

TCM-NOMA: Joint Multi-user Codeword Design and Detection in Trellis Coded Modulation based NOMA for Beyond 5G

Boya Di*, *Student Member, IEEE*, Lingyang Song*, *Fellow, IEEE*, Yonghui Li†, *Fellow, IEEE*, Geoffrey Ye Li‡, *Fellow, IEEE*

Abstract—In this paper, we propose a novel non-orthogonal multiple access (NOMA) scheme based on trellis-coded modulation (TCM). Different from those in the traditional code-domain NOMA, the incoming data streams of multiple users are jointly coded and mapped to the codewords so as to improve the coding gain of the system without any bandwidth extension. Both codeword design and multi-user detection are considered to optimize the TCM for NOMA. The multi-dimensional (MD) constellation is carefully selected and a new set partitioning algorithm is developed for MD signal set labeling. To achieve the trade-off between the BER performance and complexity, a suboptimal two-layer Viterbi algorithm is proposed for joint decoding. Simulation results show that our proposed TCM-based NOMA scheme performs significantly better than the traditional code-domain NOMA and OFDMA schemes in terms of the BER performance under the same channel conditions.

Index Terms—Non-orthogonal multiple access, trellis coded modulation, codeword design, multi-user detection.

I. INTRODUCTION

The past few years have witnessed the explosive growth of demand for massive data transmission brought by the increasing number of mobile devices and applications [2]. The traditional orthogonal frequency division multiple access (OFDMA) techniques in the fourth-generation (4G) networks fail to achieve the adequate exertion of the frequency resources due to its orthogonal nature [3], [4]. This also causes access congestion especially in a dense network, urging for more efficient solutions. Capable of supporting massive connectivity and improving the spectral efficiency [5], non-orthogonal multiple access (NOMA) has drawn widespread attention as a promising candidate to solve the above issues in future communications [6], [7].

In NOMA systems [8], [9], multiple users are allowed to share the same resources, such as a subchannel, a time slot and so on. One of the efficient schemes is the code-domain NOMA [10], [12]. Specifically, the incoming data of each layer are mapped to a sparse codeword in which the non-zero elements correspond to the occupying subcarriers of each layer. After superimposing the codewords of multiple layers, the transmitter sends multiple data streams to the

receiver simultaneously over a limited number of subcarriers, achieving the overloading function [13]. Each layer is assigned a pre-defined codebook where the mapping between non-zero elements and the data streams is constructed based on multi-dimensional (MD) modulation techniques [11], [12]. To decode the superimposed codeword, the belief probability based message passing algorithm (MPA) [14] has been applied at the receiver.

For the code-domain NOMA, previous works [15]–[18] have discussed various schemes to improve the coding gain. In [15], a constellation optimization method has been developed in which the structure of the MD complex constellation has been optimized based on the star-QAM constellation. In [16], the Turbo TCM technique has been utilized to generate the multi-dimensional codebooks for each user separately, and the pure MPA has been adopted for decoding. In [17], a new constellation has been proposed and analyzed based on a novel performance criterion. In [18], a low-complexity detector has been designed for an uplink NOMA system based on the adaptive Gaussian approximation. Most existing works [15]–[18] focus on either optimizing the procedures of MD constellation construction or improving the performance of MPA under the framework in [12]. However, the practical utilization of NOMA systems can be further improved if joint multi-user codeword design can be achieved.

In this paper, we jointly design codewords for multiple users simultaneously via MD trellis-coded modulation (TCM) techniques. The data streams of different users are jointly coded and mapped directly to the sparse codewords. Traditionally, TCM techniques [19] have been well investigated as a suitable approach to jointly optimize the error control coding and modulation through the signal set expansion instead of bandwidth expansion. Differently, we consider the MD-TCM technique purely as a method to design codewords for NOMA users, instead of replacing the channel coding. New challenges have been posed on both the codeword design and the multi-user detection. On one hand, due to the multiplexing nature, the mapping from the MD constellation point to the superimposed codeword is different from that in the traditional MD-TCM scheme. Therefore, a new method for selecting and labeling the MD constellation should be considered. On the other hand, due to the entanglement of encoding and multiplexing, the traditional MPA or Viterbi algorithm cannot be directly applied to decode the TCM-based NOMA signals, requiring a new decoding scheme.

To tackle the above challenges, we design new encoding

The

Part of this paper was presented in the International Conference on Communications (ICC) [1], 2018.

B. Di and L. Song are with the School of Electrical Engineering and Computer Science, Peking University, China. Y. Li is with the School of Electrical and Information Engineering, The University of Sydney, Australia. G.Y. Li is with School of Electrical and Computer Engineering, Georgia Institute of Technology, USA.

and decoding schemes for joint codeword design in an overloaded NOMA system to improve the coding gain. Over each subcarrier, the data streams of multiple users are rearranged into one data sequence, which is then coded and mapped to an MD constellation point. Our main contributions in this paper can be summarized as below:

- To construct the TCM encoder, a novel bipartite set partitioning algorithm based on farthest-point optimization (FPO) [20] is proposed and analyzed such that the minimum free squared Euclidean (MFSE) distance of the system can be maximized.
- A maximum likelihood sequence detection (MLSD) is developed. To balance the complexity and the BER performance, we also propose a suboptimal two-layer joint decoding scheme based on the Viterbi algorithm. The non-orthogonal nature of NOMA is utilized in this scheme and the decoding complexity is analyzed.
- Simulation results show that the proposed scheme significantly outperforms the traditional code-domain NOMA and OFDMA in terms of the BER performance. The influence of system parameters on the BER performance and decoding complexity is also investigated.

The rest of this paper is organized as follows. In Section II, we introduce the framework of TCM-based NOMA. In Section III, we discuss the TCM-based NOMA encoder design criteria and detailed steps. Based on the criteria, an FPO-based bipartite set partitioning algorithm is proposed and analyzed. In Section IV, we provide an optimal MLSD decoding scheme. To balance the complexity and BER performance, we also develop a suboptimal two-layer soft-decision based Viterbi algorithm. Simulation results are presented in Section V, and finally, we conclude the paper in Section VI.

II. SYSTEM MODEL

In this section, we first introduce downlink code-domain NOMA and then present the key idea of TCM-based joint NOMA codeword design.

A. Code-Domain NOMA Multiplexing

The general structure of a code-domain NOMA system is shown in Fig. 1. In the figure, the BS sends J data streams, each to a NOMA user. The available bandwidth is divided into K orthogonal subcarriers. Different from orthogonal access, multiple users **can share the same subcarrier simultaneously**.

At the transmitter, the data stream¹ of each user j in time unit t , denoted by $\mathbf{b}_j[t] = (b_j^1[t], \dots, b_j^q[t])$, is transmitted over N ($N \ll K$) subcarriers. Data stream $\mathbf{b}_j[t]$ is mapped to a sparse codeword $\mathbf{x}_j[t] = (x_j^{(1)}[t], \dots, x_j^{(K)}[t])$ of length K , in which N nonzero elements of this codeword represent the set of intended signal points and $(K - N)$ zero elements correspond to the unoccupied subcarriers of user j . All J NOMA users are multiplexed over K shared subcarriers.

¹In Fig. 1, we omit the subscript t due to the limited space.

Therefore, the received signal over subcarrier k can be expressed as

$$y^{(k)}[t] = h^{(k)}[t] \sum_{j=1}^J x_j^{(k)}[t] + n^{(k)}[t], \quad (1)$$

where the channel coefficient of subcarrier k in time unit t is denoted by $h^{(k)}[t]$ and the additive white Gaussian noise (AWGN) is denoted as $n^{(k)}[t] \sim \mathcal{CN}(0, \sigma^2)$, with σ^2 as the noise variance.

B. Mapping Matrix Design

The occupied subcarrier set of each user is fixed, that is, for each user the positions of non-zero elements in the sparse codeword with respect to any data streams are the same. We can use a binary $K \times J$ matrix \mathbf{F} to depict such mapping relation where each variable $f_{k,j}$ in \mathbf{F} indicates whether user j occupies subcarrier k , i.e., whether $x_j^{(k)}[t]$ is a non-zero element.

Since each codeword contains N non-zero elements out of K , there are C_K^N possible mappings between the subcarriers and each user. Note that different users occupy different subset of subcarriers. Therefore, at most $J = C_K^N$ users can be supported simultaneously. Denote d_f as the number of users sharing the same subcarrier in this case. We then have $d_f = C_{K-1}^{N-1}$.

For the NOMA system in Fig. 1 where $J = 6$, $K = 4$, $N = 2$, and $d_f = 3$, a widely used mapping matrix is presented as below:

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad (2)$$

In Fig. 1, we use a square marked by 1 to represent an occupied subcarrier by each user, otherwise a blank square marked by 0. With the mapping matrix \mathbf{F} , we can then denote the set of users occupying subcarrier k as $\mathcal{J}^{(k)} = \{j | 1 \leq j \leq J, f_{k,j} = 1\}$. For example, for the mapping matrix in (2), we have $\mathcal{J}^{(1)} = \{1, 2, 6\}$.

C. TCM-based Joint NOMA Codeword Design

In the traditional NOMA scheme [10], the codeword of each user, $\mathbf{x}_j[t]$, is only related to its data stream $\mathbf{b}_j[t]$. Each user independently selects a codeword from a pre-defined codebook. Though such a design provides a straightforward decoding method, it greatly limits the potential coding gain due to the independent coding of each user. In addition, channel coding needs to be performed ahead of the MD modulation in the existing scheme. Therefore, extra bandwidth expansion is required when both the channel coding and modulation are considered, leading to inefficient utilization of frequency resources.

To tackle the above two issues, we jointly determine the non-zero elements of all the users over each subcarrier based on their data streams. We consider a joint coding and modulation scheme MD-TCM in which the binary convolutional code and M-ary constellation are combined to obtain the coding

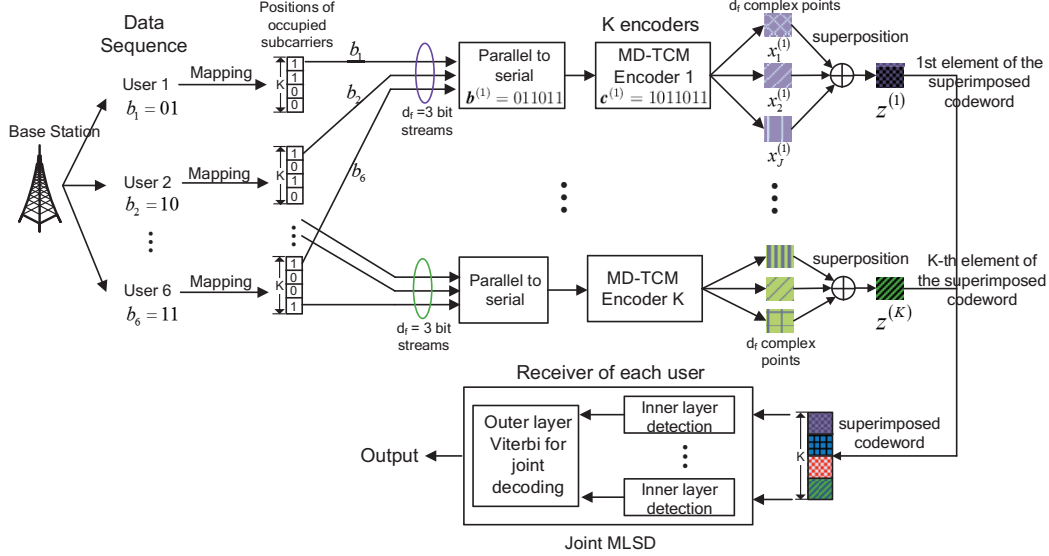


Fig. 1. The general structure of a code-domain NOMA system with joint coded modulation in which $J = 6$, $K = 4$, $N = 2$, $d_f = 3$, and the mapping matrix is shown in equation (2).

gain through the signal set expansion instead of the bandwidth expansion. The key idea can be illustrated as below.

