

Quasi-Maximum-Likelihood Detector Based on Geometrical Diversification Greedy Intensification

Amor Nafkha*, Emmanuel Boutillon[†] and Christian Roland[†]

Université Européenne de Bretagne

* SUPELEC/IETR, CNRS UMR 6164, Av. de la Boulaie CS 47601, F-35576
Cesson-Sévigné, France

Email: amor.nafkha@supelec.fr

[†] Lab-STICC, CNRS FRE 3167, Université de Bretagne Sud, BP 92116 56321
Lorient, France

Email: emmanuel.boutillon@univ-ubs.fr, christian.roland@univ-ubs.fr

Abstract

The ML detection used in many communication applications reduces down to solving an integer least-squares problem, which minimizes an objective function. If the matrix is not orthogonal, this problem is NP-hard. An exhaustive search over all feasible solutions is thus only applicable when the dimension of the problem is low. In the literature, there are two main available classes of methods. The first class comprises of exact methods whose complexities are roughly cubic in the dimension for medium to high SNR. However, the computational complexity remains exponential for low SNR. Moreover, these algorithms are iterative and thus, not well suited for hardware implementation. The second class is based on a relaxation strategy approach. However, the BER performance of this class is still not optimum. In this paper, we use a third class of methods for the resolution of this problem. Our approach, named Geometrical Diversification and Greedy Intensification (GDGI) associates two complementary heuristic optimization methods (*intensification* and *diversification*). Three versions of the GDGI are presented and compared. The inherent parallel structure of the GDGI provide a very suitable real-time hardware implementation. The GDGI allows a near optimal performance, with cubic complexity and it's almost independent of the SNR.



Index Terms

Maximum Likelihood Detection, MIMO systems, MC-CDMA systems, Singular Value Decomposition, Greedy algorithm.

I. INTRODUCTION

The main challenge of the receiver design for wireless communication systems lies in the non-orthogonality of the transmission channel. In order to secure high reliability of data transmission special attention has to be paid to the design of the receiver. Optimum maximum likelihood detection requires the determination of the signal point $\hat{\mathbf{x}}$ of the transmitter vector signal set ξ^n that minimizes the objective function $f(\cdot)$:

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in \xi^n}{\operatorname{argmin}} f(\mathbf{x}) \quad (1)$$

The function $f(\cdot)$ is the Euclidean distance between the received signal vector \mathbf{y} ($\mathbf{y} \in \mathbb{R}^m$) and the lattice point $\mathbf{H}\mathbf{x}$ ($\mathbf{H} \in \mathbb{R}^{m \times n}$).

$$f(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \quad (2)$$

In the following, $m = n$ by default. We assume that the information signal \mathbf{x} is uniformly distributed over a discrete and finite set $\xi^n = \{\pm 1\}^n$ (the set of constellation points). The channel matrix \mathbf{H} is considered as constant over a block of L consecutive time intervals and it is assumed to be perfectly known at the receiver end.

An exhaustive search over all feasible solutions ξ^n can be used to solve the maximum likelihood problems (1). However, the computational complexity is exponential in the number of possible constellation points, thus making this alternative unsuitable for practical purposes when aiming at high spectral efficiencies. Nevertheless, for low problem dimension with low-order modulation schemes such as BPSK and 4-QAM, exhaustive search methods have been shown to be feasible and efficient [14].

For higher problem dimension, the maximum likelihood detection problem (1) can be resolved using a “*smart*” efficient search method like the universal lattice decoding algorithm [1], also called sphere decoding (SD). The SD search algorithm is based on the Finke-Pohst to enumerate all the lattice points inside an hyper-sphere centered at the origin [3]. The interest in lattice

decoding has steadily increased in the last few years. One of the most efficient sphere decoding algorithms that has been proposed in the literature is the algorithm of Viterbo and Boutros (VB) in [4]. However, an important drawback in the VB algorithm, and other similar SD algorithms, is the choice of the initial value of the search radius. On one hand, if this radius is chosen too small, there may be no solution for the algorithm (no point inside the hyper-sphere). On the other hand, if the radius is chosen too large, the number of the checked points may be very high and the algorithm will be ineffective. The average complexity of the SD algorithm is shown to be polynomial time (almost cubic) over a certain ranges of rate, Signal to Noise Ratio (SNR) and dimension, while the worst-case complexity is still exponential [12]. From a hardware implementation point of view, the SD algorithm presents three weaknesses: first, it is iterative in nature and cannot be easily pipelined. Second, its average complexity increases for low SNR. Third, the number of branches evaluated in a SD algorithm has a large variation. Nevertheless, several implementations of SD has been reported in the literature [20], [22], [23]. They show that, despite of these limitations, the SD decoder can be implemented in effective way.

The second class of methods is based on a relaxation strategy approach. It consists of finding the optimal solution of $f(\cdot)$ on a convex set that includes $\{\pm 1\}^n$ and then projecting the returned solution on the discret set $\{\pm 1\}^n$. This class encompasses linear and non linear receivers. Linear receiver, like zeros forcing (ZF) and minimum mean-squared error (MMSE) [2] are computationally simple to implement, however, their performance can be far from the ML performance. Therefore, there has been a considerable interest in non linear ML approximations, which offer a better performance. The non linear receivers are successive interference cancellation (SIC), parallel interference cancellation (PIC) [10], [11] and the recently proposed semidefinite programming (SDP) receiver [5], [6], [7]. Simulation results indicate that the bit error rate (BER) performance of the SDP detector is better than those of previous non linear receivers. However, SDP is still non optimum and its complexity is high. From a hardware perspective, SDP lacks of parallelism because of its iterative structure.

In this paper, we use a third class of methods for the resolution of the ML detection problem. Following the pioneering work of Hui and Rasmussen [15], we consider the ML detection problem as an operation research problem and use a heuristic method to solve it. A heuristic technique seeks for local optimal solutions at a reasonable computational cost, but does not

guarantee their optimality. In practice, it should be emphasized that many modern heuristic techniques do give high-quality solutions [9] in the general case. This is also true for the ML detection problem as shown in [15]. The originality of the presented work is to combine the existing methods for deriving a new heuristic method having quasi-optimal performance, limited complexity and good properties for hardware implementation (low complexity and highly parallel structure), even for the problem of high dimension ($n = 60$ has been successfully tested).

The proposed detector, called Geometrical Diversification and Greedy Intensification (GDGI), combines two complementary approaches to approximate the optimal solution $\hat{\mathbf{x}}$. First, using the geometrical properties of the problem, we generate a subset $\xi_{start} \subset \xi^n$ of points \mathbf{x} “close” to the received point \mathbf{y} according to the objective function $f(\cdot)$ (*diversification approach*). Then, starting from each point of ξ_{start} , we perform the simple bit-flipping greedy algorithm to improve the initial starting solution (*intensification approach*). The best point obtained is then the decoded point. The main idea of this method is: “*don’t put all your eggs in one basket*”. The GDGI detector applies a geometrical approach to implement the defined complementary techniques and in general starts its search from a simple solution given by a linear detector (ZF or MMSE). The reduced subset ($\xi_{start} \subset \xi^n$) creation by the diversification scheme, is inspired from the very original works of H. Artes in [8] and P. Spasojevic in [13]. In [8], the authors discuss the effects of bad (“*poorly conditioned*”) channel realizations on the sub-optimum detectors’ performance and then propose a new detection method called the Sphere-Projection Algorithm (SPA). This strategy uses the solution given by ZF as the initial solution and extends the search of other best solutions into the direction of the dominant axis noise. In [13], P. Spasojevic has proposed a very efficient geometrical method for generating a subset of points close to the optimal solution. In this paper, we use this method (called “*slowest descent method*”) and two new variants to obtain the subset $\xi_{start} \subset \xi^n$.

