



HAL
open science

Binary Subchannel Decomposition of Lattice-based Multiple Access Schemes

Abir Ben Hadj Fredj, Jean-Claude Belfiore

► **To cite this version:**

Abir Ben Hadj Fredj, Jean-Claude Belfiore. Binary Subchannel Decomposition of Lattice-based Multiple Access Schemes. Wimob 2017 The 13th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications., Oct 2017, Rome, Italy. hal-01576140

HAL Id: hal-01576140

<https://hal.science/hal-01576140>

Submitted on 22 Aug 2017

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.

Binary Subchannel Decomposition of Lattice-based Multiple Access Schemes

Abir Ben Hadj Fredj
Telecom ParisTech
Paris, France

abir.benhadjfredj@telecom-paristech.fr

Jean-Claude Belfiore
Telecom ParisTech
Paris, France

jean-claude.belfiore@telecom-paristech.fr

Abstract—Lattice codes used under the Compute-and-Forward paradigm suggest an alternative strategy for the standard Gaussian multiple-access channel (MAC). It has been proven that decoding an integer linear combination of the transmitted codewords enables higher data rates compared to decoding the messages individually. Recent work of Gaspar et al. proposed a new multiple access channel technique that they called Compute-and-Forward Multiple Access Channel (CFMA). The authors proved that, in the case of a two-user MAC and without time sharing, the entire capacity region is achieved using CFMA and a single-user decoder for signal to noise ratios above $1 + \sqrt{2}$. Throughout this paper, we will work within the Compute-and-Forward framework. The strategy relies on lattice codes, here we will construct lattices using a multi-level coding based construction called construction D . To decode the lattice points, a multi-stage decoder (MSD) should be implemented at the receiver part. In this paper, we propose a new decoding strategy that helps to perform the standard MSD algorithm in a less complex way. The new technique is called Binary Subchannel Decomposition (BSCD). Since, we make use of MSD, we need at each level of the decoder to compute soft-inputs: Log-Likelihood ratios. However the calculations are very complex since we must compute infinite sums of exponentials at each stage. Hereinafter we give efficient LLR approximation for multi-stage decoders. Simulation results about the performance of BSCD are included and compared to the performance of the standard multi-stage decoder.

I. INTRODUCTION

Future 5G mobile networks rise many challenges for researchers due to the high requirements that they target; Mainly, the future mobile communication system must deal with very high data rates, very dense crowds of users as well as higher requirements on the end-to-end performance and user experience. To meet these goals, several methods and strategies already adopted in existing wireless communication systems must be reconsidered. One key element of any cellular communication system is the multiple access technology that is used. Finding the appropriate channel access method is one of the challenges for future 5G systems. Channel access methods have evolved from one mobile generation to another; going from FDMA/TDMA in the 2G, through CDMA implemented in 3G, to recently OFDMA used in recent 4G systems. There are several candidate systems that are being considered as the 5G multiple access scheme. They include a variety of different ideas; one of the most promising methods is the non-orthogonal access to the multiple access channel. Compute-and-Forward used along with lattice codes is one of the non-

orthogonal access methods that was shown to yield promising results [9]. Compute-and-forward was firstly introduced in [2] and it was based on physical-layer network coding which is a revolutionary concept introduced by Zhang in 2006 [1]. Many cooperative schemes have been also proposed after the invention of physical-layer network coding, but most of them rely on three core relaying strategies, which are: Decode-and-Forward, Compress-and-Forward and Amplify-and-Forward. The key idea is that users help relay each other's messages by exploiting the broadcast and multiple-access properties of the wireless medium. However, these strategies have certain limitations. B. Nazer and M. Gaspar, introduced in [2], the Compute-and-Forward strategy which enables relays to decode integer linear combinations of the transmitted messages using the noisy linear combinations provided by the channel. In order to be decoded reliably, the transmitted messages should be nested lattice codes whose algebraic structure ensures that integer linear combination of lattice points is also a lattice point. It was proven later in [9] that Compute-and Forward (CF) is not only a good relaying strategy but also a good multiple access technique for Gaussian multiple access channels; In this case, the receiver decodes the linear combinations of the codewords and once it gets enough equations it decodes the transmitted codewords.