1) *TCM-based NOMA Encoding*: Given the set of users occupying subcarrier k , $\mathcal{J}^{(k)}$, the data sequence transmitting over subcarrier k consists of data streams from d_f users in time unit t , which can be denoted as $\mathbf{b}^{(k)}[t] = \{\mathbf{b}_j[t] | j \in \mathcal{J}^{(k)}\}$. As mentioned above, according to the first row of the mapping matrix \mathbf{F} in equation (2), we have $\mathcal{J}^{(1)} = \{1, 2, 6\}$. Therefore, the data streams for user 1, 2, and 6 in the current time slot t in Fig. 1, i.e., $\mathbf{b}_1[t] = 01$, $\mathbf{b}_2[t] = 10$, and $\mathbf{b}_6[t] = 11$, respectively, are transmitted over subcarrier 1. Thus, we have $\mathbf{b}^{(1)}[t] = 011011$.

In each time unit t , a data sequence $\mathbf{b}^{(k)}[t]$ is encoded into a sequence $\mathbf{c}^{(k)}[t]$ and then mapped to an MD constellation point via the MD-TCM technique such that the event-error probability can be minimized. Each component of the MD constellation point represents a non-zero element of the sparse codewords over subcarrier k , i.e., $x_j^{(k)}$, $j \in \mathcal{J}^{(k)}$. In Fig. 1, after passing the MD-TCM encoder, the data sequence $\mathbf{b}^{(1)}[t] = (011011)$ is first encoded as $\mathbf{c}^{(1)}[t] = (1011011)$ and then projected onto a 6D constellation point consisting of three 2D components, i.e., $x_1^{(1)}[t]$, $x_2^{(1)}[t]$, and $x_6^{(1)}[t]$ in Fig. 1.

The k -th element of the superimposed codeword in time unit t can be obtained by adding these 2D components $\sum_{j \in \mathcal{J}^{(k)}} x_j^{(k)}[t]$, carrying the information of all users in $\mathcal{J}^{(k)}$. The superimposed codeword is then determined after all K elements are obtained through the above process. In Fig. 1, the first element of the superimposed codeword is $(x_1^{(1)}[t] + x_2^{(1)}[t] + x_6^{(1)}[t])$. Other elements in the superimposed codeword can be similarly obtained, represented by the dark squares in a vertical column.

2) *TCM-based NOMA Decoding*: At the receiver of each user, we jointly decode the data streams of all users by utilizing the soft-decision Viterbi based MLSD. Depending on the mapping matrix \mathbf{F} , the same data stream of each user transmits over different subcarriers, which can be utilized for

joint decoding and is different from the traditional Viterbi algorithm.

In each step of the sequence detection, we first estimate the transmitted signal points by utilizing the above non-orthogonal nature. We then update the survivor paths based on the soft-decision Viterbi algorithm. The data streams of each user can be recovered based on the survivor paths.

III. TCM-BASED NOMA ENCODER DESIGN

In this section, we first discuss the criteria for TCM-based NOMA encoder design. and then illustrate three phases of the encoder design in detail.

A. Encoder Design Criteria

1) *Notations*: We assume that the number of subcarriers is $K = 2^p$, and the number of bits to be coded and transmitted for each user j in each time unit t is $q = |\mathbf{b}_j[t]|$, $1 \leq j \leq J$, $\forall t > 0$. For example, in Fig. 1, we have $q = 2$ for each user. The sequence of non-zero elements in the sparse codewords with respect to subcarrier k is denoted as $\mathbf{x}^{(k)}[t] = \{x_j^{(k)}[t] | j \in \mathcal{J}^{(k)}\}$.

As shown in Fig. 2, we utilize the conventional lattice or star constellation to construct an MD mother constellation from which we select the signal set. To be specific, given a M-QAM constellation \mathcal{L} with size M , a $d_f \times M$ -QAM mother constellation \mathcal{L}^{d_f} can be constructed with the size of M^{d_f} . In Fig. 2, we have a 3×16 QAM mother constellation \mathcal{L}^3 .

Since the original constellation \mathcal{L} is composed of 2D signal points, the MD mother constellation consists of $2d_f$ -D points. Each $2d_f$ -D point $A_l \in \mathcal{L}^{d_f}$ is denoted by $A_l = (a_{l1}, a_{l2}, \dots, a_{ld_f})$ with $1 \leq l \leq M^{d_f}$, in which a_{li} represents a signal point from the original M-QAM constellation \mathcal{L} . We denote the position of a $2d_f$ -D point A_l in the 2D plane²

²For convenience, we denote the position z_l of the signal point A_l in the form of a complex number. The coordinates of this point in the 2D plane is $(\text{Re}(z_l), \text{Im}(z_l))$. In the remaining part of this paper, when we mention the position of a $2d_f$ -D point A_l , we refer to its position in the 2D plane.

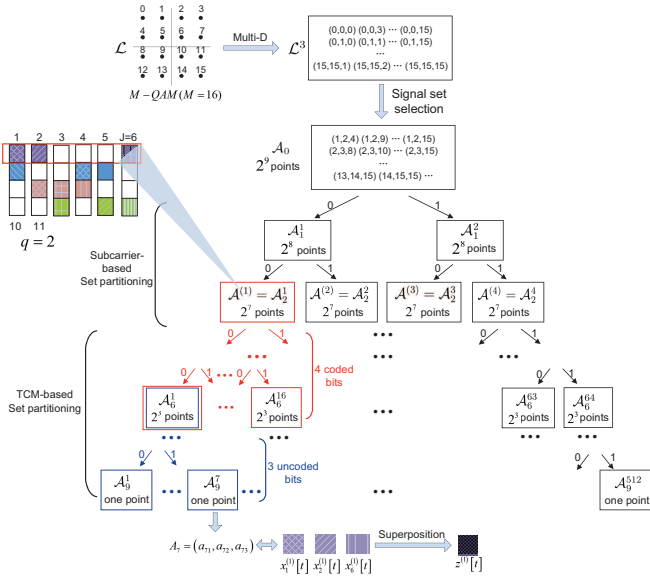


Fig. 2. MD-TCM based NOMA encoder diagram with $J = 6$, $K = 4$, $d_f = 3$, $M = 16$, $r = 3$, $p = 2$, and $q = 2$.

as $z_l = \sum_{i=1}^{d_f} a_{li}$. For example, if we have three 2D points extracted from \mathcal{L} , say, $(1 + 1i)$, $(-3 + 3i)$, and $(-1 - 1i)$ where $i = \sqrt{-1}$, and then a 6D point A_1 in Fig. 2 constructed from these points is denoted by $A_7 = (1 + 1i, -3 + 3i, -1 - 1i)$, and its position in the 2D plane is $z_7 = (-3 + 1i)$.

2) *Criteria*: With the above definitions, the process of coded modulation can be described as below. According to the TCM principle, in each time unit t , r out of qd_f bits from the data sequence $\mathbf{b}^{(k)}[t]$ are sent into a convolutional encoder of rate $r/(r+1)$. The rest of $qd_f - r$ uncoded bits from $\mathbf{b}^{(k)}[t]$ will determine a specific point in a subset of \mathcal{L}^{d_f} with the size of $2^{qd_f - r}$. In this way, the coded sequence $\mathbf{c}^{(k)}[t]$ with the length of $qd_f + 1$ is mapped to a unique $2d_f$ -D constellation point $A_l \in \mathcal{L}^{d_f}$. The k th element in the superimposed codeword can be determined based on the position of a chosen $2d_f$ -D point A_l , i.e., $z^{(k)} = \sum_{j \in \mathcal{J}^{(k)}} x_j^{(k)}[t] = \sum_{i=1}^{d_f} a_{li}$.

Take Fig. 2 as an example. For subcarrier 1, the data sequence to be transmitted is $\mathbf{b}^{(1)}[t] = 011011$. Given a signal constellation $\mathcal{A}^{(1)}$ of size 128, the first three bits of this sequence are sent into a convolutional encoder of rate $3/4$ such that the output four coded bits, say, 1011, select a signal subset of size 8, say, \mathcal{A}_6^2 . A specific 6D point of \mathcal{A}_6^2 , say, A_7 , is then selected based on the remaining three uncoded bits of $\mathbf{b}^{(1)}[t]$. The value of the first element in the superimposed codeword is actually the position of A_7 . The mapping between the coded (or uncoded) bits and the signal subset (or a 6D point) will be illustrated in Section III.B-2.

To minimize the event-error probability [19], we aim to jointly design the coding and modulation so that the MFSE distance between any two coded sequences, denoted by d_{free}^2 , can be maximized. In other words, we aim to construct an optimal mapping from the transmitted data sequence over each subcarrier to the corresponding non-zero elements in the codewords, i.e., $\mathbf{b}^{(k)} \rightarrow \mathbf{c}^{(k)} \rightarrow A_l \rightarrow \mathbf{x}^{(k)}$, $1 \leq k \leq K$, $A_l \in \mathcal{L}^{d_f}$. Since the mapping is not related to t , we omit it in the remaining part of this section.

As mentioned above, not all bits of sequence $\mathbf{b}^{(k)}$ participate

in the convolution computation, and thus, there exist parallel branches between two states of the convolutional encoder, brought by the uncoded bits. According to the TCM principle, d_{free}^2 is determined by two terms: 1) the minimum squared Euclidean (MSE) distance δ_{free}^2 between different trellis paths longer than one branch; 2) the MSE distance δ_{min}^2 between the parallel branches of the encoder. Therefore, we have $d_{free}^2 = \min\{\delta_{free}^2, \delta_{min}^2\}$. In Subsections III.B and III.C, we illustrate how to calculate δ_{min}^2 and δ_{free}^2 , respectively, as well as the relationship between them.

B. Joint Encoding

Three phases are illustrated as below, i.e., 1) signal set selection; 2) signal set labeling; 3) convolutional encoder construction.

