

# Complexity of ML lattice decoders for the decoding of linear full rate Space-Time Codes

Ghaya Rekaya, *Student Member IEEE* and Jean-Claude Belfiore, *Member IEEE*

**Abstract**—We propose here to compare, in terms of complexity, two ML decoding algorithms, the Sphere-Decoder and the Schnorr-Euchner decoder, when used to decode multi-antenna transmission schemes, using full rate space-time codes, over a Rayleigh fading channel.

**Index Terms**—Space-time codes, lattice, multi-antenna, Maximum-Likelihood decoding

## I. INTRODUCTION

IN this work we are interested in the decoding of multi-antenna systems using full rate space-time codes, over a Rayleigh fading channel by lattice decoders, the Sphere-Decoder (SD) and the Schnorr-Euchner decoder (SE). In [1], a lattice representation of multi-antenna schemes is presented. Our interest in these decoders comes from the fact that they achieve ML (Maximum Likelihood) performance with reasonable complexity. In this work we will study and compare both algorithms with the aim of choosing the appropriate decoder according to transmission parameters as the Signal to Noise Ratio (SNR) and the number of antennas. We note that in [2], Agrell *et al.* made a comparison of the SD and the SE when used to decode infinite lattices, which is not our case because we use finite lattice constellations.

In the second part we present uncoded and linearly coded multi-antenna transmission schemes. In the third part we study the two algorithms, and we modify them to decode lattice constellations. In the fourth part we study and compare the complexities of the SD and the SE by using analytical and simulation results.

Ghaya Rekaya (rekaya@enst.fr) and J.-C. Belfiore (belfiore@enst.fr) are with École Nationale Supérieure des Télécommunications (ENST), 46 rue Barrault, 75013 Paris FRANCE

This work was supported by a CNRS grant

## II. LATTICE REPRESENTATION OF MULTI-ANTENNA SCHEME

### A. Uncoded system

We consider a system with  $M$  transmit antennas and  $N$  receive antennas, that we suppose synchronous. The channel is assumed to be quasi-static (block fading channel). The received signal at each instant time is thus given by:

$$\mathbf{r} = \mathbf{H} \cdot \mathbf{a}^T + \mathbf{w} \quad (1)$$

where  $\mathbf{H}$  is the  $N \times M$  channel transfer matrix with entries  $h_{kj}$ , that is the fading between transmitter antenna  $j$  and receiver antenna  $k$ .  $\mathbf{H}$  is modeled by independent Gaussian random variables of variance 0.5 per dimension.  $\mathbf{a} = (a_1, a_2, \dots, a_M)$  denotes the modulated transmitted vector. We use  $q$ -QAM constellation, with  $q = 4, 16, \dots$ , and the average energy per bit is fixed to  $E_b = 1$ .  $\mathbf{w}$  represents the Additive White Gaussian Noise (AWGN) vector. It is an  $N \times 1$  complex vector component-wise independent with variance  $\sigma^2$  per dimension, where  $\sigma^2$  is adjusted by  $\sigma^2 = (M \cdot E_{s_{av}} / 2 \log(q)) \cdot 10^{(-SNR_{dB}/10)}$ ,  $E_{s_{av}}$  is the average symbol energy of the  $q$ -QAM constellation.

The independence of the receive antennas and the affecting fades of each sub-stream transmitted by each antenna presupposes that the transfer matrix  $\mathbf{H}$  is full rank with probability 1, i.e the event of having two or more dependent columns in  $\mathbf{H}$  is negligible with respect to the probability measure [1].

We will, in the following, usually use systems with an equal number of transmit and receive antennas, i.e.  $M = N$  a representation of the multi-antenna transmission scheme by a lattice packing was given, the system being written as :

$$\begin{aligned}
\mathbf{r}_{\mathbb{R}} &= [\Re(\mathbf{r}^T) \Im(\mathbf{r}^T)] \\
&= \mathbf{a}_{\mathbb{R}} \cdot \begin{pmatrix} \Re(\mathbf{H}^T) & \Im(\mathbf{H}^T) \\ -\Im(\mathbf{H}^T) & \Re(\mathbf{H}^T) \end{pmatrix} + \mathbf{w}_{\mathbb{R}} \\
&= \mathbf{a}_{\mathbb{R}} \cdot \mathbf{H}_{\mathbb{R}} + \mathbf{w}_{\mathbb{R}}
\end{aligned}$$