In this paper, we will use Compute-and-Forward for the Gaussian multiple access channel (MAC) using a different approach for the receiver rather than the one described in [9]. Remember that a multiple access channel model consists of several senders and receivers, each sender transmits information to each receiver over the same physical resource [4]. We will consider lattice codes constructed using multi-level coding (MLC) which requires a multi-stage decoder (MSD) at the receiver part. The construction is referred to as binary construction D . Our main contribution will be within the MSD, we will upgrade the decoder by adding a binary subchannel decomposition (BSCD) method which consists on mapping the decoding algorithm from $\mathbb{Z}[i]/(1+i)^m\mathbb{Z}[i]$ to $\mathbb{F}_2[u]/u^m$, where m is the number of levels of the multi-stage decoder. We performed this new method for the simple case when three users are transmitting uncoded messages to one single receiver and we got promising results. Our method allows less complex decoding and serves as a basis for more complex schemes especially since we map data from non-binary to binary rings.

This is specifically useful for schemes using the message passing algorithm (MPA) [13] such as the sparse code multiple access technique [12]. Since we used MSD, we calculated at each stage a soft-input for the decoders called Log-Likelihood Ratio (LLR) which is defined as follows:

$$\text{LLR} = \log \frac{\mathbb{P}(y'_{ij}|x_{ij} = 0)}{\mathbb{P}(y'_{ij}|x_{ij} = 1)} \quad (1)$$

where y'_{ji} and x_{ji} are respectively the j^{th} received bit after a modulo operation and the j^{th} transmitted bit at the i^{th} level of the multistage decoder. For Compute and Forward, decoding is performed over the whole lattice, therefore LLRs are computed over the whole lattice which induce an infinite sum of exponentials. Thus, the computations of exact LLRs are very complex for a practical implementation. To solve this complexity problem, we proposed hereinafter an efficient approximation of LLRs.

Throughout this paper, vectors and matrices are denoted by lowercase and uppercase bold letters, such as \mathbf{a} and \mathbf{A} , respectively. The probability of a given event E is denoted by $\mathbb{P}(E)$. Let \mathbb{C} denote the complex field and \mathbf{h}^* the hermitian transpose of a complex vector $\mathbf{h} \in \mathbb{C}^n$

II. SYSTEM MODEL AND RECEIVER STRUCTURE:

A. Received Signal:

Let us consider K users transmitting length- n , complex-valued messages $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\}$ to one receiver over a Gaussian channel. At the receiver part, the received signal $\mathbf{y} \in \mathbb{C}^n$ is expressed as follows:

$$\mathbf{y} = \sum_{j=1}^K h_j \mathbf{x}_j + \mathbf{z} \quad (2)$$

Where $h_j \in \mathbb{C}$ are the channel coefficients and \mathbf{z} is a length- n σ^2 -variance additive white Gaussian noise (AWGN).

In order to decode the transmitted codewords, we will use the Compute-and-Forward strategy [2] which consists in decoding an interger linear combination of the codewords instead of decoding individual messages (fig.1). To ensure reliable decoding, each user uses a nested lattice code as in [2]. A nested lattice code is the set of all fine lattice codes within the Voronoi region of the coarse lattice centered in the origin. Note that a lattice Λ is said to be nested in a lattice Λ_1 if $\Lambda \subset \Lambda_1$, in this case Λ is called the coarse lattice and Λ_1 is the fine lattice. In our system model, the transmitted codewords $\mathbf{x}_j, j \in \{1, 2, \dots, K\}$ are carved from the same lattice that we call Λ .

B. Compute-and-Forward for Multiple Access:

Compute and Forward is based on nested lattice codes. Each user's encoder maps length- k messages w from a prime-size finite field \mathbb{F}_p^k to a lattice Λ . At the receiver part, the received signal is a noisy combination of the transmitted

codewords (eq.2). To decode these latter ones, we must have K linearly independent equations while trying to make each of these noisy equations as close as possible to an integer noiseless combination of the codewords. Since we consider a complex channel model, this is made by approximating the channel coefficients h_j by Gaussian integers $a_{ij} \in \mathbb{Z}[i]$, $j \in \{1, 2, \dots, K\}$ and $i \in \{1, 2, \dots, K\}$ such that the synthesized received noiseless signals are in the lattice Λ .

$$\mathbf{y}_i = \sum_{j=1}^K a_{ij} \mathbf{x}_j \in \Lambda, \quad i \in \{1, 2, \dots, K\} \quad (3)$$