1) *Signal Set Selection*: A signal set is selected from the MD mother constellation so as to satisfy the following two constraints on a) the size of the set; b) the uniqueness of each point's position in the set. For successful decoding, each coded data sequence $\mathbf{c}^{(k)}$ of length $(qd_f + 1)$ is required to be mapped to a unique position. For generality, we also assume that any two coded data sequences $\mathbf{c}^{(k)}$ and $\mathbf{c}^{(k')}$ transmitted over different subcarriers k and k' are mapped to two different positions³. Since there are K subcarriers, 2^{p+qd_f+1} unique positions in total are required for the selected signal set.

Therefore, we aim to construct a signal set \mathcal{A}_0 of size 2^{p+qd_f+1} from the MD mother constellation \mathcal{L}^{d_f} such that the positions of any two points projecting in the 2D plane are different, which can be mathematically formulated as:

$$\text{find } \mathcal{A}_0 \quad (3a)$$

$$\text{s.t. } z_l \neq z_{l'}, \forall A_l, A_{l'} \in \mathcal{A}_0, l \neq l', \quad (3b)$$

$$|\mathcal{A}_0| = 2^{p+qd_f+1}, \quad (3c)$$

$$\mathcal{A}_0 \subseteq \mathcal{L}^{d_f}. \quad (3d)$$

This problem can be solved by the following two steps.

a) **Obtaining a unique constellation**: Note that in the MD mother constellation \mathcal{L}^{d_f} , there may exist more than one $2d_f$ -D constellation points sharing the same position in the 2D plane. Denote the set of points sharing the same positions as \mathcal{A}_s . To keep the uniqueness of each signal point, we can only select one point $A_{l^*} \in \mathcal{A}_s$ as a member of \mathcal{A}_0 while all the other points are removed. To improve the diversity among d_f non-zero elements of the sparse codewords transmitting over the same subcarrier, we select the MD point with the largest variance among its 2D components, i.e., $l^* = \arg \max_{A_l \in \mathcal{A}_s} \text{var}(A_l)$. Note that the size of \mathcal{A}_0 is required to be no smaller than 2^{p+qd_f+1} ; otherwise, the MD mother constellation \mathcal{L}^{d_f} is required to be reconstructed.

b) **Shaping**: After step a), we denote the current constellation as \mathcal{A}_c . If the size of \mathcal{A}_c is larger than 2^{p+qd_f+1} , we continue to remove points until constraint (3c) is satisfied. For a constellation, its average energy is required to be minimized while the MFSE distance between points can be as large as

³In the simulation, we will show both cases where different signal sets or identical signal sets are allocated to different subcarriers.

possible. Therefore, we tend to remove those points close to their neighbors but far away from the original point. A greedy algorithm is adopted for shaping based on the above criterion. We sort the constellation points in \mathcal{A}_c as $A_{\pi(1)}, A_{\pi(2)}, \dots$ such that

$$\frac{\min_{A_r \in \mathcal{A}_c, r \neq \pi(j)} [z_{\pi(j)} - z_r]}{|z_{\pi(j)}|^2} \leq \frac{\min_{A_r \in \mathcal{A}_c, r \neq \pi(j')} [z_{\pi(j')} - z_r]}{|z_{\pi(j')}|^2}, \quad (4)$$

holds $\forall j \leq j'$, where the numerator in the fraction is the MFSE distance between point $A_{\pi(j)}$ and other points in \mathcal{A}_c , and the denominator represents the energy of point $A_{\pi(j)}$. We remove the points one by one in \mathcal{A}_c starting from $A_{\pi(1)}$ until the size of the remained signal set is 2^{p+qd_f+1} , i.e., the target set \mathcal{A}_0 is obtained.

2) *Signal Set Labeling*: Signal set labeling refers to the process in which each coded sequence $\mathbf{c}^{(k)}$, $1 \leq k \leq K$ is assigned a unique signal point in \mathcal{A}_0 such that the MFSE distance can be maximized. As shown in Fig. 2, two steps of signal set labeling for NOMA are listed as below:

- **Subcarrier-based Set Partitioning**: As mentioned above, any two coded sequences $\mathbf{c}^{(k)}$ and $\mathbf{c}^{(k')}$ are assigned to different signal points in \mathcal{A}_0 . Therefore, we divide \mathcal{A}_0 into K subsets with the equal size, i.e., $\mathcal{A}_0 \rightarrow \{\mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \dots, \mathcal{A}^{(K)}\}$. Each subset $\mathcal{A}^{(k)}$ corresponds to all possible signal points transmitting over subcarrier k , satisfying $|\mathcal{A}^{(k)}| = 2^{qd_f+1}$.
- **TCM-based Set Partitioning**: For each subcarrier k , a signal point in \mathcal{A}_k is assigned to a coded sequence $\mathbf{c}^{(k)}$ such that the MFSE distance can be maximized. To construct the mapping based on the above criterion, we adopt the set partitioning technique in which a binary partitioning tree is utilized [19]. In each level of the tree, the current signal set is divided into two subsets such that the minimum squared subset distance (MSSD), that is, the MSE distance between signal points in the same subset, is maximized. In the last level of the tree, the leaf node refers to a constellation point. Each constellation point can be reached via a unique path through the tree.

Overall Set Partitioning Process: To further maximize the MFSE distance of the coding scheme with respect to each subcarrier, we consider adopting the binary-tree based set partitioning technique in the first step as well. This is to say, given \mathcal{A}_0 , we aim to maximize the MSSD of each subset $\mathcal{A}^{(k)}$. For the system shown in Fig. 1 and Fig. 2, the signal set labeling can be described as below. We first construct a super binary partitioning tree of 2^9 levels with the root node \mathcal{A}_0 . The first level of the tree consists of two subsets divided from \mathcal{A}_0 , denoted by \mathcal{A}_1^1 and \mathcal{A}_1^2 . For level $2 \leq j \leq 2^9$ of the tree, each branch node \mathcal{A}_j^i represents a point set which will be divided into two subsets in the next level, i.e., \mathcal{A}_{j+1}^{2i-1} and \mathcal{A}_{j+1}^{2i} . The signal sets obtained after the subcarrier-based set partitioning can be found in level 2 of the tree, i.e., $\{\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(4)}\} = \{\mathcal{A}_2^1, \dots, \mathcal{A}_2^4\}$. In the last level of the tree, we have $\mathcal{A}_9^l = \{A_l\}$, $A_l \in \mathcal{A}_0$, i.e., each leaf node represents a specific signal point. The coded sequence mapped to this point is labeled by the path towards

this leaf node. For example, we have $\mathcal{A}_9^1 = \{A_1\}$ implying that the coded sequence 0000000 transmitted over subcarrier 1 is mapped to the signal point $A_1 \in \mathcal{A}^{(1)}$.

Based on the binary subtree for each subcarrier k , $\mathcal{A}^{(k)}$, we observe that δ_{\min}^2 can be obtained by inspecting the minimum MSSD of the subsets in the $(r+1)$ th level of the subtree, i.e., the $(p+r+1)$ th level of the binary tree in Fig. 2.

$$\delta_{\min}^2 = \min_{1 \leq i \leq 2^{r+1}} \left\{ \Delta \left(\mathcal{A}_{p+r+1}^{(k-1) \cdot 2^{r+1} + i} \right) \right\}. \quad (5)$$

Basic Bipartite Set Partitioning Operation: Note that the basic operation in the above process is to divide a signal set into two subsets with the equal size. We now formulate this basic bipartite set partitioning problem and propose a novel algorithm to solve it.

a) Problem Formulation: Given a signal set \mathcal{A}_b , we aim to divide it into two subsets \mathcal{A}_b^1 and \mathcal{A}_b^2 with equal size such that the MSSD is maximized. The MSSD of each subset \mathcal{A}_b^j is given by

$$\Delta \left(\mathcal{A}_b^j \right) = \min_{l, l' \in \mathcal{A}_b^j, l \neq l'} \|z_l - z_{l'}\|_2^2, j = 1, 2. \quad (6)$$

Therefore, an optimal bipartite set partitioning of \mathcal{A}_b , i.e., $\mathcal{A}_b \rightarrow \{\mathcal{A}_b^1, \mathcal{A}_b^2\}$, can be obtained by solving the following problem:

$$\max_{\{\mathcal{A}_b^1, \mathcal{A}_b^2\}} \left[\min_{1 \leq j \leq 2} \Delta \left(\mathcal{A}_b^j \right) \right] \quad (7a)$$

$$\text{s.t. } |\mathcal{A}_b^j| = \frac{1}{2} |\mathcal{A}_b|, j = 1, 2, \quad (7b)$$

$$\mathcal{A}_b^j \subset \mathcal{A}_b, j = 1, 2, \quad (7c)$$

$$\Delta \left(\mathcal{A}_b^j \right) > 0, j = 1, 2. \quad (7d)$$

This is a non-trivial problem due to its combinatorial nature and the irregular positions of the points. The traditional multi-level coding technique [24] which is originally designed for partitioning lattice or star constellations does not fit this case any more. We then propose a modified farthest point optimization algorithm to address this problem, as illustrated below.

b) Algorithm Design: We observe that the distribution of the points in our formulated problem has the *blue noise* properties, i.e., i) the signal points cover a certain area and there are no “holes” or “clusters” in the 2D plane; ii) the points in the selected set are distributed almost irregularly. Such properties have drawn great attention in the researches on point set generation [21] where a farthest point optimization (FPO) strategy [20] has been utilized for generating point distributions with high-quality “blue noise” characteristics, i.e., large point spacing in a given area.

Instead of point generation, we aim to select points for each subset given the point distribution. By extending the FPO strategy, we then propose a novel bipartite set partitioning (FPO-BSP) algorithm to solve problem (7), consisting of two phases: i) initial subset construction; ii) FPO iteration. Details of the algorithm are shown in Algorithm 1 and are illustrated as below.