The GDGI detector is not, by far, just a generalization of the geometrical approach proposed in [13] for the diversification step, rather it is a new strategy using an intensification phase to explore promised regions. The proposed algorithm sets the initial solution to the ZF solution and then searches for candidates in the direction of the D smallest right singular vectors of the channel matrix \mathbf{H} . After the choice of the C best initial candidates in each direction, the GDGI detector uses an intensification step to provide a quasi-optimal solution for the ML problem.

The next section briefly reviews the state of the art in the existing heuristic techniques and

describes and analyze the greedy algorithm used in the intensification step. Subsequently, in Section III, three efficient geometric diversification techniques are described for different variants of the GDGI detector. Several simulation results and comparative analysis are provided in Section IV which demonstrate the efficiency of the GDGI detector as a quasi-optimal ML detector. In Section V, the computational complexity of the GDGI algorithm is explained and an architecture for the efficient hardware implementation is described. The paper concludes with a summary in Section VI.

II. THE INTENSIFICATION STEP

As shown in [15], and [17] heuristic methods such as local search, simulated annealing, and tabu search can lead to excellent decoding performance. These algorithms have a long iterative structure and are not suited for implementation. In this paper, we focus our attention on the greedy algorithm (or bit-flipping algorithm) that can be easily implemented in the hardware. The greedy algorithm is first described and its complexity is evaluated, then its behavior is analyzed and a diversification method to escape local minimum is presented.

A. Bit-flipping greedy algorithm : (*decent-1 algorithm*)

As presented in [15], [16], and [24] the principle of the bit-flipping greedy algorithm is very simple. Starting from a current solution \mathbf{x} , the set of its neighbors is generated by flipping the sign of one of its coordinates (i.e. the set $N(\mathbf{x})$ of all the points of ξ^n at a Hamming distance 1 of \mathbf{x}). The objective function $f(\mathbf{x})$ is evaluated among all the points of $N(\mathbf{x})$ and the point \mathbf{x}' leading to the minimum objective function over $N(\mathbf{x})$ is then used as the new current solution if $f(\mathbf{x}') \leq f(\mathbf{x})$. The process is iterated starting from the new current solution \mathbf{x}' . When a local optimum is found (i.e. no better solution is found), the algorithm stops and outputs the local minimum $\varphi(\mathbf{x})$.

The algorithm appears to be computationally expensive, due to the number of repeated cost function calculations required, but this may be largely reduced through simplification. Let \mathbf{x} be the current solution and let $\Omega = \mathbf{y} - \mathbf{H}\mathbf{x}$. Equation (2) is then equivalent to:

$$f(\mathbf{x}) = \|\Omega\|_2^2 = \sum_{i=0}^{n-1} \Omega(i)^2$$

where

$$\Omega(i) = \mathbf{y}(i) - \sum_{j=0}^{n-1} \mathbf{H}(i, j) \mathbf{x}(j), \quad i = 1..n \quad (3)$$

The first evaluation of the objective function $f(\cdot)$ requires n^2 additions ($\mathbf{x}(i)$ is equal to $+1$ or -1) and n squares.

If the k^{th} coordinate is flipped (i.e. $\mathbf{x}'(k) = \mathbf{x}(k) + z$, with $z = -2\mathbf{x}(k)$), then $\Omega' = \mathbf{y} - \mathbf{H}\mathbf{x}'$ is equal to:

$$\Omega'(i) = \Omega(i) + z\mathbf{H}(i, k), \quad i = 1..n \quad (4)$$

Since Ω is known from the previous cost calculation, the computation of Ω' requires only n additions. The final computation of $\|\Omega'\|_2^2$ requires n additions and n squares.

The cost to evaluate the n points of the set $N(\mathbf{x})$ is then $2n^2$ additions and n^2 squares per step. Assuming that the cost function of the initial point is known, the global complexity of the descent-1 algorithm is equal to $\theta 2n^2$ additions and θn^2 squares where θ is the number of iterations required before reaching a local optimum.

B. Analysis and performance of the intensification step

Let $\{\tilde{\mathbf{x}}_i\}_{i=1..K}$ be the K local optimums of ξ^n according to the greedy algorithm. The intensification method partitions the set of feasible solutions ξ^n into K adjacent subsets Π_i defined as:

$$\Pi_i = \{\mathbf{x} \in \xi^n | \varphi(\mathbf{x}) = \tilde{\mathbf{x}}_i\} \quad (5)$$

where $\varphi(\mathbf{x})$ is the intensification step applied on the feasible point \mathbf{x} , and $\tilde{\mathbf{x}}_i, i = 1..K$, is the result of the application of the intensification step starting from the current solution \mathbf{x} . Π_i represents a subset where any solution belonging to it can be descended to the unique local optimum $\tilde{\mathbf{x}}_i$ if Π_i ($\tilde{\mathbf{x}}_i \in \Pi_i$). By convention, the subset Π_1 leads to achieve the optimal ML solution, i.e., $\hat{\mathbf{x}} = \tilde{\mathbf{x}}_1$, or, equivalently, $\forall \mathbf{x} \in \Pi_1, \varphi(\mathbf{x}) = \hat{\mathbf{x}}$.

The number of local minimum K can be large. For example, for $n = 16$, K is superior to 6 with a probability of 60%. The average value of K increases when the dimension n of the problem becomes larger. The number of local minimum K is large enough to cause a significant degradation of the average performance of the coordinate descent-1 algorithm.

C. Needs of diversification method

To prevent from getting stuck in local optima, which is a common problem for direct search methods and most of the nonlinear optimization algorithms, several rounds of optimization can be made with different initial guesses and picking up the best local optimum as a global sub-optimum. Let ξ_{start} be a subset of ξ^n , the decoding algorithm becomes:

$$\tilde{\mathbf{x}} = \underset{\varphi(\mathbf{x}), \mathbf{x} \in \xi_{start}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{H}\varphi(\mathbf{x})\|_2^2 \quad (6)$$

One can note that the decoded point $\tilde{\mathbf{x}}$ equals the optimal point $\hat{\mathbf{x}}$ if, and only if, at least one point of ξ_{start} belongs to Π_1 , *i.e.* $\xi_{start} \cap \Pi_1 \neq \emptyset$.

Now the question arises how to generate a “good” starting set ξ_{start} . The diversification step should verify the following properties:

- 1) Cardinality of ξ_{start} is small, *i.e.*, a reduced number of intensification processes.
- 2) $\xi_{start} \cap \Pi_1 \neq \emptyset$ with a very high probability (according to the requirement of the application).
- 3) Small computational complexity.