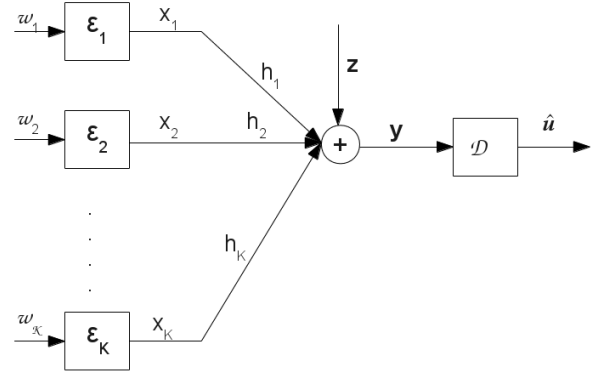


Fig. 1. A single receiver decoding an equation of K transmitted messages coming from K users

Since we have to decode K users, we need, at the receiver part, K linearly independent equations with coefficients in $\mathbb{Z}[i]$. For each equation, we scale the received signal by a factor α_i which is judiciously chosen through an optimization process (that we will detail later on in this section). The equations can be expressed as follows: for any $i \in \{1, 2, \dots, K\}$:

$$\tilde{\mathbf{y}}_i = \alpha_i \mathbf{y} = \sum_{j=1}^K a_{ij} \mathbf{x}_j + \underbrace{\sum_{j=1}^K (\alpha_i h_j - a_{ij}) \mathbf{x}_j}_{\tilde{\mathbf{z}}_i} + \alpha_i \mathbf{z} \quad (4)$$

As we mentioned earlier, α_i and a_{ij} can be easily found through a judiciously-solved optimization problem. In fact, since we need a good approximation of the received signal, the variance of the effective noise $\tilde{\mathbf{z}}_i$ must be minimized:

$$\alpha_i = \arg \min \mathbb{E}(|\tilde{\mathbf{z}}_i|^2) \quad (5)$$

The result corresponds to α_{MMSE} (Minimum Mean Square Error)

$$\alpha_i = \alpha_{\text{MMSE}} = \frac{P \mathbf{h}^* \mathbf{a}_i}{1 + P \|\mathbf{h}\|^2} \quad (6)$$

where $\mathbf{h} = [h_1, h_2, \dots, h_K]$ is the channel coefficient vector and $\mathbf{a}_i = [a_{i1}, a_{i2}, \dots, a_{iK}]$ is the vector of Gaussian integers corresponding to equation $i \in \{1, 2, \dots, K\}$ and P is the power.

On the other hand, for each equation, we associate a computation rate R_i which is a function of α_i and a_i .

$$R_i(\mathbf{h}, \mathbf{a}_i) = \max_{\alpha_i \in \mathbb{C}} \log^+ \left(\frac{P}{|\alpha_i|^2 + P \|\alpha_i \mathbf{h} - \mathbf{a}_i\|^2} \right) \quad (7)$$

Replacing α_i by α_{MMSE} we get:

$$R_i(\mathbf{h}, \mathbf{a}_i) = \log \left(\|\mathbf{a}_i\|^2 \frac{\text{SNR} |\mathbf{h}^* \mathbf{a}_i|^2}{1 + \text{SNR} \|\mathbf{h}\|^2} \right) \quad (8)$$

The computation rate must be maximized with respect to a_i in order to get better performance. Maximizing the computation rate is equivalent to minimizing the positive-definite hermitian matrix \mathbf{Q} which can be seen as the Gram matrix of a lattice that we call \mathcal{L} .

$$\hat{\mathbf{a}}_i = \arg \min_{\mathbf{a}_i \neq \mathbf{0}} \{\mathbf{a}_i^t \mathbf{Q} \mathbf{a}_i\} \quad (9)$$

where:

$$\mathbf{Q} = \mathbf{I} - \frac{\text{SNR}}{1 + \text{SNR} \|\mathbf{h}\|^2} \mathbf{H} \quad (10)$$

where $\mathbf{H} = [H_{lm}] = \mathbf{h}_l \mathbf{h}_m^*$, $1 \leq m, l \leq K$.

Finding the K best linearly independent equations is equivalent to find a reduced basis of \mathcal{L} . The Lenstra-Lenstra-Lovasz (LLL) reduction algorithm [6] enables the computation of this reduced basis that we call \mathbf{A} . Since we are working within the complex channel model, we will use and implement the complex LLL reduction which was introduced in [10].