For convenience, we first present three different distance metrics. Given a signal point A_l , its minimum distance to a set \mathcal{A}_x is defined as

$$d_l(\mathcal{A}_x) = \min_{A_r \in \mathcal{A}_x, r \neq l} |z_l - z_r|. \quad (8)$$

Based on this definition, the MSSD of a set \mathcal{A}_x in (6) can be rewritten by

$$\Delta(\mathcal{A}_x) = \min_{A_l \in \mathcal{A}_x} d_l(\mathcal{A}_x). \quad (9)$$

The average minimum distance of a set \mathcal{A}_x , denoted by $\bar{d}(\mathcal{A}_x)$, can be obtained based on each point's minimum distance, i.e.,

$$\bar{d}(\mathcal{A}_x) = \frac{1}{|\mathcal{A}_x|} \sum_{A_l \in \mathcal{A}_x} d_l(\mathcal{A}_x). \quad (10)$$

Since we tend to divide the target signal set into two subsets where the points are spread out as far as possible, both the MSSD and the average minimum distance of the subsets are encouraged to be maximized.

i) Initial Subset Construction: As shown in Phase 1 of Algorithm 1, we adopt a greedy method to divide the target set into two initial subsets. For the target set \mathcal{A}_b of size $|\mathcal{A}_b|$, there are $L = C_{|\mathcal{A}_b|}^2$ pairs of points in total. We sort these pairs in the increasing order of the distance between two points (line 2 in Algorithm 1). The list Γ can be mathematically presented as

$$\Gamma = \{(A_{\alpha(1)}, A_{\beta(1)}), (A_{\alpha(2)}, A_{\beta(2)}), \dots, (A_{\alpha(L)}, A_{\beta(L)})\}, \quad (11)$$

satisfying that

$$|z_{\alpha(j)} - z_{\beta(j)}| \leq |z_{\alpha(j')} - z_{\beta(j')}|, 1 \leq j \leq j' \leq L. \quad (12)$$

As shown in line 3 of Algorithm 1, we initialize two subsets as $\mathcal{A}_b^1 = \{A_{\alpha(1)}\}$ and $\mathcal{A}_b^2 = \{A_{\beta(1)}\}$. The remaining pairs of points in list Γ are then traversed to be added into different subsets (line 5-21). To be specific, we consider a pair $(A_{\alpha(j)}, A_{\beta(j)})$. If both points never show up in the current subsets, we then add $A_{\alpha(j)}$ and $A_{\beta(j)}$ separately into $\mathcal{A}_b^{\tilde{\alpha}}$ and $\mathcal{A}_b^{\tilde{\beta}}$ in which $\tilde{\alpha}, \tilde{\beta} \in \mathbb{Z}^+$ such that the minimum distance between the members of this pair and the subsets can be maximized (line 15-17), i.e.,

$$\min \left\{ d_{\alpha(j)}(\mathcal{A}_b^{\tilde{\alpha}}), d_{\beta(j)}(\mathcal{A}_b^{\tilde{\beta}}) \right\} \geq \min \left\{ d_{\alpha(j)}(\mathcal{A}_b^{\tilde{\beta}}), d_{\beta(j)}(\mathcal{A}_b^{\tilde{\alpha}}) \right\}, \quad \tilde{\alpha} + \tilde{\beta} = 3. \quad (13)$$

However, when the distance between $A_{\alpha(j)}$ and $A_{\beta(j)}$ is larger than the MSSD of current subsets, i.e., $|z_{\alpha(j)} - z_{\beta(j)}| > \max\{\Delta(\mathcal{A}_b^1), \Delta(\mathcal{A}_b^2)\}$, then it is not necessary that $A_{\alpha(j)}$ and $A_{\beta(j)}$ are divided into two subsets. Each of them selects a subset with a large minimum distance to join (line 11-13). If one of the points in the target pair has already been added in a subset in previous operations, then the other point is naturally added to another subset (line 19-21). The whole process ends when the size of one subset reaches $|\mathcal{A}_b|/2$. The left unchosen points in list Γ are then added to the other set which has not been fully filled (line 6-9).

ii) FPO Iteration: Based on the above initialized subsets, we then perform the FPO iteration as shown in Phase 2 of

Algorithm 1. The key idea of the FPO method is illustrated as below. Given two initial subsets \mathcal{A}_b^1 and \mathcal{A}_b^2 , we consider replacing each point in subset \mathcal{A}_b^1 with the farthest position for this subset, which is selected from the other subset \mathcal{A}_b^2 . The farthest position⁴ for \mathcal{A}_b^1 , defined as $A_{r^*} \in \mathcal{A}_b^2$, can be obtained by searching \mathcal{A}_b^2 ,

$$r^* = \arg \max_{r \in \mathcal{A}_b^2} d_r(\mathcal{A}_b^1). \quad (14)$$

For each point $A_l \in \mathcal{A}_b^1$, it is first removed from \mathcal{A}_b^1 and inserted into \mathcal{A}_b^2 (line 28). We then search for the farthest position for $\mathcal{A}_b^1 \setminus \{A_l\}$ from the set $\mathcal{A}_b^2 \cup \{A_l\}$ based on equation (14) (line 29-34). When all points in \mathcal{A}_b^1 are traversed once, one FPO iteration is finished (line 25-35). During multiple FPO iterations, both the MSSD and the average minimum distance of the subsets will be increasing until convergence, which will be proved in detail in Proposition 1.

Delaunay Triangulation for FPO: Note that in the FPO method, a large amount of operations such as point searching, removing, and inserting are required. To achieve a low computational complexity, we introduce the Delaunay triangulation (DT) method to construct a dynamic graph in which the relative positions of points can be better depicted and easily traced.

Definition 1: Given a point set \mathcal{A}_x , a DT refers to a triangulation D such that no point in \mathcal{A}_x lies inside the circumcircle of any triangle in D . Any edge in a DT is called a Delaunay edge.

In a DT, each triangle follows the property of empty circumcircle. One commonly used method for constructing such a triangulation is the on-line DT method [22]. Starting from a certain point in the set, the neighboring points are inserted one by one to form the triangles while the property of empty circumcircle is guaranteed. Given a formulated triangulation D , point inserting and removing can be completed flexibly via the local optimization procedure in which Delaunay edges are added or removed within a localized area.

Remark 1: In Algorithm 1, we refer to the above operations as DT-INSERT (D, A_l) and DT-REMOVE (D, A_l), implying the point A_l inserting into D and removing from D , respectively.

Remark 2: In each FPO iteration, suppose that the DT constructed from the set \mathcal{A}_x is D . The minimum distance between a point A_l and \mathcal{A}_x can be obtained by searching the Delaunay neighbors⁵ of A_l in D instead of searching all points in \mathcal{A}_x . We refer to this operation as DT-SEARCH (D, A_l).

Algorithm Interpretation of Phase 2: Algorithm 1 with respect to DT can be re-interpreted in detail as below. Following the on-line DT method, the triangulations of two initial point subsets \mathcal{A}_b^1 and \mathcal{A}_b^2 are constructed, denoted by D_1 and D_2 . As shown in Phase 1, points are added into two subsets in sequence (line 4, 14, 18, 22). In each FPO iteration, for each vertex $A_l \in D_1$, the minimum distance between A_l and

⁴Different from the researches on point distribution generating [20], we cannot generate new points in the given set, and thus, we redefine the farthest position as the farthest position in the existing points of the set.

⁵The Delaunay neighbors of A_l refer to those points in D sharing the same edges with A_l .

\mathcal{A}_b^1 , i.e., $d_l(\mathcal{A}_b^1)$, can be obtained by searching the Delaunay neighbors of A_l in D_1 (line 26). After recording the current minimum distance $d_{\max} = d_l(\mathcal{A}_b^1)$, vertex A_l is removed from D_1 and inserted into D_2 (line 27-29). We then traverse the vertices in the newly constructed D_2 to search for the farthest position of D_1 (line 30-36). If the MSSD of \mathcal{A}_b^2 will not decrease and $d_{r^*}(\mathcal{A}_b^1 \setminus \{A_l\})$ will not be smaller than the MSSD of \mathcal{A}_b^1 , A_l is then replaced by the farthest position of D_1 (line 35-37). The sizes of two subsets maintain to be equal since we just swap a new point in D_2 with an old one in D_1 . The FPO iterations will not stop until no changes can be made to D_1 and D_2 (line 38).

c) Analysis of the Proposed Algorithm: The convergence and complexity of the FPO-BSP algorithm are analyzed as below. The proof of Proposition 1 can be found in Appendix A.

Proposition 1: In each FPO iteration of Algorithm 1, the average minimum distance and MSSD of subset \mathcal{A}_b^1 are increasing, and the MSSD of subset \mathcal{A}_b^2 are non-decreasing. Therefore, Phase 2 is guaranteed to converge.

Remark 3: Since the MSSD of two initial subsets of Phase 1 are usually different, we set the subset with a smaller MSSD as \mathcal{A}_b^1 and the other one as \mathcal{A}_b^2 , and then send them to Phase 2. Based on Proposition 1, the MSSD of \mathcal{A}_b^1 and \mathcal{A}_b^2 will be more balanced after the iterations.

As discussed in Appendix B, the complexity of Phase 1 is $O(|\mathcal{A}_b|^2 \log |\mathcal{A}_b|)$ and the complexity of each FPO iteration in Phase 2 is $O(|\mathcal{A}_b| \log |\mathcal{A}_b|)$.

3) *Convolutional Encoder Design:* As mentioned in Subsection II.A.2, a convolutional encoder is adopted in the TCM-based NOMA scheme to generate $(r+1)$ coded bits. We now illustrate how to design the rate $r/(r+1)$ and the structure of the convolutional encoder.

As shown in equation (5) and Fig. 2, rate $r/(r+1)$ determines the level of set partitioning $p+r+1$ in a way that the value of δ_{\min}^2 increases with r . Since the decoding complexity also grows with r , a trade-off should be reached between δ_{\min}^2 and the complexity.

For the different rate $r/(r+1)$ with $1 \leq r \leq qd_f - 2$, the optimal convolutional encoder that maximizes δ_{free}^2 is different. For simplicity, we assume that the same convolutional encoder of rate $r/(r+1)$ is designed for different subcarriers. The diagram of a systematic feedback convolutional encoder (Fig. 18-16 in [19]) is adopted. Note that in the traditional MD-TCM scheme, the distance between any two nodes A_l and $A_{l'}$ is $\sum_{i=1}^{d_f} |z_{li} - z_{l'i}|$. This can be considered as an upper bound of the distance between these two nodes in our proposed TCM-based NOMA scheme since we have

$$\sum_{i=1}^{d_f} |z_{li} - z_{l'i}| \geq \left| \sum_{i=1}^{d_f} (z_{li} - z_{l'i}) \right|. \quad (15)$$

Therefore, we adopt the structure of the convolutional encoder for $d_f \times \text{M-PSK/QAM}$ in the MD-TCM scheme [19]. For a given rate $r/(r+1)$, the value of δ_{free}^2 increases with the number of register states, V , in the encoder. However, a large value of V increases the decoding complexity as well,

