_d = 2$	2.07	2.98	11.1 dB
Geom $n_d = 1$	2.00	2.79	21 dB
Geom $n_d = 0$	1.90	2.46	22 dB

It is no more efficient to project on each sub-constellation as the number of projection required grows linearly with the constellation size. Therefore, the overhead is no more neglected when switching from QPSK to 16-QAM.

Table 6 highlights that the on-the-fly configuration relies on the number of iterations and the number of scaling factors when using a 16-QAM. Both Pareto fronts show that the geometrical detector is efficient at some SNR ranges. However, the complexity gap between tree-based and geometrical detectors is not as important as for QPSK.

The hierarchy observed in QPSK remains the same for 16-QAM scheme:

- canonical K-best remains effective for the worst SNRs with very high  $K$ ,
- best-first is effective but not flexible,
- the geometrical algorithm provides good results and adaptability for moderate SNRs.

#### 6.4.3 Comparisons With Equivalent Preprocessing

Fig. 10 and 11 represent the same comparison when all detectors are using the same preprocessing as described in section 5.3. For tree-based algorithms, this results in a halving of the number of products. Indeed, the norm computation now requires only products for the squaring. In this scenario, the Pareto efficient detectors were all Pareto efficient in the previous one, and we refer to the previous table for the parameters.

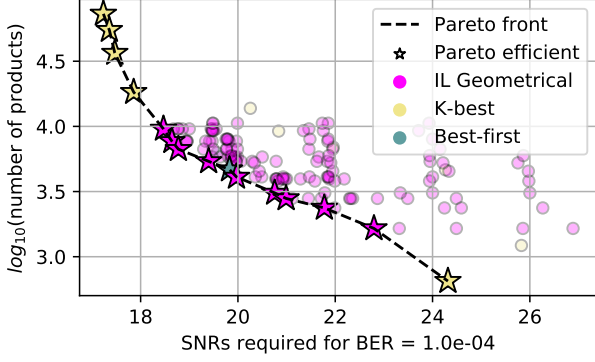


Figure 8: Pareto front for a 16QAM: number of products/BER trade-off.

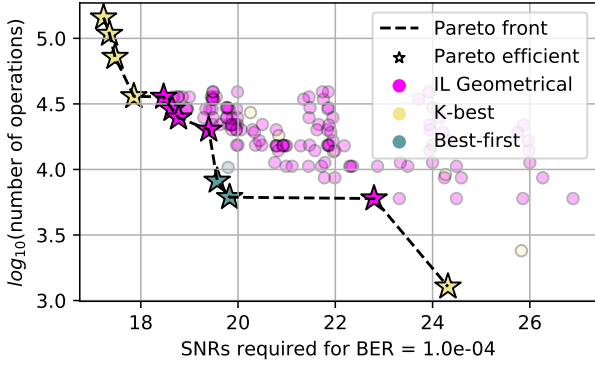


Figure 9: Pareto front for a 16QAM: number of operations/BER trade-off.

Table 6: Pareto efficient detectors for number of products/BER trade-off with 16-QAM. Geometrical detectors reported in this table are all tuned with  $n_d = 2$ . The sub-constellation projection is disabled.

Detector	$\log_{10}(\text{prods})$	$\log_{10}(\text{ops})$	SNRs for $1.10^{-4}$
K-Best $K = 256$	4.86	5.16	17.2 dB
K-Best $K = 192$	4.74	5.03	17.4 dB
K-Best $K = 128$	4.56	4.86	17.5 dB
K-Best $K = 64$	4.26	4.56	17.9 dB
Geom 5 its, $f \in \{0.25, 1, 2\}$	3.98	4.55	18.5 dB
Geom 4 its, $f \in \{0.5, 1, 2\}$	3.86	4.46	18.7 dB
Geom 5 its, $f \in \{0.5, 2\}$	3.82	4.40	18.8 dB
Geom 4 its, $f \in \{0.5, 2\}$	3.73	4.30	19.4 dB
Best-first (1, 3, 5)	3.67	3.79	19.8 dB
Geom 3 its, $f \in \{0.5, 2\}$	3.61	4.19	20 dB
Geom 4 its, $f = 1$	3.49	4.05	20.7 dB
Geom 2 its, $f \in \{0.25, 1\}$	3.45	4.02	21 dB
Geom 3 its, $f = 1$	3.74	3.94	21.9 dB
Geom 2 its, $f = 0.5$	3.21	3.78	22.8 dB
K-Best $K = 2$	2.81	3.11	24.3 dB

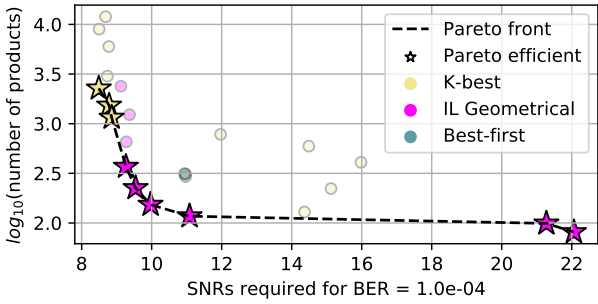


Figure 10: Pareto front for a QPSK: number of products/BER trade-off.