The next section describes an efficient method to construct ξ_{start} , based on the geometrical properties of the \mathbf{H} matrix.

III. THE DIVERSIFICATION STEP

The problem of determining a “good” starting set ξ_{start} (in the sense defined above) is a hard problem. The choice is normally guided by the intuition and the “*trial and error*” method because very few theories are available. A clear trade-off can be observed between small and large neighborhoods. A large number of point in ξ_{start} would seem promising but, as a drawback, it will be time-consuming.

The simplest solution to generate ξ_{start} is to pick up random points. In that case, the generation process of ξ_{start} is very simple but in return, the cardinality of ξ_{start} will become an important factor to fulfill $\xi_{start} \cap \Pi_1 \neq \emptyset$ with a high probability. The complexity of the algorithm is then dominated by the intensification stage. An other solution is to define a fixed set of point “spread” among the set ξ^n . Several results using this approach with an extended BCH code are presented

in [18]. This type of approach is rather efficient but limited to some values of n . Moreover, the cardinality of ξ_{start} remains still important.

In this section, we propose to generalize the geometrical method presented in [13] and use it in order to generate a subset ξ_{start} . We present the principle of the geometrical approach and we give three variants to create starting subset.

A. Principle of the geometrical approach

In general, the channel matrix is modeled by an $m \times n$ random matrix \mathbf{H} mentioned in the introduction (see Eq.1). The singular value decomposition of the matrix \mathbf{H} is $\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^T$, where the diagonal matrix Σ contains the singular values $\{\lambda_k\}_{k=1}^n$, supposed to be indexed in increasing order *i.e.* $\lambda_1 \leq \dots \leq \lambda_n$. The unitary matrices \mathbf{U} and \mathbf{V} contain, respectively, the left $\{\mathbf{u}_k\}_{k=1}^m$ and right $\{\mathbf{v}_k\}_{k=1}^n$ singular vectors of the matrix \mathbf{H} as columns. The geometrical approach can be described as follows: let $\mathbf{x}_0 = \mathbf{H}^+\mathbf{y}$, where $\mathbf{H}^+ = (\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}^T$ is the pseudo-inverse of \mathbf{H} (\mathbf{H}^T stands for the transpose matrix of \mathbf{H}), and \mathbf{x}_0 be the solution given by the ZF detector.

For all points $\mathbf{x} \in \xi^n$, the vector $\mathbf{x} - \mathbf{x}_0$ can be expressed in the \mathbf{V} base as $\mathbf{x} - \mathbf{x}_0 = \sum_{k=1}^n \alpha_k \mathbf{v}_k$ where $\{\alpha_k\}_{k=1}^n$ are real coefficients. The value of the objective function at all points $\mathbf{x} \in \xi^n$ can be expressed as:

$$\begin{aligned} f(\mathbf{x}) &= \|\mathbf{H}(\mathbf{x} - \mathbf{x}_0)\|_2^2 \\ &= (\mathbf{x} - \mathbf{x}_0)^T \mathbf{V} \Sigma^2 \mathbf{V}^T (\mathbf{x} - \mathbf{x}_0) \\ &= \sum_{k=1}^n \alpha_k^2 \lambda_k^2 \end{aligned} \quad (7)$$

Let us define $\Delta_k = \{\mathbf{z} \in \mathbb{R}^n / \mathbf{z} = \mathbf{x}_0 + \gamma \mathbf{v}_k, \gamma \in \mathbb{R}\}$ the line in \mathbb{R}^n defined by the point \mathbf{x}_0 and the vector \mathbf{v}_k . Since $\lambda_1 \leq \dots \leq \lambda_n$, we can note that the increase in the objective function $f(\cdot)$ is much slower along the first D lines $\{\Delta_1, \dots, \Delta_D\}$ than the last $(n - D)$ lines. The diversification step consists of choosing the feasible points in the “vicinity” of the lines $\Delta_1, \dots, \Delta_D$ in order to create the starting set ξ_{start} .

The method to create ξ_{start} is based on several steps. Let us first consider the line Δ_k . The first step consists of computing the intersections of the line Δ_k with a given set of Nb hyperplanes $\mathcal{S} = \{\mathcal{H}_p, p = 1..Nb\}$ in order to obtain a list of Nb points. These points $\mathbf{c}_1^p, p = 1..Nb$ are then projected on ξ^n to generate the list $\mathcal{I}_k = \{\bar{\mathbf{c}}_k^p, p = 1..Nb\}$. All these points are then evaluated

with the objective function $f(\cdot)$ ¹ and just the best C points are selected to generate a subset ξ_k . The same process is iterated on the first D lines $\Delta_1, \Delta_2, \dots, \Delta_D$ to generate $\xi_{start} = \cup_{k=1}^D \xi_k$. Then the intensification process is performed on each point of ξ_{start} . The summary of the GDGI method is given in the algorithm 1.

Algorithm 1 GDGI($\mathbf{H}, \mathbf{y}, \mathcal{S}, C, D$)

Pre-process 1: Extract the D smallest right singular vectors of the channel matrix \mathbf{H} , *i.e.*, $\{\mathbf{v}_k\}_{k=1}^D$.

Pre-process 2: Compute $\mathbf{H}^+ = (\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}^T$

1: Calculate $\mathbf{x}_0 = \mathbf{H}^+\mathbf{y}$.

2: Create an empty list ξ_{start} , initialize $k = 1$.

repeat

3: Generate the line Δ_k defined by \mathbf{x}_0 and vector \mathbf{v}_k .

4: Calculate the intersection of the line Δ_k and all hyperplanes of \mathcal{S} to generate \mathbf{c}_k^p , $p = \{1, \dots, Nb\}$.

5: Project \mathbf{c}_k^p , $p = \{1, \dots, m\}$ on ξ^n to generate \mathcal{I}_k .

6: Evaluate the objective function $f(\mathbf{x})$, $\forall \mathbf{x} \in \mathcal{I}_k$.

7: Create ξ_k by selecting C distinct points which minimize the objective function on \mathcal{I}_k .

8: Update ξ_{start} ($\xi_{start} = \xi_{start} \cup \xi_k$), $k = k + 1$.

until $k = D + 1$

9: Starting from the set ξ_{start} , perform the intensification process as described in the subsection II.A.

In the next three sections, three different variants of Geometrical Diversification (steps 4 and 5 of the algorithm 1) are presented. All this variants correspond to three different sets of hyperplane \mathcal{S} . For the sake of brevity, the index k of the direction will be omitted in those three subsections.