Algorithm 1 FPO-based bipartite set partitioning (FPO-BSP) algorithm

Input: a point set \mathcal{A}_b of size N

Output: two subsets \mathcal{A}_b^1 and \mathcal{A}_b^2 of size $N/2$ with maximized MSSD

```

1: Phase 1: Initial Subset Construction
2: Construct a list  $\mathcal{L}$  based on equations (11) and (12).
3: Set  $\mathcal{A}_b^1 = \{A_{\alpha(1)}\}$  and  $\mathcal{A}_b^2 = \{A_{\beta(1)}\}$ .
4: Initialize  $D_1 = \{A_{\alpha(1)}\}$  and  $D_2 = \{A_{\beta(1)}\}$ .
5: for  $j = 2$  to  $L$  do
6:   if  $|\mathcal{A}_b^1| = N/2$  or  $|\mathcal{A}_b^2| = N/2$  then
7:     Add remained points to the subset not fully filled.
8:     break;
9:   else if  $\{A_{\alpha(j)}, A_{\beta(j)}\} \not\subset \mathcal{A}_b^1 \cup \mathcal{A}_b^2$  then
10:    if  $|p_{\alpha(j)} - p_{\beta(j)}| > \max\{\Delta(\mathcal{A}_b^1), \Delta(\mathcal{A}_b^2)\}$  then
11:      For each point  $A_l \in \{A_{\alpha(j)}, A_{\beta(j)}\}$ , set  $i^* = \arg \max_{i=1,2} d_l(\mathcal{A}_b^i)$ .
12:       $A_l$  joins the subset  $\mathcal{A}_b^{i^*}$ .
13:    else if  $A_{\alpha(j)}, A_{\beta(j)} \notin \mathcal{A}_b^1 \cup \mathcal{A}_b^2$  then
14:      Find  $\tilde{\alpha}$  and  $\tilde{\beta}$  that satisfies inequality (13).
15:      Add  $A_{\alpha(j)}$  and  $A_{\beta(j)}$  into  $\mathcal{A}_b^{\tilde{\alpha}}$  and  $\mathcal{A}_b^{\tilde{\beta}}$ , respectively.
16:    else
17:      Set  $A_l$  as the point in  $\{A_{\alpha(j)}, A_{\beta(j)}\}$  which is not a member of  $\mathcal{A}_b^1 \cup \mathcal{A}_b^2$ .
18:      Add  $A_l$  to the subset of which the other point is not a member, say,  $\mathcal{A}_b^2$ .
19: Construct two DTs  $D_1$  and  $D_2$ .
20: Phase 2: FPO Iteration
21: repeat
22:   for each vertex  $A_l$  in  $D_1$  do
23:     Obtain  $d_l(\mathcal{A}_b^1)$  by DT-SEARCH( $D_1, A_l$ ) in Remark 2
24:     Set  $(A_{r^*}, d_{\max}) = (A_l, d_l(\mathcal{A}_b^1))$ 
25:     Set  $\Delta_{\min} = \Delta(\mathcal{A}_b^2)$ 
26:     DT-REMOVE( $D_1, A_l$ ) and DT-INSERT( $D_2, A_l$ ) based on Remark 1
27:   for each vertex  $A_{l'}$  in  $D_2$  do
28:     DT-INSERT( $D_1, A_{l'}$ )
29:     Obtain  $d_{l'}(\mathcal{A}_b^1)$  by DT-SEARCH( $D_1, A_{l'}$ )
30:     Obtain  $d_{l'}(\mathcal{A}_b^2)$  by DT-SEARCH( $D_2, A_{l'}$ )
31:     DT-REMOVE( $D_1, A_{l'}$ )
32:     if  $d_{l'}(\mathcal{A}_b^1) > d_{\max}$  and  $d_{l'}(\mathcal{A}_b^2) \geq \Delta_{\min}$  then
33:       Set  $(A_{r^*}, d_{\max}) = (A_{l'}, d_{l'}(\mathcal{A}_b^1))$ 
34:       DT-REMOVE( $A_{r^*}, D_2$ ), DT-INSERT( $A_{r^*}, D_1$ ).
35: until  $D_1$  and  $D_2$  do not change
36: return  $\mathcal{A}_b^1$  and  $\mathcal{A}_b^2$ 

```

requiring a trade-off between δ_{free}^2 and V . Since the optimal convolutional encoders have always been found by computer search [24], we do not illustrate the detailed process in this paper.

Note that the above method is suitable for the AWGN channels. However, for the Rayleigh fading, the number of parallel branches should be reduced and the encoder state

diagram should be redesigned such that a) the shorted error event path length can be increased [25]; b) the product of the squared branch distance with respect to that path can be maximized [26]. Given an encoder designed for AWGN channels, we can modify the state diagram by reducing the parallel branches based on the above criteria.

IV. TCM-BASED NOMA DECODER DESIGN

Due to the non-orthogonal nature of the NOMA, signals of multiple users are superimposed and transmitted over the same subcarrier, and thus a joint decoding technique is required. Since the convolutional encoding is adopted, an efficient sequence detection should be considered in the decoding scheme. In this section, we design a two-layer Viterbi-based algorithm for the joint TCM-based NOMA decoding in which the soft-decision based MLSD technique is utilized.

A. Criterion for Joint Decoding

We assume that η data sequences are sent into each TCM encoder sequentially, i.e., one sequence per time unit. Denote m as the number of time units required for the encoder registers to be cleared. The set of coded sequences is $(\mathbf{c}^{(k)}[1], \mathbf{c}^{(k)}[2], \dots, \mathbf{c}^{(k)}[\eta + m])$. In the traditional ML detection, the decoder should produce a set of estimated $\hat{\mathbf{c}}^{(k)}$ of $\mathbf{c}^{(k)}$ given the received sequence such that $\sum_{t=1}^{\eta+m} \log P(y^{(k)}[t]|\mathbf{c}^{(k)}[t])$ can be maximized.

However, due to the multiplexing nature of NOMA, the input data sequences of different encoders overlap with each other, which can be utilized for joint decoding. For example, given the mapping matrix in (2), the data streams of users 1, 2, and 3 are coded and transmitted over subcarrier 1, and that of users 1, 4, and 5 over subcarrier 2. A successful decoding scheme requires that the decoded data sequence of the same user obtained from different subcarriers in the one time unit should be the same. In other words, the data sequence of user 1 decoded from the first two subcarriers, i.e., $\mathbf{b}_1^{(1)}[t]$ and $\mathbf{b}_1^{(2)}[t]$, should be identical and the same is true with other users. Mathematically, we have

$$\mathbf{b}_j^{(k)}[t] = \mathbf{b}_j^{(k')}[t], \text{ if } j = \mathcal{J}^{(k)} \cap \mathcal{J}^{(k')}, 1 \leq k, k' \leq K. \quad (16)$$

Based on the above idea, we consider to adopt the cross check in the ML detection for joint NOMA decoding. Given the encoder state of time unit $(t-1)$ and the input data sequence of time unit t , a TCM encoder only produce one out of $2^{q_{df}+1}$ possible code sequences as the output. To depict this, we introduce a binary probability variable $P(\mathbf{c}|\mathbf{s}^{(k)}[t-1], \mathbf{b}^{(k)}[t])$ in which $\mathbf{s}^{(k)}[t-1]$ is the encoder state in time unit $(t-1)$. Given $\mathbf{s}^{(k)}[t-1]$ and $\mathbf{b}^{(k)}[t]$, the probability of obtaining \mathbf{c} over subcarrier k is

$$P(\mathbf{c}|\mathbf{s}^{(k)}[t-1], \mathbf{b}^{(k)}[t]) = \begin{cases} 1, & \text{if } \mathbf{c} \text{ is the output of} \\ & \text{the encoder in unit } t, \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

By combining condition (16) and definition (17), we present the following proposition.

Proposition 2: Given the encoder states $\{\mathbf{s}^{(k)}[t-1] | 1 \leq k \leq K\}$ in time unit $t-1$, if K code sequences in time unit t are correctly estimated, then the following condition is satisfied:

$$\prod_{k=1}^K P(\hat{\mathbf{c}}^{(k)}[t]|\mathbf{s}^{(k)}[t-1], \mathbf{b}^{(k)}[t]) = 1, \forall 1 \leq t \leq \eta + m, \quad (18)$$

in which the kit of data sequences $\mathbf{B}[t] = \{\mathbf{b}^{(k)}[t] | 1 \leq k \leq K\}$ satisfies condition (16).

For convenience, we denote a kit of code sequences as $\hat{\mathbf{C}}[t] = \{\hat{\mathbf{c}}^{(k)}[t] | 1 \leq k \leq K\}$. Since there are K encoders, we define the super encoder state in time unit t as the combination of encoder states of all K encoders, i.e., $\mathbf{S}[t] = \{\mathbf{s}^{(1)}[t], \mathbf{s}^{(2)}[t], \dots, \mathbf{s}^{(K)}[t]\}$. Proposition 2 provides a necessary condition, and thus, there may exist multiple *qualified* kits of code sequences in each time unit t . For any possible super encoder state $\mathbf{S}[t-1]$, the set of qualified kits of coded sequences satisfying Proposition 2 in time unit t is contained in $\mathcal{C}[t]$.

The criterion for joint NOMA decoding can then be described as below. The decoder is required to estimate $\hat{\mathbf{c}}^{(k)}$ given the received sequence \mathbf{y} by optimizing the following problem:

$$\max_{\hat{\mathbf{C}}[t]} \sum_{t=1}^{\eta+m} \sum_{k=1}^K \log P(y^{(k)}[t]|\hat{\mathbf{c}}^{(k)}[t]) \quad (19a)$$

$$\text{s.t. } \hat{\mathbf{C}}[t] \in \mathcal{C}[t], \quad (19b)$$

in which $\sum_{k=1}^K \log P(y^{(k)}[t]|\hat{\mathbf{c}}^{(k)}[t])$ is called the branch metrics.

B. Joint Decoding Scheme Design

To solve the ML detection problem in (19), we first present an optimal solution extended from the Viterbi algorithm [27]. To obtain a more practical solution with tolerable complexity, we then design a suboptimal two-layer Viterbi-based decoding algorithm.

1) *Optimal Maximum Likelihood Decoding Scheme:* For each encoder, we assume that the number of registers is V , and thus there are 2^V encoder states. Since there are K encoders, we have 2^{KV} super encoder states in total. A super encoder state diagram can then be constructed based on the 2^{KV} super encoder states. Considering the cross check requirement of the NOMA decoder, we assume that one super encoder state can only transfer to another if the input data sequences and output coded sequences, i.e., $\mathbf{B}[t]$ and $\mathbf{C}[t]$, satisfy Proposition 2.

Given the super encoder state diagram, the soft-decision based Viterbi algorithm [27] can then be performed. For each super encoder state, we aim to update the survivor path in each time unit. A survivor path in time unit t refers to a record consisting of a series of super encoder states $\mathbf{S}[t']$, $1 \leq t' \leq t$ and the branch between every two sequent states $\mathbf{S}[t'-1]$ and $\mathbf{S}[t']$. A *branch* is denoted by the estimated input data sequences and output code sequences, i.e., $\mathbf{B}[t']$ and $\mathbf{C}[t']$.

Mathematically, a *survivor path* with respect to $\mathbf{S}[t]$ can be given by

$$\mathbf{S}[0] \xrightarrow{\mathcal{B}[1], \mathcal{C}[1]} \mathbf{S}[1] \rightarrow \dots \rightarrow \mathbf{S}[t'] \xrightarrow{\mathcal{B}[t'+1], \mathcal{C}[t'+1]} \mathbf{S}[t'+1] \rightarrow \dots \rightarrow \mathbf{S}[t]. \quad (20)$$

To update the survivor path for each $\mathbf{S}[t]$, we utilize the branch metrics in (19a) to select a state $\mathbf{S}[t-1]$ as well as the branch bridging $\mathbf{S}[t-1]$ and $\mathbf{S}[t]$. Specifically, the branch metrics in each time unit t can be depicted by the total Euclidean distances between K received signals and the signal points mapped to the candidate code sequences, i.e., $\sum_{k=1}^K d(y^{(k)}[t], g(\mathbf{c}^{(k)}[t]))$, in which $g(\cdot)$ maps a coded sequence to the output signal point in the constellation.