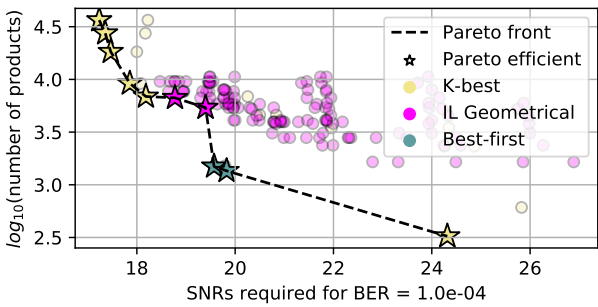


Figure 11: Pareto front for a 16QAM: number of operations/BER trade-off.

We do not know any tree-based implementation featuring this kind of preprocessing. However, [15] reports in section IV-C1 a computation method leading to an equivalent decrease in complexity. Indeed, the products  $\mathbf{H}\mathbf{x}$ , are computed only by shifts and additions, which restricts similarly the number of products required.

Fig. 11 illustrates that although being Pareto efficient, the geometrical detector is not a wise choice in 16-QAM when preprocessing is possible for the tree-based detectors. K-best is preferable to save in SNRs, and best-first provides a significantly simpler option.

Fig. 10 shows that the proposed geometrical detector is particularly relevant in QPSK. The complexity gain is considerable, and the trade-off between computational complexity and BER performance can be easily achieved with few parameters.

## 7 Conclusion

In this paper, we proposed an extensive comparison of a new geometrical detector with tree-based references. The performance-complexity trade-off is studied using the Pareto efficiency framework. We presented new exploration techniques to improve the performance-complexity trade-off and extend the geometrical detectors to higher-order modulation schemes. Moreover, a preprocessing method is introduced to reduce further the number of products required.

The Pareto fronts show that K-best is suitable in the worst SNR regimes, whereas the geometrical detector and best-first are efficient when SNRs are moderate. Besides, the proposed detector outperforms the two references in the QPSK scenario by providing a significant gap in complexity and by allowing a simple on-the-fly configuration. Moreover, the gap is large enough so that switching from *mode 1* to *mode 3* is not enough for K-best to reach the same performance as the proposed algorithm.

## References

- [1] D. Micciancio, “The hardness of the closest vector problem with preprocessing,” *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 1212–1215, Mar. 2001. [Online]. Available: <http://ieeexplore.ieee.org/document/915688/>
- [2] U. Fincke and M. Pohst, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,” *Mathematics of Computation*, vol. 44, no. 170, pp. 463–471, 1985.
- [3] C. P. Schnorr and M. Euchner, “Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems.” *Mathematical programming*, vol. 66, no. 1-3, pp. 181–199, 1994.
- [4] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, “Closest point search in lattices,” *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.
- [5] J. Jalden and B. Ottersten, “On the complexity of sphere decoding in digital communications,” *IEEE Transactions on Signal Processing*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.
- [6] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, “V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless

- channel,” in *URSI International Symposium on Signals, Systems, and Electronics.*, Oct. 1998, pp. 295–300.
- [7] A. Riadi, M. Boulouird, and M. M. Hassani, “ZF/MMSE and OSIC Detectors for UpLink OFDM Massive MIMO systems,” in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, Apr. 2019, pp. 767–772.
- [8] W. Chin, A. Constantinides, and D. Ward, “Parallel multistage detection for multiple antenna wireless systems,” *Electronics Letters*, vol. 38, no. 12, p. 597, 2002.
- [9] C. Studer, S. Fateh, and D. Seethaler, “ASIC Implementation of Soft-Input Soft-Output MIMO Detection Using MMSE Parallel Interference Cancellation,” *IEEE Journal of Solid-State Circuits*, vol. 46, no. 7, pp. 1754–1765, Jul. 2011.
- [10] K. K. Y. Wong and P. J. McLane, “A Low-Complexity Iterative MIMO Detection Scheme Using the Soft-Output M-Algorithm,” *IST Mobile Summit*, p. 5, Jun. 2005.
- [11] Z. Guo and P. Nilsson, “Algorithm and implementation of the K-best sphere decoding for MIMO detection,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 491–503, Mar. 2006.
- [12] Z. Yan, G. He, Y. Ren, W. He, J. Jiang, and Z. Mao, “Design and Implementation of Flexible Dual-Mode Soft-Output MIMO Detector With Channel Preprocessing,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 11, pp. 2706–2717, Nov. 2015.
- [13] S.-J. Choi, S.-J. Shim, Y.-H. You, J. Cha, and H.-K. Song, “Novel MIMO Detection With Improved Complexity for Near-ML Detection in MIMO-OFDM Systems,” *IEEE Access*, vol. 7, pp. 60 389–60 398, 2019.
- [14] C.-H. Liao, T.-P. Wang, and T.-D. Chiueh, “A 74.8 mW Soft-Output Detector IC for 8 x 8 Spatial-Multiplexing MIMO Communications,” *IEEE Journal of Solid-State Circuits*, vol. 45, no. 2, pp. 411–421, Feb. 2010.
- [15] G. He, X. Zhang, and Z. Liang, “Algorithm and Architecture of an Efficient MIMO Detector With Cross-Level Parallel Tree-Search,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 2, pp. 467–479, Feb. 2020, conference Name: IEEE Transactions on Very Large Scale Integration (VLSI) Systems.
- [16] J. C. Hedstrom, C. H. Yuen, R.-R. Chen, and B. Farhang-Boroujeny, “Achieving Near MAP Performance With an Excited Markov Chain Monte Carlo MIMO Detector,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 7718–7732, Dec. 2017, conference Name: IEEE Transactions on Wireless Communications.
- [17] G. C. G. Fernandes and M. G. S. Bruno, “Complexity-Reduced Suboptimal Equalization with Monte Carlo Based MIMO Detectors,” in *European Signal Processing Conference (EUSIPCO)*. A Coruna, Spain: IEEE, Sep. 2019, pp. 1–5. [Online]. Available: <https://ieeexplore.ieee.org/document/8902499/>
- [18] N. Samuel, T. Diskin, and A. Wiesel, “Deep MIMO detection,” in *EEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Jul. 2017, pp. 1–5, iSSN: 1948-3252.
- [19] J. C. Marinello and T. Abrão, “Lattice Reduction Aided Detector for MIMO Communication Via Ant Colony Optimisation,” *Wireless Personal Communications*, vol. 77, no. 1, pp. 63–85, Jul. 2014. [Online]. Available: <http://link.springer.com/10.1007/s11277-013-1495-z>
- [20] B. Nasiri and M. R. Meybodi, “History-driven firefly algorithm for optimisation in dynamic and uncertain environments,” *International Journal of Bio-Inspired Computation*, vol. 8, no. 5, p. 326, 2016. [Online]. Available: <http://www.inderscience.com/link.php?id=10000417>
- [21] A. Datta and V. Bhatia, “A near maximum likelihood performance modified firefly algorithm for large MIMO detection,” *Swarm and Evolutionary Computation*, vol. 44, pp. 828–839, Feb. 2019. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S22106502173045>
- [22] B. Trotobas, Y. Akourim, A. Nafkha, Y. Louet, and J. Weiss, “Evaluation of the Complexity, Performance and Implementability of Geometrical MIMO Detectors: the Example of the Exploration and Exploitation List Detector,” *International Journal On Advances in Telecommunications*, 2020. [Online]. Available: <https://hal-centralesupelec.archives-ouvertes.fr/hal-02615174>
- [23] E. Björnson, E. G. Larsson, and T. L. Marzetta, “Massive MIMO: Ten Myths and One Critical Question,” *IEEE Communications Magazine*, vol. 54, no. 2, pp. 114–123, Feb. 2016, arXiv: 1503.06854. [Online]. Available: <http://arxiv.org/abs/1503.06854>

- [24] C. Xu, D. Liang, S. Sugiura, S. X. Ng, and L. Hanzo, “Reduced-Complexity Approx-Log-MAP and Max-Log-MAP Soft PSK/QAM Detection Algorithms,” *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1415–1425, Apr. 2013. [Online]. Available: <http://ieeexplore.ieee.org/document/6461035/>
- [25] W. Koch and A. Baier, “Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference (applicable to digital mobile radio receivers),” in *IEEE Global Telecommunications Conference and Exhibition (GLOBECOM)*, Dec. 1990, pp. 1679–1684.
- [26] P. Robertson, E. Villebrun, and P. Hoeher, “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain,” in *IEEE International Conference on Communications (ICC)*, vol. 2, Jun. 1995, pp. 1009–1013.
- [27] B. Hochwald and S. ten Brink, “Achieving near-capacity on a multiple-antenna channel,” *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389–399, Mar. 2003.
- [28] A. Nafkha, E. Boutillon, and C. Roland, “Quasi-maximum-likelihood detector based on geometrical diversification greedy intensification,” *IEEE Transactions on Communications*, vol. 57, no. 4, pp. 926–929, Apr. 2009.
- [29] IEEE Std 802.16e, “IEEE Standard for Local and metropolitan area networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands,” 2005.
- [30] V. Taranalli, B. Trotobas, and contributors, “CommPy: Digital Communication with Python, revision f4c68b,” Mar. 2020. [Online]. Available: [github.com/veeresht/CommPy](https://github.com/veeresht/CommPy)