where  $\mathbf{a}_{\mathbb{R}} = [\Re(\mathbf{a}) \Im(\mathbf{a})] \in \mathbb{Z}^{2M}$ ,  $\mathbb{Z}$  is the ring of integers ( $\mathbf{v}_{\mathbb{R}} = [\Re(\mathbf{v}^T) \Im(\mathbf{v}^T)] \in \mathbb{R}^{2M}$ ,  $\mathbb{R}$  is the field of real numbers,  $\Re[\mathbf{v}]$  and  $\Im[\mathbf{v}]$  denotes respectively the real and the imaginary part of the vector  $\mathbf{v}$ ). The dimension of the equivalent lattice is  $2M$ .

Note that the rank of the matrix  $\mathbf{H}_{\mathbb{R}}$  is almost always  $2M$ , and its Gram matrix  $\mathbf{G}_{\mathbf{H}} = \mathbf{H}_{\mathbb{R}} \mathbf{H}_{\mathbb{R}}^T$ , is positive definite. Considering therefore this new representation of the multi-antenna scheme, we can apply the universal lattice decoders, like Sphere Decoder and Schnorr-Euchner decoder, to decode such systems.

### B. Coded system

We now encode the system with a linear Space Time Code (STC). We will use the full rate, fully diverse STC codes presented in [3]. Under another form, these codes have been generalized in [4]. The received signal matrix is then written as :

$$\mathbf{R} = \frac{1}{\sqrt{M}} \cdot \mathbf{H} \cdot \mathbf{B} + \mathbf{W} \quad (2)$$

where  $M$  is the number of transmit antennas,  $N$  is the number of receive antennas and  $T$  is the temporal code-length.  $\mathbf{R}$  is the  $N \times T$  received matrix,  $\mathbf{H}$  is the  $N \times M$  channel transfer matrix,  $\mathbf{W}$  is  $N \times T$  noise matrix, and  $\mathbf{B}$  is  $M \times T$  code word matrix. The expression of the  $\mathbf{B}$  elements as a function of the transmitted symbols is given in [3]. We rewrite (2) to have a simpler equation to consider, we obtain (3), where  $\mathbf{r}$  and  $\mathbf{w}$  are  $N \cdot T$  vectors, obtained by concatenating the columns of respectively matrices  $\mathbf{R}$  and  $\mathbf{W}$  (it is equivalent to the vectorization of (2)). Vector  $\mathbf{a}$  is the vector of the transmitted symbols and the matrix  $\phi$  is deduced from matrix  $\mathbf{B}$  (See ref. [5] for the 2 antenna case) as :

$$\mathbf{r} = \frac{1}{\sqrt{M}} \cdot \mathbf{H}' \cdot \phi \cdot \mathbf{a}^T + \mathbf{w} \quad (3)$$

We notice that when using  $M$  antennas in the real transmission scheme, the equivalent number of complex dimensions in the transmission scheme model

described by (3) is  $M^2$ . We therefore have to work with a high number of antennas, even at low SNRs.

Let us now separate imaginary and real parts of each vector and matrix component, to construct the lattice representation of the system in (4).

$$\mathbf{r}_{\mathbb{R}} = \frac{1}{\sqrt{M}} \mathbf{H}'_{\mathbb{R}} \phi_{\mathbb{R}} \mathbf{a}_{\mathbb{R}} + \mathbf{w}_{\mathbb{R}} \quad (4)$$

developed in the following equation,

$$\mathbf{r}_{\mathbb{R}} = \begin{bmatrix} \Re(\mathbf{H}) & & & -\Im(\mathbf{H}) & & \\ & \ddots & & & \ddots & \\ & & \Re(\mathbf{H}) & & & -\Im(\mathbf{H}) \\ \Im(\mathbf{H}) & & & \Re(\mathbf{H}) & & \\ & \ddots & & & \ddots & \\ & & \Im(\mathbf{H}) & & & \Re(\mathbf{H}) \end{bmatrix} \cdot \begin{bmatrix} \Re(\phi) & -\Im(\phi) \\ \Im(\phi) & \Re(\phi) \end{bmatrix} \cdot \begin{bmatrix} \Re(\mathbf{a}) \\ \Im(\mathbf{a}) \end{bmatrix} + \begin{bmatrix} \Re(\mathbf{w}) \\ \Im(\mathbf{w}) \end{bmatrix} \quad (5)$$