Once we have the matrix \mathbf{U} , which is a transformation of the reduced lattice basis \mathbf{A} , the scaling factors $\alpha_i, i \in \{1, 2, \dots, K\}$ as well as the decoded lattice points $\hat{\mathbf{y}}$, we will only have to invert the $K \times K$ unimodular matrix \mathbf{U} to get the K decoded codewords $\hat{\mathbf{x}}$.

$$\hat{\mathbf{x}} = \mathbf{U}^{-1} \hat{\mathbf{y}} \quad (11)$$

III. CONSTRUCTION D OF LATTICES:

We cannot cite or mention all related works on lattices here but we can point the interested reader to an excellent survey by Zamir [3]. By reading this survey, the reader will be able to understand the usefulness of lattices, their definitions, applications and interests. Notice that in this survey, the main focus was on \mathbb{Z} -lattices, however in our paper our fundamental interest will be on $\mathbb{Z}[i]$ -lattices. It goes without saying that any $\mathbb{Z}[i]$ -lattice can be seen as two \mathbb{Z} -lattices.

An Hermitian n -dimensional lattice Λ is an additive discrete

subgroup (of maximal rank) of an Hermitian space [8]. One way to define a lattice is using its generator matrix:

$$\Lambda = \{\mathbf{y} \in \mathbb{C}^n; \mathbf{y} = \sum_{i=1}^n \mathbf{b}_i x_i = \mathbf{B} \mathbf{x}, x_i \in \mathbb{Z}[i]\} \quad (12)$$

\mathbf{y} are points of the lattice, \mathbf{b}_i are the generating vectors where each corresponds to the i^{th} column of the $n \times n$ generator matrix \mathbf{B} . Each Gaussian integer x_i is an element of the column vector \mathbf{x} . Thus a lattice is the span of column space of the generator matrix \mathbf{B} . A pair of euclidean lattices (Λ_1, Λ_2) is nested if $\Lambda_2 \subseteq \Lambda_1$, and the finite quotient group Λ_1/Λ_2 is called a lattice partition with index $|\Lambda_1/\Lambda_2|$. The partition index is the number of cosets of Λ_2 in Λ_1 . A family of m nested lattices defines a lattice partition chain $\Lambda_1/\Lambda_2/\dots/\Lambda_m$. There are several lattice constructions. Binary construction A is one of the less complex ways to build lattices. However, since it has several limitations in terms of coding gain, it is better to go for more sophisticated constructions such as binary construction D [7]. Binary construction D produces lattice packings from a family of nested binary linear codes and is well-known for the construction of Barnes-Wall lattices from Reed-Muller codes. It is based on multi-level coding and can be decoded using multistage decoding. This multilevel nature allows lower complexity decoders in comparison with other constructions. The signal flow of the multi-stage decoding scheme is shown in figure 2.

As an example, let us consider the lattice partition chain $\mathbb{Z}^n[i]/(1+i)\mathbb{Z}^n[i]/2\mathbb{Z}^n[i]/\dots/(1+i)^m\mathbb{Z}^n[i]$ and the linear binary code $C_j(n)$ of length n , the code associated to the partition $(1+i)^j\mathbb{Z}^n[i]/(1+i)^{j+1}\mathbb{Z}^n[i]$. We consider as well a family of m nested codes $C_0 \subseteq C_1 \subseteq \dots \subseteq C_{m-1}$. Then the lattice Λ can be defined as follows:

$$\Lambda = (1+i)^m\mathbb{Z}^n[i] + (1+i)^{m-1}C_{m-1}(n) + \dots + C_0(n) \quad (13)$$

Note that binary construction D with m levels can also be represented as a construction A over the polynomial ring $\mathbb{F}_2[u]/u^m$, the ring of polynomials with binary coefficients and degree m .

Hereinafter, the considered transmitted messages will be carved out from a lattice Λ constructed using construction D (which also can be seen as construction A over $\mathbb{F}_2[u]/u^m$).

IV. BINARY SUBCHANNEL DECOMPOSITION WITH MULTISTAGE DECODING:

A lattice Λ constructed using construction D is written as follows:

$$\Lambda = (1+i)^m\mathbb{Z}[i]^n + \mathcal{C}_{\mathbb{F}_2[u]/u^m}(n) \quad (14)$$

where $\mathcal{C}_{\mathbb{F}_2[u]/u^m}(n)$ is a linear code of length n and m is a positive integer which represents the number of levels of the