¹The evaluation of the objective function can also be performed with the $l-1$ norm to simplify the design. The simulation results, not shown in this paper, show only a very slight degradation of performance

B. Hypercube Intersection and Selection method (HIS)

Geometrically, the set $\xi^n = \{\pm 1\}^n$ includes the vertices of the unit hypercube of dimension n . The basic idea of this method is to determine the intersection points between Δ and the $Nb_{\mathcal{H}} = 2n$ faces of the unit hypercube. The set \mathcal{S} is thus defined as:

$$\mathcal{S}_{\mathcal{H}} = \{\mathcal{H}(i, s), s = \{-1, 1\}, i = \{1..n\}\} \quad (8)$$

where

$$\mathcal{H}(i, s) = \{\mathbf{z} \in \mathbb{R}^n | \mathbf{z}(i) = s\} \quad (9)$$

Let us analyze the intersection between a given line Δ and the hyperplane $\mathcal{H}(i, s)$. The problem is simply to obtain $\gamma^{s,i}$ expressed in the form:

$$\mathbf{x}_0(i) + \gamma^{s,i} \mathbf{v}(i) = s \quad (10)$$

Assuming $\mathbf{v}(i) \neq 0$, equation (10) has a unique solution given by $\gamma^{s,i} = \frac{s - \mathbf{x}_0(i)}{\mathbf{v}(i)}$. The generated intersection point between Δ and the hyperplane $\mathcal{H}(i, s)$ is then

$$\mathbf{c}^{s,i} = \gamma^{s,i} \mathbf{v} + \mathbf{x}_0 \quad (11)$$

Since only the projection $\bar{\mathbf{c}}^{s,i}$ of $\mathbf{c}^{s,i}$ over ξ^n is required, explicit computation of $\mathbf{c}^{s,i}$ is not mandatory. In fact, the j^{th} coordinate of $\bar{\mathbf{c}}^{s,i}$ is equal to

$$\bar{\mathbf{c}}^{s,i}(j) = \text{sign}((s - \mathbf{x}_0(i))\mathbf{v}(j) + \mathbf{x}_0(j)\mathbf{v}(i)) \times \text{sign}(\mathbf{v}(i)), j = 1, \dots, n. \quad (12)$$

where $\text{sign}(x)$ returns 1 if $x > 0$, returns -1 if $x < 0$ and returns 0 otherwise.

If the j^{th} coordinate of $\bar{\mathbf{c}}^{s,i}(j) = 0$ equals zero, than $\bar{\mathbf{c}}^{s,i}(j)$ is set to $-\text{sign}(\mathbf{x}_0(i))$ if $j = i$, and set to $\text{sign}(\mathbf{x}_0(j))$ otherwise. This additional rule allows to obtain a solution even if $\mathbf{v}(i) = 0$.

In the following, we neglect the hardware complexity of the sign processing in (12). The direct computation of $\bar{\mathbf{c}}^{s,i}$ requires then 2 multiplications and 2 additions for each coordinate, i.e. $2n$ multiplications and $2n$ additions for its n coordinates. Since \mathcal{I} contains $Nb_{\mathcal{H}} = 2n$ points, the overall complexity of a direct computation for each direction is then $4n^2$ multiplications and $4n^2$ additions. However, it is possible to factorize some computation to reduce the overall cost.

Let $\mathbf{M} = \mathbf{v}\mathbf{x}_0^T$, then:

$$(\mathbf{M} - \mathbf{M}^T)(i, j) = \mathbf{x}_0(j)\mathbf{v}(i) - \mathbf{x}_0(i)\mathbf{v}(j) \quad (13)$$

and thus,

$$\bar{\mathbf{c}}^{s,i}(j) = \text{sign}((\mathbf{M} - \mathbf{M}^T)(i, j) + s \times \mathbf{v}(j)) \text{sign}(\mathbf{v}(i)) \quad (14)$$

The computation of $\mathbf{M} - \mathbf{M}^T$ requires n^2 multiplications and n^2 additions². Once $\mathbf{M} - \mathbf{M}^T$ is determined, then according to (14) the computation of $\bar{\mathbf{c}}^{s,i}$ requires just a single addition per coordinate, thus a total of n additions per points and $2n^2$ additions for the $Nb_{\mathcal{H}} = 2n$ points of \mathcal{I} .

C. Basis Intersection and Selection (BIS)

This variant is the same as proposed by P. Spasojevic in [13]. In this method, the set $\mathcal{S}_{\mathcal{B}}$ of hyperplanes contains the $Nb_{\mathcal{B}} = n$ hyperplanes defined by:

$$\mathcal{S}_{\mathcal{B}} = \{\mathcal{H}(i, 0), i = \{1..n\}\} \quad (15)$$

where $\mathcal{H}(i, 0)$ is defined as in (9).

In this variant, the method to compute the $\bar{\mathbf{c}}^{0,i}$ is the same as for the HIS method. By construction, one can note that the i^{th} coordinate of $\bar{\mathbf{c}}^{0,i}$ is equal to zero. The value of $\bar{\mathbf{c}}^{0,i}(i)$ is then determined by $-\text{sign}(\mathbf{x}_0(i))$ as defined above.

One can note, that, since $s = 0$, the complexity to determine the set of intersecting point is limited to the computation of $\mathbf{M} - \mathbf{M}^T$, i.e. n^2 additions and n^2 multiplications to compute the $Nb_{\mathcal{B}} = n$ points of the set \mathcal{I} .

D. Orthogonal Intersection and Selection method (OIS)

As mentioned before, the objective of the diversification process is to find points closed to the line Δ . In the two previous approaches, the hyperplane $\mathcal{H}(i, s)$ and the line Δ are not necessarily orthogonal. As a result, the intersecting point $\mathbf{c}^{s,i}$ can be very far from the set ξ^n . Ideally, the hyperplanes should be orthogonal to the line Δ but this leads to a complex solution. The OIS method is an attempt to obtain a set of hyperplanes “roughly” orthogonal to the line Δ . To do so, the set of hyperplanes $\mathcal{S}_{\mathcal{O}}$ is no more constant but constructed “on the fly”. For a given

²Using the fact that $(\mathbf{M} - \mathbf{M}^T) = -(\mathbf{M} - \mathbf{M}^T)^T$, the number of addition can be further reduced to $n^2/2$

direction k , $\mathcal{S}_\mathcal{O}$ is defined as the set of hyperplanes orthogonal to the quantized vector $\mathbf{n} = Q(\mathbf{v})$ of \mathbf{v} and containing at least one point of ξ^n . The quantization function $Q(\mathbf{v})$ is defined as:

$$Q(\mathbf{v})(j) = \text{sign}(\mathbf{v}(j)) \text{ if } |\mathbf{v}(j)| > \max(\mathbf{v})/4, 0 \text{ otherwise} \quad (16)$$

where $\max(\mathbf{v})$ is the maximum module of the coordinates of the vector \mathbf{v} ³.

The coordinates of \mathbf{n} can take their values in $\{-1, 0, 1\}$. Let l be the number of non zero coordinates of \mathbf{n} . It is easy to show that the set $\mathcal{S}_\mathcal{O}$ contains exactly the $Nb_\mathcal{O} = l + 1$ hyperplanes defined as:

$$\mathcal{O}^t = \{\mathbf{z} \in \mathbb{R}^n | \mathbf{z}^T \mathbf{n} = t, t = \{-l, -l + 2, \dots, l - 2, l\}\} \quad (17)$$

For example, let $n = 4$ and $\mathbf{n} = (0, 1, -1, 1)^T$, i.e. $l = 3$, if $\mathbf{x} \in \xi^4$ then $\mathbf{x}^T \mathbf{n}$ can only takes the value $-3, -1, 1$, and 3 ($\mathbf{x} = (1, -1, 1, -1)^T$ gives $\mathbf{x}^T \mathbf{n} = -3$ and so on...).