By defining  $\mathfrak{H} = \frac{1}{\sqrt{M}} \cdot \mathbf{H}'_{\mathbb{R}} \phi_{\mathbb{R}}$ , and rewriting equation(4),

$$\mathbf{r}_{\mathbb{R}} = \mathfrak{H} \cdot \mathbf{a}_{\mathbb{R}} + \mathbf{w}_{\mathbb{R}} \quad (6)$$

we obtain the lattice representation, where  $\mathfrak{H}$  is the generator matrix of the lattice, *i.e.* a  $2M^2 \times 2M^2$  real matrix. The multi-antenna coded system represented in equation (6) can therefore be decoded by lattice decoders.

## III. LATTICE DECODERS

The SD and the SE are ML decoding algorithms for lattice codes, when used over an independent flat fading channel, with channel state information known at the receiver. The SD and SE algorithms were used to decode infinite lattices. In the following paragraphs, we will present modifications to both algorithms to decode finite lattice constellations. We note that in [6] the authors have mentioned this idea without going into detail.

### A. Sphere-decoder (SD)

The sphere decoder, also known as Viterbo-Boutros algorithm, was presented in [7] for the Gaussian channel, then it was extended in [6] for the Rayleigh fading channel case, and finally the case of MIMO channels was introduced in [1] and in [8]. The algorithm searches for the closest point among

all lattice points inside a sphere of a given radius centered at the received point. The closest point searching method consists in calculating minimum and maximum bounds for each vector component, and checking all lattice points inside the sphere by scaling increasingly each vector component interval. If no lattice point is found in the sphere, then, the sphere radius is increased and the search is started again.

The algorithm given in [6] is available when infinite lattices are used, but in our transmission scheme the information symbols belong to QAM constellations. Therefore there is no longer an infinite  $\mathbb{Z}$ -lattice but a finite lattice constellation, which means a finite subset of a lattice. Consequently the vector found by the SD must imperatively belong to this constellation.

There are two ways to proceed :

- 1) Checking over the whole lattice, and keeping only vectors belonging to the constellation.
- 2) Seeking lattice constellation directly by checking only vectors belonging to the constellation.

The first method is more expensive than the second in terms of operations. In our simulations we apply the second method, whose flow chart is presented in figure 4. The notation used in the SD flow chart is in conformity with that used in [6]. The function  $rint(x)$  returns the closest integer to  $x$ , which belongs to the constellation .

At initialization of the SD, the sphere radius is fixed. We note that for small SNRs, a large radius is needed initially. However, for large SNRs, a small radius is sufficient since the point to be detected is closed to the received point. For this reason, taking initially a small radius limits the search time. The improvement made to the algorithm consists in calculating the sphere radius according to the SNR. This idea was presented in [9] and [10]. In [9], the given formula to calculate the radius is as follows:

$$BLAST_{radius} r^2 = 2 \cdot \sigma^2 \cdot N - \mathbf{r}^H \cdot (\mathbf{I} - \mathbf{H} \cdot (\mathbf{H}^H \cdot \mathbf{H})^{-1} \cdot \mathbf{H}^{-1}) \cdot \mathbf{r} \quad (7)$$

where  $N$  represents the lattice dimension. Unfortunately this formula is not adapted to our application, since  $\mathbf{H}$ , the transfer matrix of the channel is invertible, which means that the radius depends only on the SNR and not on the channel. Therefore we use the formula presented in [10], which is :

$$r^2 = 2 \cdot \sigma^2 \cdot N \quad (8)$$

Simulation results confirm the need to adapt the radius according to the SNR, considering the gain in number of operations. The results of figure 1 were obtained by simulating the uncoded system described in section 2; 4 antennas were used at the transmission and the reception. We have represented the number of multiplications used by the SD until convergence as a function of the SNR. The three curves correspond to an adapted radius, to an initial radius of 1 and to an initial radius of 2 respectively. We notice that by using an adapted radius, we obtain the smallest number of operations especially at small SNRs, for example at 4dB, we have 500 operations less than the other cases. This improvement of the SD will obviously improve the total complexity of the SD.