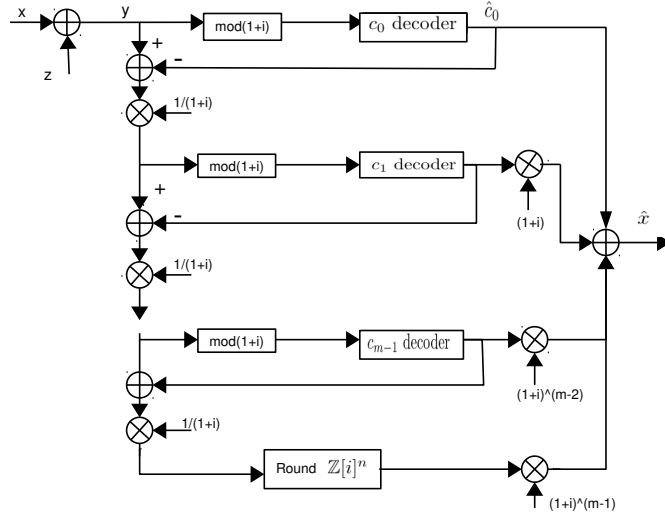


Fig. 2. The signal flow of the multistage decoding for the lattice partition $\mathbb{Z}^n[i] | (1+i)\mathbb{Z}^n[i] | \dots | (1+i)^m \mathbb{Z}^n[i]$

multi level coder. Hereinafter, $\mathcal{C}_{\mathbb{F}_2[u]/u^m}(n)$ will be called $\mathcal{C}_{\mathcal{R}}$ and will be defined as follows:

$$\mathcal{C}_{\mathcal{R}} = \mathcal{C}_{\mathbb{F}_2}^0 + u\mathcal{C}_{\mathbb{F}_2}^1 + u^2\mathcal{C}_{\mathbb{F}_2}^2 + \dots + u^{m-1}\mathcal{C}_{\mathbb{F}_2}^{m-1} \quad (15)$$

The decoding algorithm is mapped from $\mathbb{Z}[i]/(1+i)^m\mathbb{Z}[i]$ to the polynomial ring $\mathbb{F}_2[u]/u^m$ which is possible thanks to the ring isomorphism $\tau : (1+i) \mapsto u$.

Note that a ring isomorphism τ is defined as follows:

- $\tau(a+b) = \tau(a) + \tau(b)$
- $\tau(a.b) = \tau(a).\tau(b)$

$$\tau : \mathbb{Z}[i]/(1+i)^m\mathbb{Z}[i] \rightarrow \mathbb{F}_2[u]/u^m$$

$$\hat{y}_i \mapsto t_i$$

Each component of the resulting point is written as follows:

$$t_i = b_{i,0} + ub_{i,1} + u^2b_{i,2} + \dots + u^{m-1}b_{i,m-1} \quad (16)$$

$b_{i,k} \in \{0,1\}, k \in \{0,1,\dots,(m-1)\}$. To decode the system of K linearly independent equations defined in section two, we will use a multistage decoder. The received signal $y \in \mathbb{C}$ undergo the process described in section two, the resulting system of equations will be mapped to the polynomial ring $\mathbb{F}_2[u]/u^m$. The output of this mapping is the system t_i defined in (eq.16). For each level k , we decode all the codes corresponding to this level $k \in \{1,2,\dots,(m-1)\}$, in each equation $t_i, i \in \{1,2,\dots,K\}$. Once the level is decoded, we decode the corresponding level of the transmitted messages by inverting the matrix defined in (eq.11) U which is also mapped to $\mathbb{F}_2[u]/u^m$. The resulting decoded subchannel will serve as a side information for the decoding of the next subchannels. The operation is performed for all m levels till we decode the whole

codewords. In other words, each equation, of the system of K equations, will be split into m binary parallel subchannels. Each subchannel will be decoded for each equation and will be sent to the next subchannel to decode it.

For this multistage decoder used along with binary subchannel decomposition, we need at each stage soft-input values which are the log-likelihood ratios defined by:

$$\text{LLR} = \log \frac{\mathbb{P}(y'_i | x_i = 0, \hat{x}_{\hat{i}})}{\mathbb{P}(y'_i | x_i = 1, \hat{x}_{\hat{i}})} \quad (17)$$

where y'_i is the image of the observation y in $\mathbb{F}[u]/u^4$ and x_i is the transmitted message at the level i and where $\hat{x}_{\hat{i}}$ are the decoded bits at levels $\hat{i} < i$. However, exact LLR computation taking into account all lattice points is very complex and is time and energy consuming. Hereinafter, we propose an efficient LLR approximation which is simply a function of hyperbolic and circular cosines.