The intersection between Δ and the hyperplane \mathcal{O}^t is then given by:

$$(\mathbf{x}_0 + \gamma^t \mathbf{v})^T \mathbf{n} = t \quad (18)$$

The value of γ^t is then equal to:

$$\gamma^t = \frac{t - \mathbf{x}_0^T \mathbf{n}}{\mathbf{v}^T \mathbf{n}} \quad (19)$$

and the intersection point \mathbf{c}^t is then:

$$\mathbf{c}^t = \left(\frac{t - \mathbf{x}_0^T \mathbf{n}}{\mathbf{v}^T \mathbf{n}} \right) \mathbf{v} + \mathbf{x}_0. \quad (20)$$

The returned point $\bar{\mathbf{c}}^t = \text{sign}(\mathbf{c}^t)$ can be defined by the same process as for the HIS and the BIS algorithms. In that case, it should be noted that for the OIS method, the simplification of equation (14) is no more feasible. The direct computational complexity, for a given dimension is then $2n \times Nb_\mathcal{O}$ multiplications and $2n \times Nb_\mathcal{O}$ to generate \mathcal{I} .

However, it is possible to dramatically simplify this complexity using the fact that all the hyperplans \mathcal{O}^t are parallel. In fact, according to (20), for $t = -l, -l + 2, \dots, l - 2$:

$$\mathbf{w} = \mathbf{c}^{t+2} - \mathbf{c}^t = \left(\frac{2}{\mathbf{v}^T \mathbf{n}} \right) \mathbf{v} \quad (21)$$

Thus, in the OIS case, it is more efficient to determine explicitly a first point (say \mathbf{c}^{-l} for example), and then find recursively the others using equation (21). Let us describe a possible sequence of computation and their precise associated costs.

³ $\max(\mathbf{v})/4$ is easy to compute and this value gives good simulation result

- 1) $d = \mathbf{v}^T \mathbf{n}$: n additions⁴;
- 2) $e = \mathbf{x}_0^T \mathbf{n}$: n additions;
- 3) $f = 1/d$: one division;
- 4) $\gamma^{-l} = f(l - e)$: one addition and one multiplication;
- 5) $\mathbf{c}^{-l} = \gamma^{-l} \mathbf{v} + \mathbf{x}_0$: n multiplications and n additions.
- 6) $\mathbf{w} = 2f\mathbf{v}$: n multiplications.
- 7) determine the last l points using (21): $l \times n$ additions.

Thus, neglecting the step 4, the global complexity is then reduced to $((3 + l)n = (2 + Nb_O)n$ additions, $2n$ multiplications and one division.

To illustrate the differences between the three variants of the algorithm, Fig. 1, 2 and 3 show all the hyperplanes, intersection points \mathbf{c} and their corresponding feasible points $\{\bar{\mathbf{c}}\}$ in the case of a $n = 2$ -dimensional problem with $D = 1$.

IV. SIMULATION RESULTS

In this section the performance of the proposed algorithm is evaluated with computer simulations. We have compared the performance of the GDGI (three variants: HIS, BIS and OIS) detector with those of other decoders including SD, SDP and MMSE.

A. Computer simulation

First, we consider a multiple-input / multiple-output (MIMO) system with N transmitter and M receiver antennas in a Rayleigh flat fading channel. The elements of the channel matrix $\tilde{\mathbf{H}} \in \mathbb{C}^{M \times N}$ are drawn from an independent and identically distributed zero-mean, unit variance Gaussian distribution. The received vector $\tilde{\mathbf{y}} \in \mathbb{C}^M$ was constructed as

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{b}} \quad (22)$$

where each entry of $\tilde{\mathbf{x}}$ was taken from $\{\pm 1 \pm j\}^N$ (4-QAM), the elements of the noise vector $\tilde{\mathbf{b}} \in \mathbb{C}^M$ are drawn from an i.i.d zero-mean Gaussian distribution. Treating real and imaginary part of (22) separately, the system model can be rewritten as $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{b}$, with the real-valued

⁴Since \mathbf{n} contains l non null values, the number of additions can be reduced to l .

channel matrix

$$\mathbf{H} = \begin{bmatrix} \Re(\tilde{\mathbf{H}}) & -\Im(\tilde{\mathbf{H}}) \\ \Im(\tilde{\mathbf{H}}) & \Re(\tilde{\mathbf{H}}) \end{bmatrix} \in \mathbb{R}^{m \times n}$$

and the real-valued vectors

$$\mathbf{x} = \begin{bmatrix} \Re(\tilde{\mathbf{x}}) \\ \Im(\tilde{\mathbf{x}}) \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \Re(\tilde{\mathbf{y}}) \\ \Im(\tilde{\mathbf{y}}) \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \Re(\tilde{\mathbf{b}}) \\ \Im(\tilde{\mathbf{b}}) \end{bmatrix}$$

where $\Re(\tilde{\mathbf{x}})$ and $\Im(\tilde{\mathbf{x}})$ denote the real and imaginary part of $\tilde{\mathbf{x}}$, respectively. By defining $m = 2M$ and $n = 2N$ the dimension of the real channel matrix is given by $m \times n$. Likewise the dimension of the vectors are given by $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{x} \in \xi^n \equiv \{\pm 1\}^n$.

The simulations have been done for the values of $n = [10, 40, 60]$ and $m = n$.

Figure 5 shows the results of simulation when $n = 10$. It can be seen that the SD and the HIS variant detector (when $D = 2$ and $C = 4$) outperform the SDP detector as well as the MMSE detector. In fact, the required SNR for a BER of 10^{-4} is 2.5 dB lower than that of the SDP detector.

Let's simulate high-dimensional systems $n = 40$ or $n = 60$. This is a very hard instance of the detection problem and is therefore a good benchmark for comparing the performance of various detection algorithms. The performance of the proposed GDGI detector variants are shown in Figure 6 and Figure 7. For $n = 40$, the results given in Figure 6 indicate that the three GDGI variants (HIS, BIS, and OIS), when $D = 4$ and $C = 4$ have almost identical performance and obtain about 0.6 dB performance gain compared with SDP scheme at BER 10^{-4} . Figure 7 illustrates the result in the case of $n = 60$. The OIS and the HIS schemes are 0.3 dB from the optimal SD and 0.6 dB better than the SDP detector (given in [25]) at BER 10^{-4} .

In second experiment, we present the performance of HIS, SD (sphere decoding) and other known detection algorithms for a downlink MC-CDMA system Figure 8. The channel coefficients are modified for each transmitted symbol. All users have the same power. We assume the power control being perfect, *i.e.*, at each time i , the received symbol power is equal to the transmitted symbol power. Each user symbol is spread over $L_c = N_p = 16$, where N_p is the number of sub-carriers and L_c is the length of spreading code, with a real Walsh-Hadamard sequence. Figure 8 shows the performance of the HIS variant in a Rayleigh fading channel for a fully loaded downlink MC-CDMA system with $N_u = 16$ users and employing an uncoded 4-QAM modulation. It can be seen that the performance of the proposed detector is excellent, using

$D = 4$ and $C = 4$. In fact, the required SNR for a BER of 10^{-3} is 0.8 dB lower than that of the SDP detector.