### B. Schnorr-Euchner Decoder (SE)

The Schnorr-Euchner algorithm we are studying here was presented in [2]. It was used in cryptography applications. This algorithm has the same principle as the SD : the search for the closest point.

This algorithm is based on two stages. The first stage consists in searching for the ‘‘Babai point’’ (BP), which represents a first estimation, but is not necessarily, the closest point. Finding the BP gives us a bound on the error. In the second stage, we modify the BP until the closest point is reached. We zigzag around each BP component in turn to build the closest point (unlike the sphere decoder, there is no minimum and maximum bound for each BP component). The time needed to find the closest point is closely related to BP, which means closely related to the SNR. In fact, if the BP is very far from the closest point, i.e for low SNRs, the algorithm takes much more time to converge. However, if the BP is close to the closest point, i.e for high SNRs, the algorithm converges rapidly.

The algorithm presented in [2] uses an infinite lattice, which is not our case since we use a lattice constellation. Therefore, similarly to the SD, we modified the SE algorithm to consider only vectors that belong to the constellation. Our first choice to carry out this modification was to scale only vectors that belong to the constellation. Unfortunately, this always results in an infinite loop, or an incorrect result. This can be explained by the fact that by modifying

the way of scaling to seek only constellation points, the right point is lost and is never refound.

For all these reasons we adopt a searching method which can go beyond the constellation, but only keeps those belonging to the constellation. The flow chart of this version of the algorithm is presented in figure 5. The notation used in the SE flow chart is in conformity with that used in [2]. The function  $rint(x)$  returns the closest integer to  $x$ , which belong to the constellation .

#### IV. COMPARISON OF THE SD AND THE SE

Both SD and SE are ML decoders, which enables us to conclude that the two algorithms perform well. This was proved in [1] where the SD was used to decode uncoded multi-antenna schemes. The two algorithms have the same principle, the search for the closest point, but differ mainly in the search method. In the following we will compare the complexities of these two algorithms.

Since the multiplications are the most expensive operations in terms of machine cycles compared to addition and comparison, only multiplications will be taken in account to measure the complexity. The complexity of the algorithm is defined by the number of multiplications carried out until convergence.

However both algorithms, before attacking the closest point searching phase, need a preparation phase, which we will qualify by pre-decoding phase, and also an initialization phase (see SD and SE flow charts in figures 4 and 5). To study the complexity of both algorithms, it is worth studying and comparing first their respective pre-decoding and initialization phases, and subsequently their respective closest point searching method. Finally we will compare their respective total complexities .

##### A. Comparison of pre-decoding and initialization phases

As shown on the flow-charts of the SD and the SE, in pre-decoding and initialization phases we have essentially two operations : the first one consists in the calculation of a triangular form of the matrix  $\mathbf{H}$ . For that we can use either QR decomposition or Cholesky decomposition. The second one consists in the calculation of the Zero Forcing point (ZF).

When using QR decomposition, we decompose  $\mathbf{H}^T$ , and then we define  $\mathbf{G} = \mathbf{R}^T$ , where  $\mathbf{G}$  is

a lower triangular matrix, this needs  $\frac{2}{3}N^3$  operations. When using Cholesky decomposition, we have first to calculate the Gram matrix of  $\mathbf{H}$ , defined as  $\mathbf{Gram} = \mathbf{H}\mathbf{H}^T$ , and so we decompose  $\mathbf{Gram}$  to obtain  $\mathbf{Gram} = \mathbf{U}^T\mathbf{U}$ , where  $\mathbf{U}$  is an upper triangular matrix, and then  $\mathbf{G} = \mathbf{U}^T$ . The total number of operations needed is  $\frac{7}{6}N^3 + N$ . By comparing the number of operations needed for each decomposition, we remark immediately that the QR decomposition is less expensive in terms of operations.