V. AN EFFICIENT LLR APPROXIMATION:

In a previous work, we have given efficient LLR computations for different lattice constructions [5]; The computations were expressed in terms of sums and products of Jacobi Theta functions. In the case of real binary construction D, LLRs are expressed as follows:

$$\text{LLR} = \log \left(\frac{\sum_{k \in \mathbb{Z}} e^{-\frac{\|y-2k\|^2}{2\sigma^2}}}{\sum_{k \in \mathbb{Z}} e^{-\frac{\|y-(2k+1)\|^2}{2\sigma^2}}} \right)$$

$$= \log \frac{\vartheta_3\left(\frac{iy}{\sigma^2}; e^{-\frac{2}{\sigma^2}}\right)}{\vartheta_2\left(\frac{iy}{\sigma^2}; e^{-\frac{2}{\sigma^2}}\right)} \quad (18)$$

where the Jacobi Theta functions of type 2 and type 3 are defined in the general case as follows [11]:

$$\vartheta_2(\mathbf{u}; q) = \sum_{k=-\infty}^{\infty} q^{(k+\frac{1}{2})^2} e^{(2k+1)i\mathbf{u}} \quad (19)$$

$$\vartheta_3(\mathbf{u}; q) = \sum_{k=-\infty}^{\infty} q^{k^2} e^{2ik\mathbf{u}} \quad (20)$$

Note that Jacobi Theta functions are the elliptic analogs of the exponential functions and that they are defined as functions of two complex variables u and q .

In this paper, we will further simplify these expressions and we will propose an efficient LLR approximation which will only be expressed in terms of circular and hyperbolic cosines. This approximation makes the calculations much easier and as we will prove in this section, allows to get the same results as the real expressions of the log-likelihood ratios. As for the LLR computations, we begin by giving the approximations of the Theta functions of the sublattices and lattice translates $2\mathbb{Z}$, $2\mathbb{Z} + 1$ as well as $4\mathbb{Z}$ and all its translates. Notice here that we limited ourselves to the 16-QAM constellation but the results can be easily extended.

$$\begin{aligned} \Theta_{2\mathbb{Z}}(y, q) &= \vartheta_3\left(\frac{iy}{\sigma^2}; e^{-\frac{2}{\sigma^2}}\right) \simeq \frac{e^{\rho \cos(\pi y)}}{2I_0(\rho)} \\ \Theta_{2\mathbb{Z}+1}(y, q) &= \vartheta_2\left(\frac{iy}{\sigma^2}; e^{-\frac{2}{\sigma^2}}\right) \simeq \frac{e^{\rho \cos(\pi(y-1))}}{2I_0(\rho)} \end{aligned} \quad (21)$$

where $I_0(\rho)$ is the modified Bessel function of first order and ρ is the positive solution of the equation

$$\frac{e^\rho}{2I_0(\rho)} = \vartheta_3\left(e^{-\frac{4}{2\sigma^2}}\right) \quad (22)$$

In the general case, any Jacobi Theta function of a binary sublattice of \mathbb{Z} can be expressed as follows:

$$\Theta_{2^m\mathbb{Z}} \simeq \frac{e^{\rho_{2^m} \cos(\frac{2\pi}{2^m} y)}}{2^m I_0(\rho_{2^m})} \quad (23)$$

It is also easy to prove that the Jacobi Theta function of each translate of a sublattice of \mathbb{Z} can be expressed as follows:

Let us define first:

$$\phi_\Lambda(y) = \left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n \sum_{x \in \Lambda} e^{-\frac{\|y-x\|^2}{2\sigma^2}} \quad (24)$$

Then for any translation vector y_0 , we have:

$$\phi_{\Lambda+y_0}(y) = \left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n \sum_{x \in \Lambda+y_0} e^{-\frac{\|y-x\|^2}{2\sigma^2}} \quad (25)$$

if we suppose $x = u - y_0$, then:

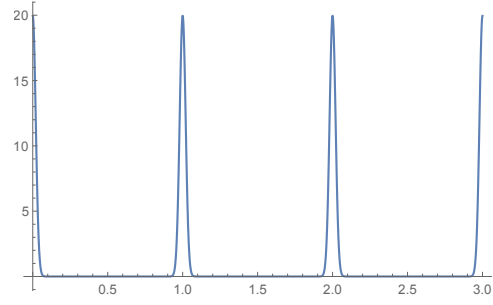


Fig. 4. Approximation of the Jacobi Theta function of \mathbb{Z} for $\sigma = 0.6$