B. GDGI Parameters Impact for the OIS version for $n = 10$

The impact of the parameters D and C on the performance is studied. Table I shows the SNR difference between the optimal decoder (the SD) and the OIS variant for a BER= 10^{-4} when D and C both vary from 1 to 4. It can be observed that the performance increases both with C and D as expected.

V. COMPUTATIONAL COMPLEXITY AND IMPLEMENTATION STRUCTURE

In this section, the computational complexity model of the GDGI is proposed according to the parameters D , C and the type of variants (HIS, BIS and OIS). Based on [21]), we assign a relative cost for each operation: 1 for one addition, 5 for one square, 10 for one multiplication, and 40 for one division. We assume the channel matrix to be static over a sufficiently long period of time, so that the computational complexity of any preprocessing step (SVD decomposition and pseudo-inverse of the channel matrix) is negligible. The different levels of parallelism of GDGI algorithm are also described.

In the following, the variable Nb indicate the cardinality of set \mathcal{S} : $Nb = Nb_{\mathcal{H}} = 2n$ in case of HIS method, $Nb = Nb_{\mathcal{B}} = n$ in case of BIS method, and $Nb = Nb_{\mathcal{O}} = l + 1$ in case of OIS method.

A. Computational Complexity

We concentrate here on processing part of GDGI detector. To find the point \mathbf{x}_0 , the received signal vector will be multiplied by the pseudo-inverse of channel matrix. The resulting complexity for a single transmitted vector is hence: n^2 additions and n^2 multiplications. For the D studied directions, the diversification comes down to finding all intersection points \mathcal{I}_k , $k = 1..D$. The total complexity of this phase would be:

- $Dn^2 + NbDn$ additions and Dn^2 multiplications for HIS method.
- Dn^2 additions and Dn^2 multiplications for BIS method.

- $n(2 + Nb)D$ additions, $2nD$ multiplications, and D divisions for OIS method, where $Nb = \bar{l} + 1$, where \bar{l} indicate the average number of non zero coordinate of $\mathbf{n}_k = Q(\mathbf{v}_k)$, see the equation (16).

Given all subset $\mathcal{I}_k \subset \{\pm 1\}^n$, $k = 1..D$, the evaluation step needs $(n + n^2 Nb)D$ additions and $nNbD$ squares. For each studied direction \mathbf{v}_k , the GDGI detector sort in ascending order and select C best points having the minimum objective function values, then for the D direction we need $CDNb$ comparison (i.e. additions). The final step of the presented detector is the intensification step over CD starting points. The computational complexity of the above step is hence: $2CD\theta n^2$ additions and $CD\theta n^2$ squares.

For a given D and C , the computational complexity of the GDGI detector is almost constant, over the entire SNR range, compared to that of the sphere decoding⁵.

The impact of the parameters D and C on the complexity of the OIS variant is shown in Table II for $n = 10$. The association of table II and I allows to tune optimally the tradeoff between performance and complexity of the GDGI decoder.

B. Implementation architecture

The proposed detector is more simple than existing iterative suboptimal detectors, and offer improved performance and is not too complex to be feasible for hardware implementation. Moreover, compared to other high performance algorithms such as SDP approach and SD, the level of complexity is much simpler. The SDP requires expensive hardware iterative structure [19], and the SD algorithm requires a large amount of calculation in the preprocessing and searching stages [20]. The advantage of our approach is that its complexity is fixed and easily adjustable, at the expense of a possible performance penalty.

Among various sub-optimal detectors, the GDGI detector is particularly attractive for a hardware implementation due to its inherent parallel structure. In fact, the D directions can be processed in parallel, and in each direction each point of the list \mathcal{I}_k can be determined and evaluated in parallel. Then, once ξ_{start} is obtained, the coordinate descent-1 algorithm can also be performed in parallel.

⁵for example, for $n = 60$, \bar{l} varies from 17.5 (SNR = 0 dB) downto 14.9 (SNR = 12 dB) and θ varies from 5.3 (SNR = 0 dB) downto 3.8 (SNR = 12 dB)

The architecture proposed for implementing the GDGI detector is seen in Figure 4. The Pre-Processing step is used to compute the SVD decomposition of the channel matrix as well as calculate ZF solution \mathbf{x}_0 . This step generates the D smallest right singular vectors $\{\mathbf{v}_k\}_{k=1}^D$ and the pseudo-inverse matrix \mathbf{H}^+ , and it can be implemented with a digital signal processing (DSP) device instead of systolic FPGA architecture. For a given received symbol (y), $\mathbf{x}_0 = \mathbf{H}^+ \mathbf{y}$ is first computed. The geometrical intersection (GI) module creates the subset $\mathcal{I}_k \subset \{\pm 1\}^n$ using the ZF solution and the k^{th} smallest right singular vector. Evaluation (EVA) module computes the value of the objective function, $f(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2$, where \mathbf{x} belongs to the subset \mathcal{I}_k . SORT module sorts the results of the EVA module in ascending order and select C best points having the minimum distances. Finally, intensification (INT) module performs the coordinate descent-1 algorithm over the C selected feasible points.

VI. CONCLUSION

A new method is presented to solve the ML problem in a quasi optimal way for the case of a constellation $\xi^n = \{\pm 1\}^n$ and a square channel matrix \mathbf{H} of dimension n . This method, named GDGI, is based on common tools in the domain of heuristic techniques: diversification, using geometrical properties of the channel matrix \mathbf{H} and intensification using the greedy algorithm. The two new variants of the Geometrical Diversification presented are: HIS and OIS. All variants, HIS, BIS and OIS exhibit almost optimal performance. For example, for a square matrix \mathbf{H} of size 60×60 , GDGI performs only 0.6 dB lower than the optimal SD solution at the BER of 10^{-5} . Compare to the BIS, the HIS has a higher complexity and similar performance. In the case where $\xi = \{\pm 1\}$, BIS is preferable to HIS.

The superiority of BIS over HIS is still an open issue in the cases where $\xi = \{-3, -1, 1, 3\}$ (MAQ-16 constellation) and for higher order constellations. In fact, the HIS method can be naturally extended to this type of constellations, while it is not the case for the BIS. Compare to the BIS, the new OIS method offers significant reduction of computation, since the number of multiplications of the geometrical diversification part is reduced from $2n^2$ to $2n$ while number of additions remain the same. Moreover, the cardinality of the subset \mathcal{I} is significantly reduced (by a factor greater than 3 for $n = 60$). It is also interesting to note that the trade-off performance/complexity can be finely tuned by setting the C and D parameters. In addition to this, the GDGI is very well suited for hardware implementation: its complexity is almost

constant for all SNR and many level of parallelism can be exploited (the number D of direction, the computation of the intersection, the evaluation of the point in (I) , the realization of the GI) to pipeline the architecture and obtain very efficient, high throughput decoding receiver.

As a future work, we propose to carry on some theoretical studies to understand and optimize the diversification process in a better way and to generalize it in the case of higher dimensional constellations. We also plan to develop the GDGI algorithm on a FPGA architecture.

ACKNOWLEDGMENT

This research work has been supported by the PALMYRE CPER project funded by the Brittany region, the Morbihan Department and the Lorient Agglomeration Community. The authors would also like to thank the anonymous reviewers whose comments and suggestions have greatly helped in improving the quality of the original manuscript.