For the second operation, we need first to calculate the inverse of the transfer matrix of the channel  $\mathbf{H}$ . The Zero Forcing point is defined by the equation (9), where  $\mathbf{r}$  is the received vector. We must note that the ‘‘Babai Point’’ evoked in [2] is the same as the ZF point.

$$\boldsymbol{\rho} = \mathbf{r}.\mathbf{H}^{-1} = (\mathbf{a}.\mathbf{H} + \mathbf{w}).\mathbf{H}^{-1} \quad (9)$$

For the SD, we will use matrix  $\mathbf{G}$  to build matrix  $\mathbf{Q}$ , as defined in [6]. Using the matrix  $\mathbf{Q}$  and the ZF point we calculate the minimum and maximum bound of each closest point component. For the SE also, we do not need the matrix  $\mathbf{G}$  but its inverse. For the SE, the ZF point represents the first point found by the algorithm which will be adjusted in the following to obtain the closest point.

The total number of multiplications necessary to carry out the pre-decoding and the initialization phases using QR decomposition for the SD and the SE respectively are :  $\frac{5}{3}N^3 + \frac{3}{2}N^2$ ,  $\frac{11}{6}N^3 + \frac{3}{2}N^2 + \frac{1}{3}N$ .

We remark immediately that the pre-decoding and initialization phases of the SE are heavier than that of the SD. In fact the SE uses  $\frac{5}{6}N^3 + \frac{1}{3}N$  multiplications more than the SD. How critical this disadvantage is depends on the lattice dimension  $N$ .

In fact, for small lattice dimensions, the number of multiplications in the pre-decoding phase is of the same order of magnitude as that of the searching phase, so the pre-decoding phase has an influence on the total complexity of both algorithms. This influence is more significant for fast fading channels, where the pre-decoding phase are made more frequently. For large lattice dimensions, the number of multiplications in the pre-decoding phase is very small compared to those in the searching phase, and we can say that the pre-decoding phase doesn’t influence the total complexity of the algorithm.

### B. Comparison of searching phases

Let's compare now the respective closest point searching methods of the SD and the SE. Unlike the pre-decoding phase, trying to build an exact expression of the number of multiplications is meaningless, since the number of loops in the algorithm is totally random. Hence to calculate the number of multiplications we have simulated the uncoded and coded schemes described in section 2 using as a decoder both the SD and the SE. We have counted the number of multiplications of each algorithm in the searching phase until convergence.

In figure 2, we simulate an uncoded multi-antenna scheme at an SNR of 15 dB, the number of multiplications in the searching phase of SD and SE respectively is plotted as a function of the number of antennas. We can distinguish three parts in the curves:

- for less than 5 antennas : the two algorithms have almost the same complexity
- between 6 and 9 antennas : the SE has at least 1000 multiplications less than the SD
- for more than 9 antennas : the SD outperforms the SE.

In figure 3, we simulate a coded multi-antenna scheme at an SNR of 5 dB, the ratio of the number of multiplications in the searching phase of SE by SD is plotted as a function of the number of antennas. For 2 antennas, the SE needs one and a half times more multiplications than the SD, which represents a little advantage for the SE thus can be exploited.

For more than 2 antennas, the SD outperforms the SE, for example for 3 antennas, which correspond to  $9 \times 2$  antennas in the simulated scheme described by (6), the SD has 8 times more multiplications than the SE.

In conclusion, we can say that by considering only the searching phase, simulation results of the uncoded and the coded schemes have confirmed that for a high number of antennas the SD outperforms the SE. And we note that for a low number of antennas the SE has an advance compared to the SD, which is good for systems using slowly fading channels.

### C. Comparison of the total complexity

As we consider a transmission scheme using a block fading channel, the pre-decoding phase is necessary for each codeword and consequently the complexity of this phase will affect the total complexity.

To evaluate the total complexity of both algorithms we have added the number of multiplications of the three phases, pre-decoding, initialization and searching.

In figure 2 we have also plotted the total number of multiplications of the SD and the SE respectively as a function of the number of antennas. We see that for less than 9 antennas, both algorithms have almost the same complexity. We can say therefore that there was a compensation between SD's advantage in the pre-decoding phase and SE's advantage in the searching phase. For more than 9 antennas, SD is less complex than SE, and we can say that for a high number of antennas the searching phase complexity has a strong influence on the total complexity.