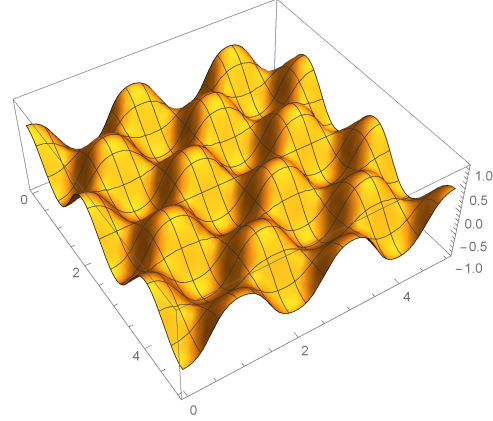


Fig. 5. Representation of the estimation of LLR for the partition $\mathbb{Z}^2 \setminus D_2$ and for $\sigma = 0.6$

$$\begin{aligned} \phi_{\Lambda+y_0}(y) &= \left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n \sum_{u \in \Lambda+y_0} e^{-\frac{\|y-y_0-u\|^2}{2\sigma^2}} \\ &= \phi(y - y_0) \end{aligned} \quad (26)$$

We conclude then, that computing the approximation of the Jacobi Theta function that corresponds to the coset 0 is sufficient. Those corresponding to the other cosets are simple translations of the coset 0. Hereinafter, since we will consider the 16-QAM constellation, we will deal with the partition chain $\mathbb{Z}^2 \setminus D_2 \setminus 2\mathbb{Z}^2 \dots$. Two main LLRs correspond to the latter partition chain. LLRs corresponding to the D_2 -like cosets and LLRs corresponding to the \mathbb{Z}^2 -like cosets (see fig.3), they can be expressed respectively as follows:

$$LLR_{D_2} \simeq \log\left(\frac{\cosh(\rho_{2^m} (\cos(\frac{2\pi}{2^m}(y_1 - k_1)) + \cos(\frac{2\pi}{2^m}(y_2 - k_2))))}{\cosh(\rho_{2^m} (\cos(\frac{2\pi}{2^m}(y_1 - k_1))) - \cos(\frac{2\pi}{2^m}(y_2 - k_2)))}\right) \quad (27)$$

$$LLR_{\mathbb{Z}^2} \simeq 2\rho_{2^m} (\cos(\frac{2\pi}{2^m}(y_1 - k_1)) + \cos(\frac{2\pi}{2^m}(y_2 - k_2))) \quad (28)$$

where $y = (y_1, y_2)$ and $k = (k_1, k_2)$ is the translation vector. Log-likelihood ratios and Jacobi theta function's estimations are represented in figures 4 and 5 in the one dimensional and two-dimensional cases.

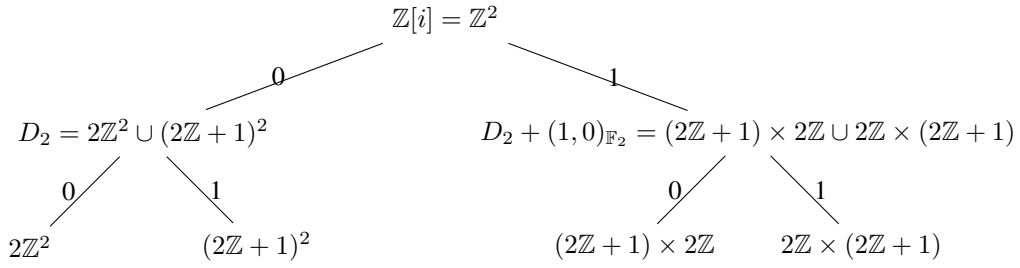


Fig. 3. Representation of the partition $\mathbb{Z}[i]|(1+i)\mathbb{Z}[i]|2\mathbb{Z}[i]$