REFERENCES

- [1] W. H. Mow, "Maximum Likelihood Sequence Estimation from the Lattice viewpoint," *IEEE Trans. Inf. Theory*, vol. 40, no. 5, pp. 1591-1600, Sep 1994.
- [2] S. Verdú, *Multiuser Detection*, Cambridge University Press, 1998.
- [3] U. Fincke and M. Pohst, "Improved Methods for Calculating Vectors of Short Length in a lattice, Including a Complexity Analysis," *Math. Comput.*, vol. 44, pp. 463-471, Apr 1985.
- [4] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1639-1642, Jul 1999.
- [5] P. H. Tan, and L. K. Rasmussen, "The application of semidefinite programming for detection in CDMA," *IEEE J. Sel. Areas in Commun.*, vol. 19, no. 8, pp. 1442-1449, Aug 2001.
- [6] W. Ma, T. N. Davidson, K. M. Wong, Z. Lou, and P. Ching, "Quasi maximum likelihood multiuser detection using semidefinite relaxation with application to synchronous CDMA," *IEEE Trans. Signal Process.*, vol. 50, no. 4, pp. 912-922, Apr 2002.
- [7] M. Abdi, H. Nahas, A. Jard, and E. Moulines, "Semidefinite relaxation of the maximum likelihood criterion applied to multiuser detection in a CDMA context," *IEEE Signal Process. Lett.*, vol. 9, no. 6, pp. 165-167, Jun 2002.
- [8] H. Artes and D. Seethaler, "Efficient Detection Algorithms for MIMO Channels: A Geometrical Approach to Approximate ML Detection," *IEEE Trans. Signal Process.*, vol. 51, no. 11, pp. 2808 - 2820, Nov 2003.
- [9] V. J. Rayward-Smith, I. H. Osman, C. R. Reeves, and G. D. Smith, *Modern Heuristic Search Methods*, Wiley, 1996.
- [10] P. Patel and J. M. Holtzman, "Analysis of a Simple Successive Interference Cancellation Scheme in DS/CDMA Systems," *IEEE J. Sel. Areas in Commun.*, Special Issue on CDMA, vol. 12, pp. 796-807, Jun 1994.
- [11] M. K. Varanasi and B. Aazhang, "Multistage detection in asynchronous code division multiple-access communications," *IEEE Trans. Commun.*, vol. 38, no. 4, pp. 509-519, Apr 1990.
- [12] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474-1484, Apr 2005.

- [13] P. Spasojevic and C. N. Georghiades, "The slowest descent method and its application to sequence estimation," *IEEE Trans. Commun.*, vol. 49, no. 9, pp. 1592-1604, Sep 2001.
- [14] A. Burg, N. Felber, and W. Fichtner, "A 50 Mbps Maximum Likelihood Decoder for Multiple-Input Multiple-Output Systems with QPSK Modulation," presented at the ICECS 2003, Sharjah, U.A.E.
- [15] P. Hui T.L. Rasmussen, "Multiuser detection in CDMA - a comparison of relaxations, exact and heuristic search methods," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1802-1809, Sept 2004.
- [16] J. Luo, G. Levchuk, K. Pattipati and P. Willett, "A Class of Coordinate Descent Methods for Multiuser Detection," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, vol. 5, 2000, pp. 2853-2856.
- [17] P. H. Tan and L. K. Rasmussen, "Tabu search multiuser detection in CDMA," *Radio Sci. and Comm. Conf.*, Stockholm, Sweden, pp. 744-748, Jun 2002.
- [18] A. Nafkha, E. Boutillon, C. Roland, "Near Maximum Likelihood Detection for MIMO systems using an intensification strategy over a BCH codes", presented at the SSD, Hammamet, Tunisie, Mar. 2007.
- [19] S.Chabouh, P.Ciblat, J.C.Belfiore, "Iterative algorithms for SDP relaxation associated with MIMO ML detection problem," presented at the ISIT, Parma, Italy, Oct. 2004.
- [20] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI Implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Sol. State Cir.*, vol. 40, no. 7, pp. 1566-1576, Jul. 2005.
- [21] Nortel Networks, *R1-031085: Complexity Comparison of OFDM HS-DSCH Receivers and Advanced Receivers for HSDPA and Associated Text Proposal*, 3GPP TSG-RAN-1, Seoul, Korea. Oct 2003.
- [22] D. Garrett, L. Davis, S. ten Brink, B. Hochwald, and G. Knagge, "Silicon complexity for maximum likelihood MIMO detection using spherical decoding," *IEEE J. Sol. State Cir.*, vol. 39, no. 9, pp. 1544-1552, Sep. 2004.
- [23] B. Widdup, G. Woodward, and G. Knagge, "A highly-parallel VLSI architecture for a list sphere detector," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2004, vol. 5, pp 2720-2725.
- [24] Krukowski, A., I. Kale, K. Hejn and G. D. Cain, "A "bit-flipping" approach to multistage two-path decimation filter design of two coefficients", presented at the Second Int. Symp. DSP Commun. Sys, SPRI, South Australia, Apr. 1994.
- [25] http://www.ece.umn.edu/users/luozq/software/sw_simulations.html

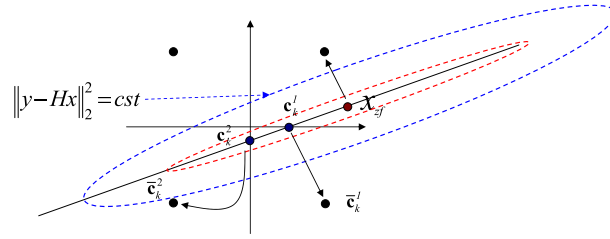


Fig. 2. Basis intersection method : One to one mapping from $\{\mathbf{x}_0, \mathbf{c}_k^1, \dots, \mathbf{c}_k^n\}$ to $\{\text{sign}(\mathbf{x}_0), \bar{\mathbf{c}}_k^1, \dots, \bar{\mathbf{c}}_k^n\}$ for $n = 2$. Each intersection point \mathbf{c}_k^i is at equal distance from its two neighboring candidate points. $\bar{\mathbf{c}}_k^i$ is chosen to be one of these two candidate points that is on the opposite side of the i^{th} coordinate hyper-plane with respect to $\text{sign}(\mathbf{x}_0)$.

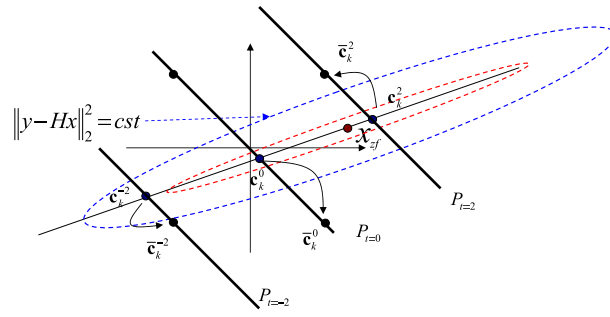


Fig. 3. Orthogonal intersection method: One to one mapping from $\{\mathbf{c}^t\}_{t=-l, -l+2, \dots, l-2, l}$ to $\{\bar{\mathbf{c}}^t\}_{t=-l, -l+2, \dots, l-2, l}$ for $n = 2$.