In figure 3 we have also plotted the ratio of the total number of multiplications SE by SD as a function of the number of antennas. We remark that the two curves are close, and we conclude that for coded schemes, or systems with a high number of antennas, the SD outperforms the SE. This result is foreseeable given the conclusions of the two previous paragraphs.

## V. CONCLUSION

We have studied the complexities of the Sphere decoder and the Schnorr-Euchner decoder, when used to decode uncoded and linearly coded multi-antenna transmission schemes, over a Rayleigh fading channel. The complexity of both algorithms is defined as the number of multiplications carried out until convergence.

Our study has shown that the SE has heavier initialization and pre-decoding phases than the SD, and conversely, the search phase of the SE is less complex than that of the SD.

To conclude on the total complexity of both decoders when used to decode our transmission scheme, we have to distinguish two cases. For a small number of antennas, both stages compensate each other in terms of complexity, and consequently the total complexities of the two algorithms are too close with a little advantage for the SE. For a high number of antennas, the complexity of the searching stage has a stronger influence on the total complexity, and therefore SD is more advantageous than SE. We have also noted that the SNR variations affect the complexities of the two decoders, especially the SE. In fact, the lower the SNR, the worse the estimation

of the closest point by the BP is, and consequently the longer the search stage .

## REFERENCES

- [1] M. O. Damen, A. Chkeif, and J.-C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Communications Letters*, vol. 4, pp. 161–163, May 2000.
- [2] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Transactions on Information Theory*, vol. 48, pp. 2201–2214, August 2002.
- [3] S. Galliou and J.-C. Belfiore, "A new family of full rate, fully diverse space-time codes based on Galois theory," in *Proceedings ISIT*, p. 419, July 2002.
- [4] H. El Gamal and M. O. Damen, "Universal space-time coding," *submitted to IEEE Transactions On Information Theory*, January 2002.
- [5] M. Damen, A. Tewfik, and J. Belfiore, "A construction of a space-time code based on theory of numbers," *IEEE Transactions on Information Theory*, vol. 48, pp. 753–760, March 2002.
- [6] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Transactions on Information Theory*, vol. 45, pp. 1639–1642, July 1999.
- [7] E. Viterbo and E. Biglieri, "A universal lattice decoder," in *GRETSI 14<sup>eme</sup> colloque*, (Juan les Pins), September 1993.
- [8] M. O. Damen, K. Abed-Meraim, and J.-C. Belfiore, "Generalized sphere decoder for asymmetrical space-time communication architecture," *Electronics Letters*, vol. 36, pp. 166–167, January 2000.
- [9] B. M. Hochwald and S. ten Brink, "Acheiving near-capacity on a multiple-antenna channel," *Bell Laboratories Lucent Technologies*, August 2001. <http://mars.bell-labs.com>.
- [10] B. Hassibi and H. Vikalo, "On the expected complexity of sphere decoding," *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1051–1055, 2001.

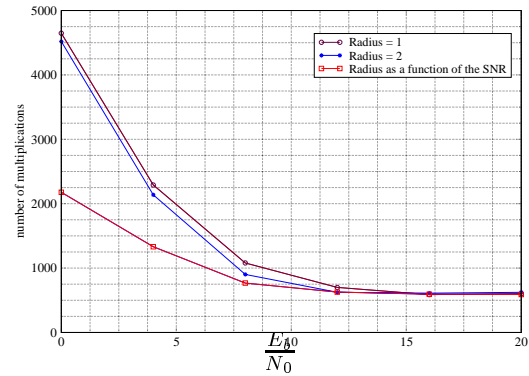


Fig. 1. Number of multiplications as a function of  $\frac{E_b}{N_0}$ , for uncoded system,  $M = N = 4$  antennas, Sphere Decoder