In the last time unit $\eta + m$, we select the shortest survivor path terminated with an all-zero super encoder state as the final path. The optimality of this cross-check based soft-decision Viterbi algorithm can be guaranteed since the final survivor path is always the ML path (Theorem 12.1 in [19]).

2) Suboptimal Two-layer Viterbi-based Joint Decoding Scheme: In the optimal decoding algorithm, for each time unit the decoder is required to traverse all 2^{KV} super encoder states and possible branches to update the survivor path for each super encoder state, leading to a prohibitively high computation complexity. To reduce the complexity, we consider only checking the most possible super encoder states and branches to update no more than λ survivor paths in each time unit in the suboptimal decoding scheme.

a) Inner Layer Soft-decision Viterbi Operation: For convenience, we denote the set of stored super encoder states corresponding to the survivor paths in time unit $(t-1)$ as $\mathcal{S}[t-1] = \{\mathbf{S}_1[t-1], \mathbf{S}_2[t-1], \dots, \mathbf{S}_\lambda[t-1]\}$. For each $\mathbf{S}_i[t-1]$, we first illustrate how to select the candidate branches based on the received signals over K subcarriers in time unit t . The inner layer operation is performed over each subcarrier k separately.

For a received signal point $y^{(k)}[t]$, most candidate output signal points lie in the neighborhood of the received signal point since the noise power obeys the Gaussian distribution with a given variance σ^2 . Therefore, instead of recoding all the signal points corresponding to potential branches, we only record those lying in the neighborhood⁶.

To be specific, for each subcarrier k , given the current encoder state $\mathbf{s}_i^{(k)}[t-1]$, the set of all possible input data sequences is denoted as \mathcal{B} and $|\mathcal{B}| = 2^{qd_f}$. The set of all possible output coded sequences can be denoted as $\mathcal{C}_{poss,i}^{(k)}[t]$ such that each $\mathbf{c}^{(k)}[t] \in \mathcal{C}_{poss,i}^{(k)}[t]$ satisfies that

$$P(\mathbf{c}^{(k)}[t] | \mathbf{s}_i^{(k)}[t-1], \mathbf{b}^{(k)}) = 1, \forall \mathbf{b}^{(k)} \in \mathcal{B}. \quad (21)$$

The candidate output signal points over subcarrier k can then be obtained by

$$\mathcal{A}_{cand,i}^{(k)}[t] = \left\{ A_l \in \mathcal{A}_k | d(y^{(k)}[t], A_l) \leq a\sigma^2, A_l = g(\mathbf{c}^{(k)}[t]), \mathbf{c}^{(k)}[t] \in \mathcal{C}_{poss,i}^{(k)}[t] \right\}, \quad (22)$$

⁶The neighborhood of the received signal point can be defined by a circle whose center is this signal point.

where $d(y_t^{(k)}, A_l)$ is the Euclidean distance between $y_t^{(k)}$ and the signal point A_l , a is the radius parameter. Based on the definitions of $\mathcal{A}_{cand,i}^{(k)}[t]$ and $\mathcal{C}_{poss,i}^{(k)}[t]$, we observe that each $A_l \in \mathcal{A}_{cand,i}^{(k)}[t]$ corresponds to an output coded sequence and an input data sequence. For convenience, we denote them as members in the set of candidate coded sequences $\mathcal{C}_{cand,i}^{(k)}[t]$ and the set of candidate data sequences $\mathcal{B}_{cand,i}^{(k)}[t]$, respectively.

b) Outer Layer Soft-decision Viterbi Operation: In time unit t , given a stored super encoder state $\mathbf{S}_i[t-1]$ and the outcome from the inner layer operation, we perform the cross check to obtain the survivor paths for time unit t .

For each super encoder state $\mathbf{S}_i[t-1]$, multiple candidate branches are generated based on $\mathcal{C}_{cand,i}^{(k)}[t]$ and $\mathcal{B}_{cand,i}^{(k)}[t]$. Specifically, a candidate branch $(\mathbf{C}_{cand,i}[t], \mathbf{B}_{cand,i}[t])$ can be constructed by

$$\begin{aligned} \mathbf{C}_{cand,i}[t] &= \left\{ \mathbf{c}^{(k)}[t] | \mathbf{c}^{(k)}[t] \in \mathcal{C}_{cand,i}^{(k)}[t], 1 \leq k \leq K \right\} \\ \mathbf{B}_{cand,i}[t] &= \left\{ \mathbf{b}^{(k)}[t] | \mathbf{b}^{(k)}[t] \in \mathcal{B}_{cand,i}^{(k)}[t], 1 \leq k \leq K \right\}, \end{aligned} \quad (23)$$

in which $P(\mathbf{c}^{(k)}[t] | \mathbf{s}_i^{(k)}[t-1], \mathbf{b}^{(k)}) = 1$ for $1 \leq k \leq K$.

The set $\mathcal{C}[t]$ (defined in Section III.A) can then be constructed by performing the cross check on the candidate branches. If a candidate branch $(\mathbf{C}_{cand,i}[t], \mathbf{B}_{cand,i}[t])$ satisfies Proposition 2, we refer to it as a *qualified branch*. For the qualified branch, we then add $\mathbf{C}_{cand,i}[t]$ into the set $\mathcal{C}[t]$ and record the corresponding super encoder state $\mathbf{S}_i[t-1]$.

Given the set $\mathcal{C}[t]$, we then select at most λ survivor paths for time unit t based on the total path lengths. For each $\mathbf{C}_{cand,i}[t] \in \mathcal{C}[t]$, we can obtain the super encoder state $\mathbf{S}[t]$ based on the corresponding input $\mathbf{B}_{cand,i}[t]$ and the previous state $\mathbf{S}_i[t-1]$. The path length with respect to $\mathbf{S}[t]$ is then calculated as

$$\begin{aligned} PLength(\mathbf{S}[t]) &= PLength(\mathbf{S}_i[t-1]) \\ &+ \sum_{k=1}^K d(y^{(k)}[t], g(\mathbf{c}^{(k)}[t])), \end{aligned} \quad (24)$$

in which $\mathbf{c}^{(k)}[t] \in \mathbf{C}_{cand,i}[t]$ and $PLength(\mathbf{S}_i[t-1])$ is the history total path length with respect to $\mathbf{S}_i[t-1]$.

Based on the above equation, we select up to λ code sequences from $\mathcal{C}[t]$ and the corresponding survivor paths with respect to various super encoder states $\mathbf{S}[t]$.

c) Overall Suboptimal Decoding Scheme: The whole algorithm is presented in detail in Algorithm 2. In each time unit t , we aim to update the survivor paths based on the stored paths in time unit $t-1$ and the received signals in time unit t . Specifically, for each subcarrier k and each stored encoder state $\mathbf{s}_i^{(k)}[t-1]$ ($1 \leq i \leq \lambda$), we first perform an inner layer soft-decision operation (line 5-9). The sets of candidate output coded sequences and input bit sequences are selected and temporally stored given the received signal $y^{(k)}[t]$. We then perform the outer layer soft-decision Viterbi operation (line 10-20) in which the set of qualified branches satisfying Proposition 2 is selected based on the candidate coded sequences and bit sequences. By sorting the path lengths with respect to these selected branches, we update up to λ survivor paths and super encoder states in time unit t with the shortest path lengths (line 19-20). After $\eta + m$ time units, the

shortest survivor path terminating at the all-zero super encoder state is selected as the final path (line 21-22). Different from the traditional Viterbi algorithm, the decoded bit sequence is already stored in each time unit, and thus, it can be directly obtained from the final path.

Algorithm 2 Two-layer Soft-decision Viterbi Algorithm for Joint NOMA Decoding

Input: K received signal sequences $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(K)}$ of length $\eta + m$, the TCM encoder state diagram, NOMA mapping matrix \mathbf{F} , noise variance σ^2 , channel coefficients

Output:

J decoded bit sequences for J users

- 1: **Initialization:**
 - 2: Set $\mathbf{S}[0]$ as the all-zero super encoder state
 - 3: Set $\mathcal{S}[0] = \{\mathbf{S}[0]\}$
 - 4: **for** $t = 1$ to $\eta + m$ **do**
 - 5: Set $\mathcal{C}[t] = \emptyset$
 - 6: **for** $i = 1$ to $|\mathcal{S}[t-1]|$ **do**
 - 7: **for** $k = 1$ to K **do**
 - 8: Construct the set of candidate output signal points $\mathcal{A}_{cand,i}^{(k)}[t]$ according to (22)
 - 9: Obtain the candidate sets $\mathcal{C}_{cand,i}^{(k)}[t]$ and $\mathcal{B}_{cand,i}^{(k)}[t]$ based on (21) and (22)
 - 10: **repeat**
 - 11: Construct the candidate branch $(\mathbf{C}_{cand,i}^{(k)}[t], \mathbf{B}_{cand,i}^{(k)}[t])$ from $\mathcal{C}_{cand,i}^{(k)}[t]$ and $\mathcal{B}_{cand,i}^{(k)}[t]$ according to (23)
 - 12: **if** the candidate branch never shows up and satisfies Proposition 2 **then**
 - 13: Add $\mathbf{C}_{cand,i}^{(k)}[t]$ to $\mathcal{C}[t]$
 - 14: Record $\mathbf{B}_{cand,i}^{(k)}[t]$ and $\mathbf{S}_i[t-1]$
 - 15: **until** there is no new candidate branch any more
 - 16: **for each** $\mathbf{C}[t] \in \mathcal{C}[t]$ **do**
 - 17: Obtain $\mathbf{S}[t]$ based on the recorded $\mathbf{B}_{cand}[t]$ and $\mathbf{S}[t-1]$
 - 18: Calculate the path length $PLength(\mathbf{S}[t])$ according to (24)
 - 19: Sort the obtained path lengths and update the first λ paths with the shortest lengths
 - 20: as survivor paths (shown in (20))
 - 21: Select the shortest survivor path in time unit $\eta + m$ which terminates at the all-zero super
 - 22: encoder state as the final path
 - 23: **return** the decoded bit sequences corresponding to the final path
-

d) Computational Complexity Analysis: The computational complexity of Algorithm 2 in each time unit t comes from three parts:

- *Inner layer operation (line 6-9):* For each super encoder state $\mathbf{S}_i[t-1]$ and each subcarrier k , at most 2^{qd_f} bit sequences are checked to construct $\mathcal{C}_{cand,i}^{(k)}[t]$ and $\mathcal{B}_{cand,i}^{(k)}[t]$ based on (21) and (22). Since there are K subcarriers and at most λ stored super encoder states in time unit t , the

complexity of the inner layer operation is $O(\lambda K 2^{qd_f})$. Note that in a practical NOMA scheme, we usually set $qd_f \leq 6$ such that the complexity is tolerable.