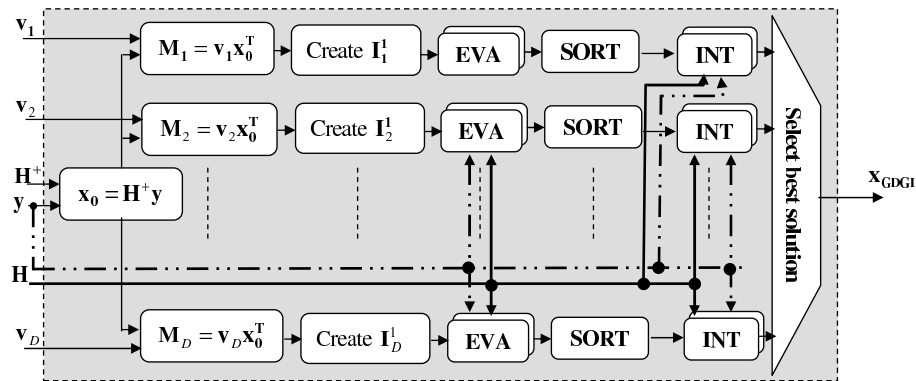


Fig. 4. Block diagram of the GDGI detector.

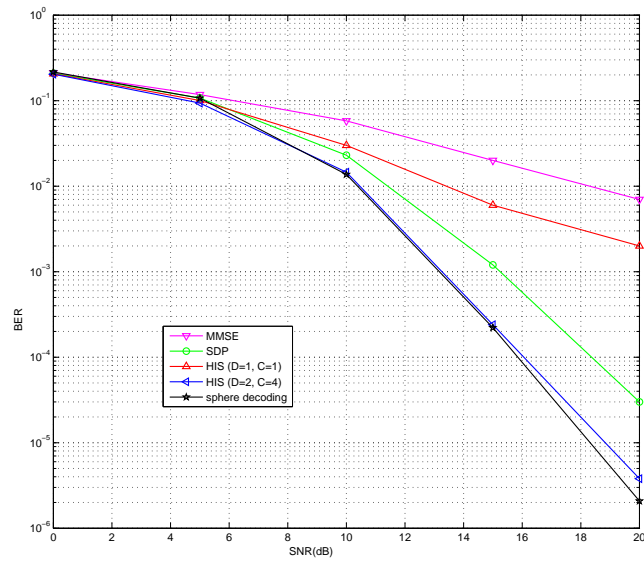


Fig. 5. BER versus SNR for $n = 10$, comparison of HIS variant, sphere decoding, SDP and MMSE detectors.

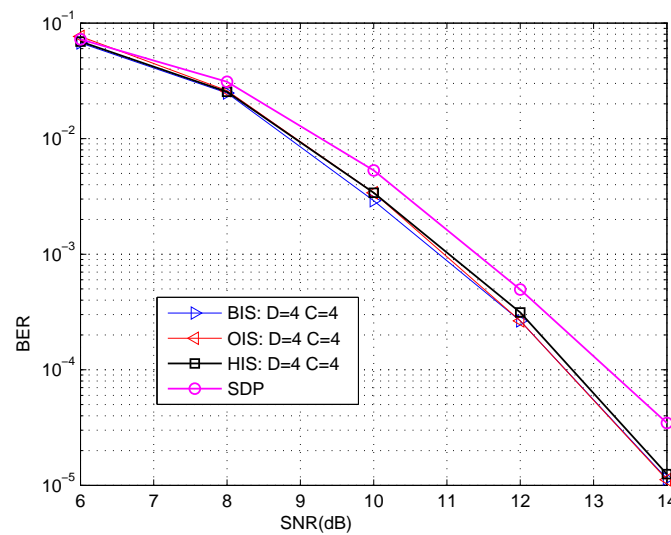


Fig. 6. BER versus SNR for $n = 40$, comparison of GDGI variants (HIS, OIS and BIS) detectors and SDP detector [25].

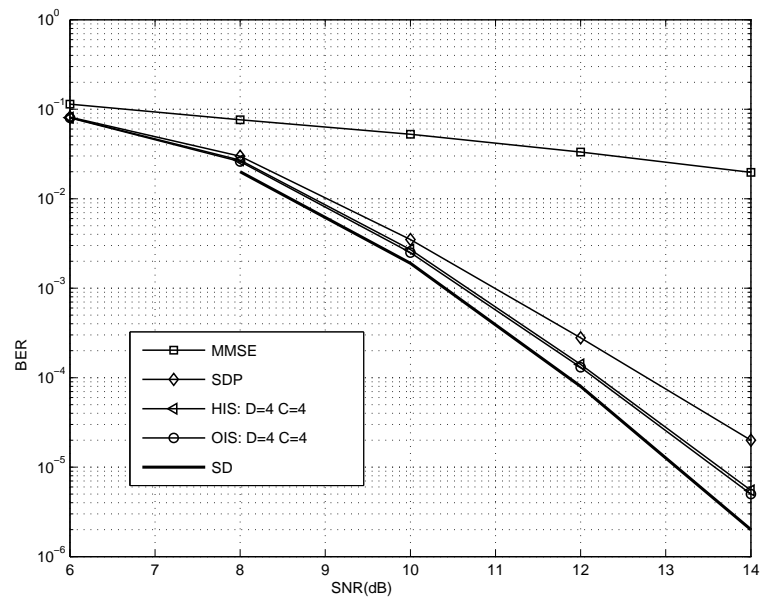


Fig. 7. Performance of GDGI (OIS and HIS) detectors, SDP [25], SD (sphere decoding) [25], $n = 60$, and perfect channel estimation.

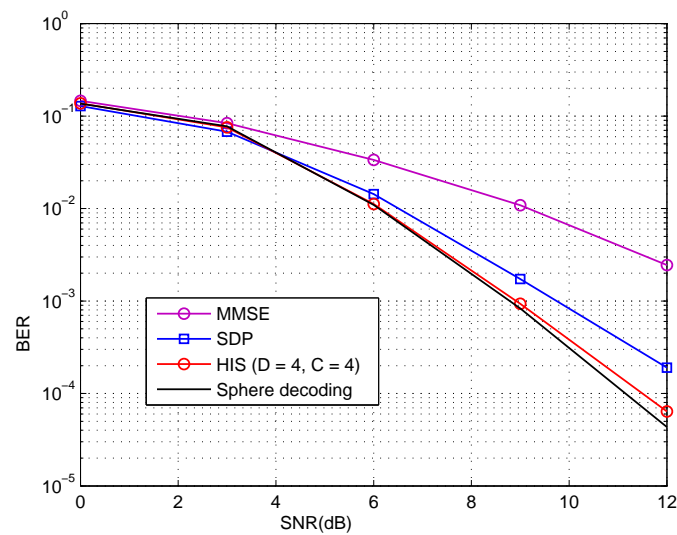


Fig. 8. Comparison between SD detection, HIS variant and others sub-optimum detectors in case of MC-CDMA system ($N_u = N_p = L_c = 16$)