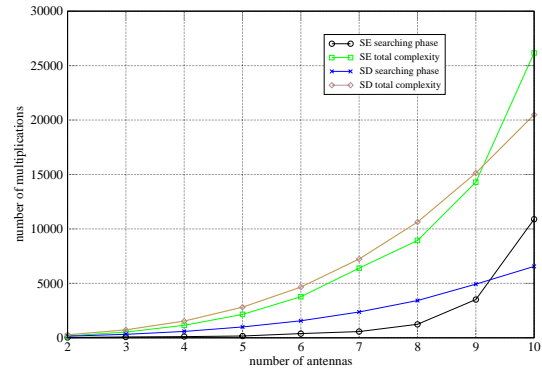


Fig. 2. Number of multiplications of SE and SD as a function of the number of antennas, at  $\frac{E_b}{N_0} = 15$  dB, for uncoded schemes

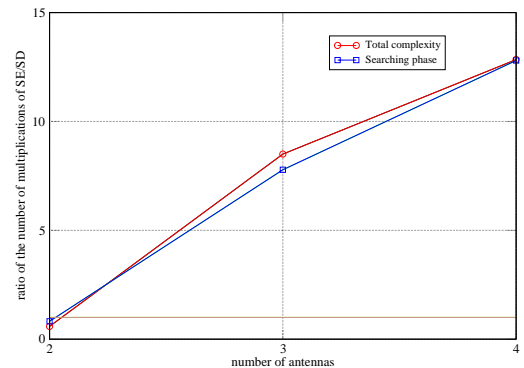


Fig. 3. Ratio of the number of multiplications SE/SD as a function of the number of antennas, at  $\frac{E_b}{N_0} = 5$  dB, for coded schemes