- *Searching for qualified branches (line 10-15):* Suppose that the number of candidate bit sequences for each super encoder state $\mathbf{S}_i[t-1]$ and each subcarrier k is $M_i^{(k)}$ and $M_i^{(k)} \leq 2^{qd_f}$. For each $\mathbf{S}_i[t-1]$, the number of candidate branches is $\prod_{k=1}^K M_i^{(k)}$. The process of checking whether a candidate branch satisfies Proposition 2 can be performed when constructing such a branch. Therefore, the complexity of constructing all qualified branches is $O(\lambda \prod_{k=1}^K M_i^{(k)})$. In practice, the value of $M_i^{(k)}$ is directly influenced by the radius parameter a and σ^2 in (22), as will be discussed in Section V.
- *Updating survivor paths (line 16-20):* Suppose that the number of qualified branches⁷ given a super encoder state $\mathbf{S}_i[t-1]$ is Q_i , i.e., $|\mathcal{C}[t]| = \sum_{i=1}^{\lambda} Q_i$. To update the survivor paths, we need to sort the qualified branches and the complexity can be given by $O\left(\left(\sum_{i=1}^{\lambda} Q_i\right)^2\right)$.

The total complexity of Algorithm 2 can be obtained by adding up the above three parts. We will present detailed results in Fig. 5 in Section V.

V. SIMULATION RESULTS

In this section, we consider a NOMA system as shown in Fig. 1. For the convolutional encoder in AWGN channels, the coefficients of the parity-check sequences in octal form are selected based on the look-up table in [24]. For simplicity, we only assume perfect channel state information at the receiver. The link-level performance of our proposed TCM-based NOMA scheme is compared with the traditional 4-point code-domain NOMA scheme in [10]. To evaluate the design of the MD-constellation for TCM-based NOMA, we also propose a lattice constellation TCM (LC-TCM) based NOMA scheme as comparison, in which the constellation $\mathcal{A}^{(k)}$ constructed for each subcarrier k is the same as the traditional 2D lattice constellation [28] of size 2^{qd_f+1} . Major simulation parameters are listed in Table I.

Fig. 3 and Fig. 4 illustrate the BER performance versus the E_b/N_0 over an AWGN channel with and without channel coding of rate 1/2, respectively. We set $\lambda = 25$ and $a = 5$, and the spectrum efficiency are 3 bits/tone and 1.5 bits/tone, respectively. In both figures, the BER performance of the TCM-based NOMA outperforms that of the traditional code-domain NOMA scheme with high E_b/N_0 and the gap between them increases with E_b/N_0 . When E_b/N_0 is small, the error correcting capacity of the TCM system is restricted due to the high noise level, which is very common in the coded systems. This leads to a worse performance than the traditional scheme. However, as E_b/N_0 increases, the influence of bursty channel errors on the TCM-based NOMA becomes smaller, and thus, the performance gain obtained from the joint coded

⁷The upper bound of Q_i is closely related to the mapping matrix \mathbf{F} . For example, based on \mathbf{F} in (2), we have $Q_{\max} = M_i^{(1)} \cdot \min\{2^{q(d_f-1)}, M_i^{(2)}\}$.
2^q. The detailed proof is omitted due to the limitation of space.

TABLE I
MAJOR SIMULATION PARAMETERS

Parameter	Value
The number of users sharing the same subcarrier d_f	3
The number of non-zero elements in a sparse codeword N	2
The number of bits to be coded for a user in each time unit q	2
Convolutional encoder rate $r/(r+1)$	3/4
MD mother constellation	$d_f \times 16\text{QAM}$
Frame length	1000 bits
Spectral efficiency	1.5 or 3 bis/tone
radius parameter a	4 ~ 6
Maximum number of survivor paths in each time unit λ	5 ~ 35
Number of register states V	4
Interleaver size	32×16 (in symbol)
Maximum Doppler shift in Rayleigh channels	50 Hz
Sampling frequency for Rayleigh channels	1/1800 s

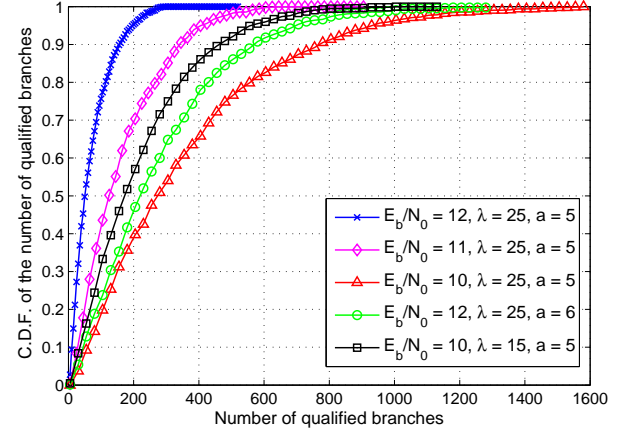


Fig. 5. C.D.F. of the number of qualified branches in Algorithm 2.

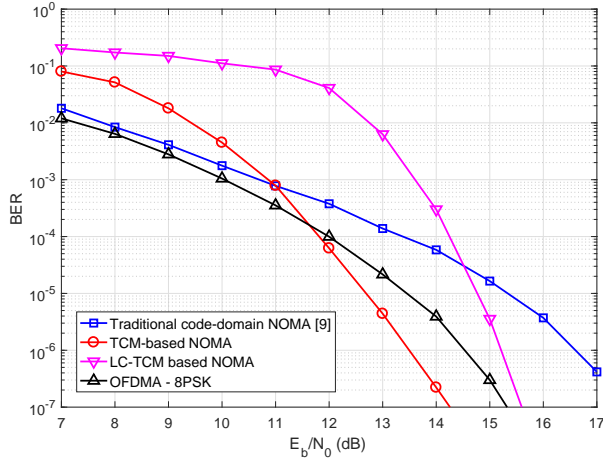


Fig. 3. BER v.s. E_b/N_0 over AWGN channels without channel coding.

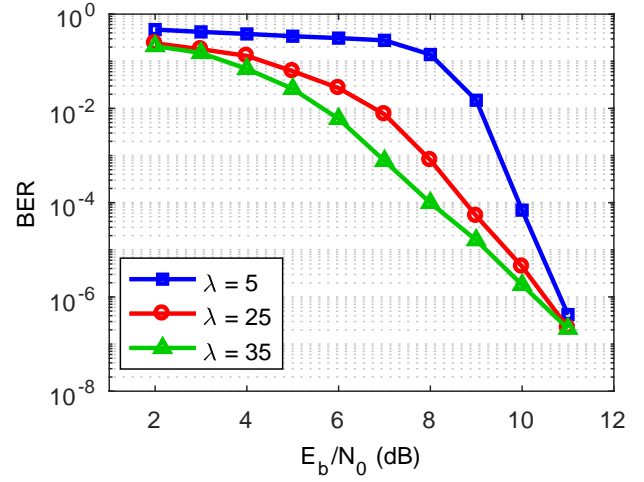


Fig. 6. BER performance v.s. E_b/N_0 with different number of survivor paths.

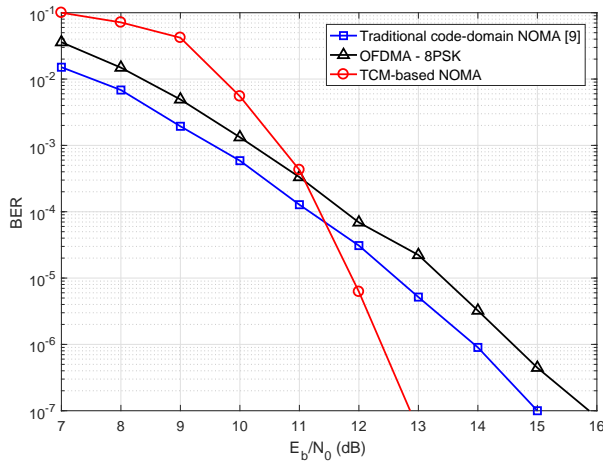


Fig. 4. BER v.s. E_b/N_0 over AWGN channels with convolutional channel coding of rate 1/2.

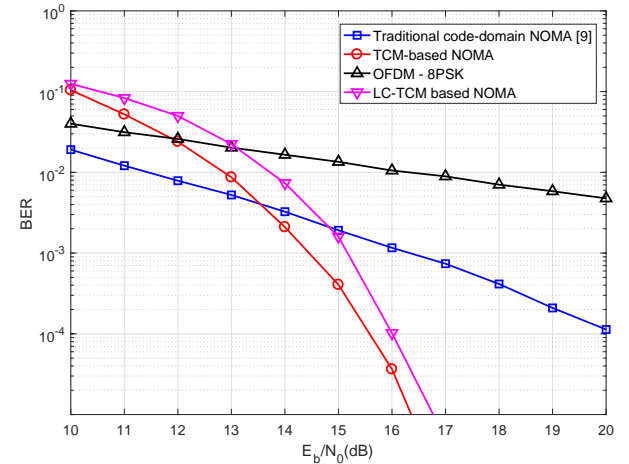


Fig. 7. BER v.s. E_b/N_0 over Rayleigh fading channels.

modulation grows. Fig. 3 also shows that the TCM-based NOMA outperforms the LC-TCM based NOMA scheme. The lattice constellation applied in the LC-TCM based scheme fails

to optimize the positions of multiple MD points, leading to a small MFSE distance of the system and bad performance

compared to the proposed scheme. This implies that the construction of the MD-mother constellation is of vital importance for joint coded modulation.

Specifically, In Fig. 3 we observe that without channel coding, the BER performance of the traditional code-domain NOMA is worse than that of the OFDMA scheme due to the existence of multi-user interference. Benefited from the joint codeword design of different users, our proposed TCM-based NOMA can achieve a better BER performance than both the OFDMA scheme and the traditional NOMA scheme when E_b/N_0 is over 11 dB. This implies that TCM-based NOMA provides a method to compensate for the degraded performance of traditional NOMA when the channel coding is not considered. In Fig. 4, the traditional NOMA scheme outperforms the OFDMA scheme when channel coding is adopted, which is in consistent with existing works on NOMA.

To evaluate the computational complexity of our proposed decoding scheme, we adopt the number of qualified branches for each time unit t as the metrics, i.e., $|\mathcal{C}[t]| = \sum_{i=1}^{\lambda} Q_i$, mentioned in Section IV.B.2.d. Fig. 5 shows the cumulative distribution function (C.D.F.) of $|\mathcal{C}[t]|$, $\Pr(|\mathcal{C}[t]| \leq \tilde{c})$, versus \tilde{c} for different E_b/N_0 and decoding parameters. For a fixed number of limited survivor paths λ and radius parameter a , the complexity grows as E_b/N_0 decreases. A smaller E_b/N_0 implies a higher noise level σ^2 , and thus the number of candidate output signal points grows according to (22), leading to more candidate branches and higher complexity. This also reveals the reason that the complexity increases with the radius parameter a when E_b/N_0 and λ are fixed. Besides, Fig. 5 also shows that the complexity of our proposed decoding scheme decreases with λ becoming smaller.