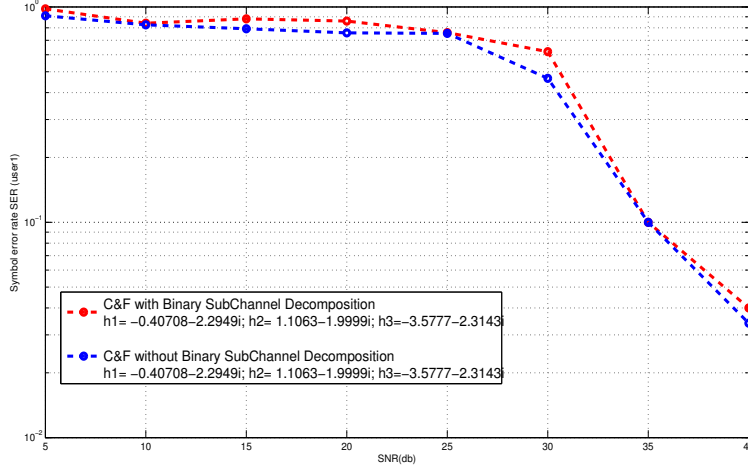


Fig. 6. Comparison between CF with BSCD and CF without BSCD in the uncoded case

VI. SIMULATION RESULTS FOR BSCD:

For our simulations, we chose to work with one single receiver and three users. Each user sends its messages independently to the receiver, however all the transmitted codewords belong to the same lattice. The lattice is constructed according to a construction based on multi level coding and a multistage decoding and it is called construction D. The transmitted messages are carved out from an uncoded lattice $\mathbb{Z}[i]|(1+i)^4\mathbb{Z}[i]$ (16-QAM constellation) and the number of levels is chosen to be $m = 4$. At the receiver part, we performed binary subchannel decomposition (BSCD) along with the MSD. The results are almost similar to when we perform only Compute-and-Forward without BSCD over the same system. Simulation results are shown in figure 6. As you can notice, the performance of multi-stage decoding with BSCD is very close to the standard multi-stage decoding. However, in our case the complexity is significantly reduced.

VII. CONCLUSION:

In this paper, we considered Compute-and-Forward for the Gaussian multiple access channel and we presented a different approach for the receiver. We introduced as well a new technique called binary subchannel decomposition to

perform the multi-stage decoding algorithm of Compute-and-Forward. The results show that we have the same performance as the original decoding algorithm of Compute-and-Forward. Binary subchannel decomposition allows important gains in complexity. It can serve as a basis for more complex schemes such as sparse code multiple access (SCMA), which was recently introduced in [12]. In this case of SCMA, we use the message passing algorithm (MPA) multi-user detection technique. Since, the BSCD allows us to deal only with binary data, as we map the whole decoding algorithm to the field $\mathbb{F}_2[u]/u^m$, this can largely simplify the decoding algorithm used in such multiple access strategy. To further simplify the multi-stage decoder, we also gave useful LLR approximations based on the LLR expressions given in a previous work [5]. The LLR approximations are simple functions of circular and hyperbolic cosines.

REFERENCES

- [1] S Zhang SC Liew and P P Lam, *Hot-topic Physical layer network coding*. in Proc, 12th MobiCom, pages 358-365, New York, NY, USA 2006.
- [2] B. Nazer and M. Gastpar. *Compute-and-forward: Harnessing interference through structured codes*. Information theory,IEEE transactions,vol.57,pp.6463-6486 Oct2011.
- [3] Ram Zamir, *Lattices are Everywhere* . Springer New York,2010
- [4] Jingge Zhu, Michael Gastpar *Gaussian Multiple Access via Compute-and-Forward*. IEEE Transactions on Information Theory Aug 2016
- [5] A. Ben Hadj Fredj, J.C. Belfiore *LLR Computation for Multistage Decoding*. Networks and Communications (EuCNC), 2016 European Conference June 2016
- [6] Lenstra, A. K., Lenstra, H. W., Jr., Lovsz, L., *Factoring polynomials with rational coefficients*. Mathematische Annalen, 1982
- [7] Yuh-Chih Huang and Krishna R. Narayanan *Construction π_A and π_D Lattices: Construction, Goodness, and decoding algorithms*. International Symposium on Information Theory 2014
- [8] J. Conway and N.J..A Sloane, *sphere Packings, Lattices and groups*.. Springer New York,2010
- [9] Jingge Zhu and Michael Gasptar *Gaussian Multiple Access via Compute and Forward* IEEE Transactions on Information Theory July 2014
- [10] Ying Hung Gan, Cong Ling, Wai Ho Mow *Complex Lattice Reduction Algorithm for Low-Complexity Full-diversity MIMO Detection* IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 57, NO. 7 JULY 2009
- [11] Milton Abramowitz and Irene A. Stegun *Handbook of Mathematical Functions* Verlag Harri Deutsch -Thun - Frankfurt/Main 19
- [12] Mahmoud Taherzadeh, Hosein Nikopour,Alireza Bayesteh, and Hadi Baligh *SCMA Codebook Design* Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th 2014
- [13] Kavitha Sunil, Poorna Jayraj, K.P.Soman *Message Passing Algorithm: A Tutorial Review* IOSR Journal of Computer Engineering JULY-August 2012