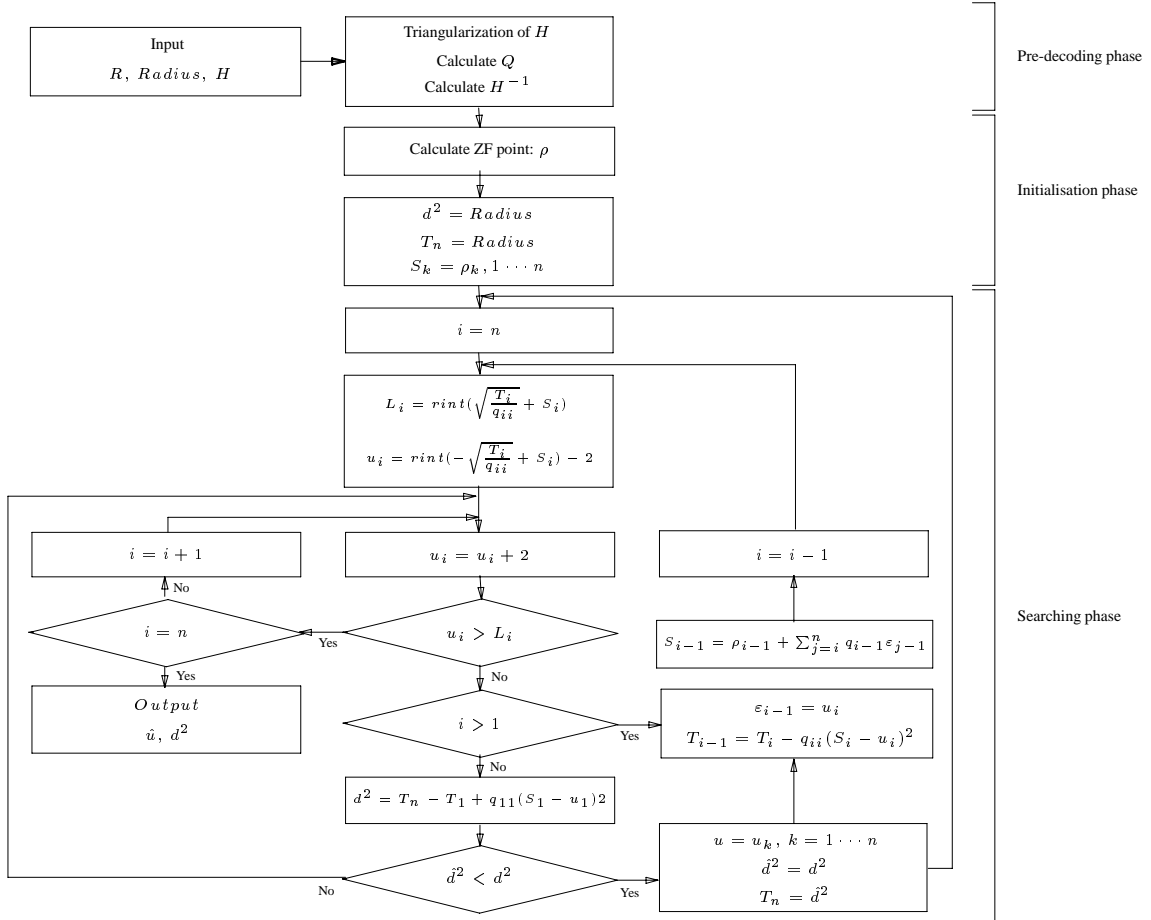


Fig. 4. Sphere decoder flow chart

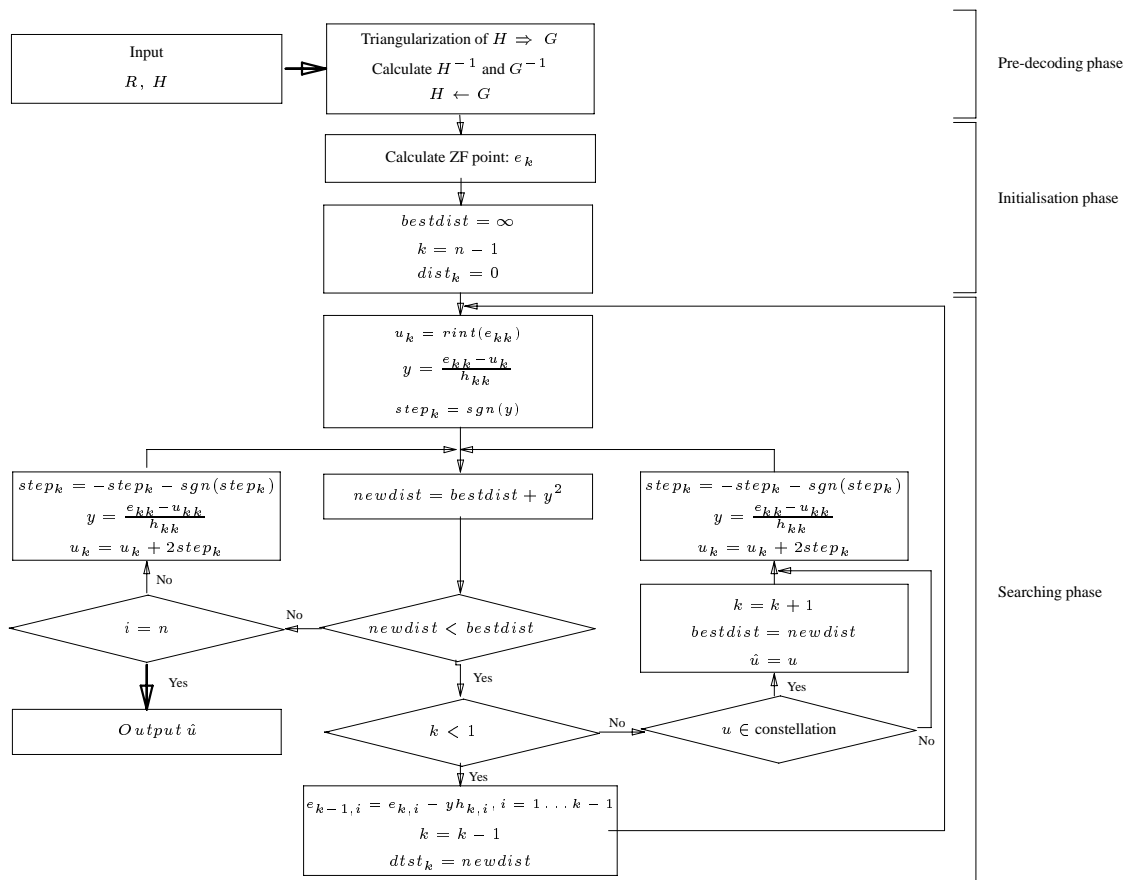


Fig. 5. Schnorr-Euchner decoder flow chart