Fig. 6 shows the BER performance of the TCM-based NOMA scheme versus the E_b/N_0 over an AWGN channel for different numbers of survivor paths λ . As λ increases, the BER performance improves since the probability that a correct path can be found in the survivor paths is larger. Note that when E_b/N_0 is too small or high enough, the increase of λ does not have much influence on the BER performance. When E_b/N_0 is small, it is still hard for the decoder to select the right path from a large number of survivor paths due to interference caused by high noise level. When E_b/N_0 is large enough, the correct path is usually the shortest survivor path for each time unit, and thus, it is not necessary to increase λ any more. Since both the BER performance and the complexity grow with λ , a trade-off can be achieved between these two metrics.

Fig. 7 presents the BER performance versus the E_b/N_0 over a Rayleigh channel with $\lambda = 25$ and $a = 6$. Compared to the case with an AWGN channel, the encoder state diagram is modified as mentioned in Section III.B.3 and the interleaving across the time-frequency resources is performed. Since the number of branches leaving from one state is larger than that of the registers in the convolutional encoder, the shortest error path length is limited. Therefore, when E_b/N_0 is small, the bursty channel errors of the fading channels may be beyond the scope of the system's error control capacity. As E_b/N_0 grows, the performance gain brought by joint encoding exceeds the negative effects caused by the noise level. Thus,

our proposed scheme significantly outperforms the traditional NOMA scheme when E_b/N_0 is larger than 13dB. Fig. 7 also shows that 1) the carefully designed MD constellation in the TCM-based NOMA scheme brings better BER performance than the lattice constellation in the LC-TCM scheme; 2) the traditional code-domain NOMA outperforms the OFDMA scheme in Rayleigh fading environments.

VI. CONCLUSIONS

In this paper, we have proposed a TCM-based NOMA scheme in which the codewords of multiple users are jointly designed based on the MD-TCM techniques, aiming at improving the coding gain of the system. For the encoder design, an MD constellation is constructed and the signal set is selected and labeled to maximize the MFSE distance of the system. To perform the joint decoding, the detection criteria for MLSD are formulated in which the non-orthogonal nature is utilized. A suboptimal two-layer soft-decision Viterbi decoding scheme is then proposed. Simulation results have showed that the TCM-based NOMA scheme performed better than the traditional code-domain NOMA scheme and OFDMA scheme in terms of the BER performance for both the AWGN channels and the Rayleigh fading channels with or without channel coding.

APPENDIX A

PROOF OF PROPOSITION 1

As shown in Algorithm 1, in each FPO iteration, a point $A_l \in \mathcal{A}_b^1$ is swapped with the farthest point of \mathcal{A}_b^1 , i.e., $A_{r^*} \in \mathcal{A}_b^2$. Based on line 35-36 in Algorithm 1, after swapping the minimum distance between A_{r^*} and \mathcal{A}_b^1 is larger than that between A_l and \mathcal{A}_b^1 , i.e.,

$$d_{r^*}(\mathcal{A}_b^1 \cup \{A_{r^*}\} \setminus A_l) > d_l(\mathcal{A}_b^1), \quad (25)$$

and the minimum distance between A_l and \mathcal{A}_b^2 is no smaller than the MSSD of \mathcal{A}_b^2 . Based on (9), we can infer that the MSSD of \mathcal{A}_b^1 is also increasing since the minimum distance of each point in \mathcal{A}_b^1 is increasing. According to (10) and (25), the average minimum distance of the updated subset \mathcal{A}_b^1 , i.e., $\bar{d}(\mathcal{A}_b^1)$, is larger than that of the original subset \mathcal{A}_b^1 after one FPO iteration. Since there always exists an upper bound of $\bar{d}(\mathcal{A}_b^1)$, Phase 2 will stop within a limited number of iterations.

APPENDIX B

PROOF OF COMPLEXITY OF ALGORITHM 1

For convenience, we assume that $|\mathcal{A}_b| = N$. In Phase 1, the complexity of the sorting process in line 2 is $O(L \log L)$ with $L = C_N^2$. The insertion of a point into the DT is $O(1)$, and thus, the complexity of the greedy algorithm (line 5-22) is $O(N)$. Therefore, the complexity of Phase 1 is $O(N^2 \log N)$.

For each FPO iteration of Phase 2, the complexity of the DT-SEARCH operation is $O(g)$ in which g is the number of a point's Delaunay neighbors. It is commonly assumed that $g = O(1)$ for well-distributed point sets [23]. From line 25-37, the complexity is $O(N^2)$ in the worst case. By using a binary search tree to update the vertices in the DT, the complexity of one FPO iteration can be reduced to $O(N \log N)$ [20].

REFERENCES

- [1] B. Di, L. Song, Y. Li, and S. Zhang, "Trellis coded modulation for code-domain non-orthogonal multiple access networks," in *IEEE Int. Conf. Commun. (ICC)*, Kansas, MI, May 2018.
- [2] Z. Ding, L. Dai, and V. Poor, "MIMO-NOMA design for small packet transmission in the Internet of Things," *IEEE Access*, pp. 1393-1405, vol. 4, Apr. 2016.
- [3] B. Di, L. Song, Y. Li, and G. Li, "Non-orthogonal multiple access for high-reliable and low-latency V2X communications in 5G systems," *IEEE J. Sel. Areas Commun.*, pp. 2383-2397, vol. 35, no. 10, Oct. 2017.
- [4] L. Song, Y. Li, Z. Ding, and H. Poor, "Resource management in non-orthogonal multiple access networks for 5G and beyond," *IEEE Network*, pp. 8-14, vol. 31, no. 4, Aug. 2017.
- [5] Z. Ding, P. Fan, and H. Poor, "Impact of user pairing on 5G nonorthogonal multiple-access downlink transmissions," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6010-6023, Aug. 2016.
- [6] Z. Ding, X. Lei, G. Karagiannidis, R. Schober, J. Yuan, V. Bhargava, "A survey on non-orthogonal multiple access for 5G networks: research challenges and future trends," *IEEE J. Sel. Areas Commun.*, pp. 2181-2195, vol. 35, no. 10, Oct. 2017.
- [7] L. Dai, B. Wang, Y. Yuan, S. Han, C. I, and Z. Wang, "Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends," *IEEE Communications Magazine*, pp. 74-81, vol. 53, no. 9, Sept. 2015.
- [8] Y. Saito, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, "Non-orthogonal multiple access (NOMA) for cellular future radio access," in *IEEE Veh. Tech. Conf. (VTC)*, Dresden, Germany, Jun. 2013.
- [9] B. Di, L. Song, and Y. Li, "Sub-channel assignment, power allocation and user scheduling for non-orthogonal multiple access networks," *IEEE Trans. Wireless Commun.*, pp. 7686-7698, vol. 15, no. 11, Nov. 2016.
- [10] H. Nikopour, and H. Baligh, "Sparse Code Multiple Access," *IEEE Int. Symp. Personal Indoor and Mobile Radio Commun.*, pp. 332-336, Sept. 2013, London, UK.
- [11] M. Beko, and R. Dinis, "Designing Good Multi-Dimensional Constellations," *IEEE Wireless Commun. Lett.*, vol. 1, no. 3, pp. 221-224, Jun. 2012.
- [12] M. Taherzadeh, H. Nikopour, A. Bayesteh, and H. Baligh, "SCMA codebook design," *IEEE Veh. Tech. Conf.*, pp. 1-5, Sep. 2014, Vancouver, BC.
- [13] M. Vaeri, Z. Ding, and H.V. Poor, *Multiple access techniques for 5G wireless networks and beyond*, Springer, 2019.
- [14] Y. Wu, S. Zhang, and Y. Chen, "Iterative multiuser receiver in sparse code multiple access systems," *IEEE Int. Commun. Conf.*, London, UK, May 2015.
- [15] L. Yu, X. Lei, P. Fan, and D. Chen, "An optimized design of SCMA codebook based on star-QAM signaling constellations," in *Int. Conf. Wireless Commun. & Signal Process. (WCSP)*, pp. 1-5, Oct. 2015, Nanjing, China.
- [16] C. Yan, G. Kang, and N. Zhang, "A dimension distance-based SCMA codebook design," *IEEE Access*, pp. 5471-5479, vol. 5, Mar. 2017.
- [17] J. Bao, Z. Ma, Z. Ding, G. Karagiannidis, and Z. Zhu, "On the design of multiuser codebooks for uplink SCMA systems," *IEEE Commun. Lett.*, pp. 1920-1923, vol. 20, no. 10, Oct. 2016.
- [18] Y. Du, B. Dong, Z. Chen, J. Fang, P. Gao, and Z. Liu, "Low-complexity detector in sparse code multiple access systems," *IEEE Commun. Lett.*, pp. 1812-1815, vol. 20, no. 9, Sept. 2016.
- [19] S. Lin, and D. Costello, *Error control coding*, Prentice Hall, 2004.
- [20] T. Schlomer, D. Heck, and O. Deussen, "Farther-point optimized point sets with maximum minimum distance", in *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*, pp. 135-142, Vancouver, Canada, Aug. 2011.
- [21] Z. Chen, Z. Yuan, Y. Choi, L. Liu, and W. Wang, "Variational blue noise sampling", *IEEE Trans. Visualization Computer Graphics*, pp. 1784-1796, vol. 18, no. 10, Oct. 2012.
- [22] L. Floriani, and E. Puppo, "An on-line algorithm for constrained Delaunay triangulation," *CVGIP: Graphical Models and Image Processing*, pp. 290-300, vol. 54, no. 4, July. 1992.
- [23] J. Erickson, "Dense point sets have sparse Delaunay triangulations," *Discrete & Computational Geometry*, pp. 83-115, vol. 33, no. 1, Jan. 2005.
- [24] S. Pietrobon, R. Deng, A. LaFanechere, G. Ungerboeck, and D. Costello, "Trellis coded multi-dimensional phase modulation," *IEEE Trans. Inform. Theory*, pp. 63-89, vol. 36, no. 1, Jan. 1990.
- [25] D. Divsalar, and M. Simon, "The design of trellis coded MPSK for fading channels: performance criteria," *IEEE Trans. Commun.*, pp. 1004-1012, vol. 36, no. 9, Sept. 1988.
- [26] S. Jamali, and T. Le-Ngoc, "A new 4-state 8PSK TCM scheme for fast fading, shadowed mobile radio channels," *IEEE Trans. Veh. Technol.*, pp. 216-222, vol. 40, no. 1, Feb. 1991.
- [27] J. Hagenauer, and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *IEEE Global Commun. Conf. (GlobeCom)*, Dallas, TX, Dec. 1989.
- [28] G. Forney, and L. Wei, "Multidimensional constellations-Part I: introduction, figures of merit, and generalized cross constellations," *IEEE Trans. Inf. Theory*, vol. 7, no. 6, pp. 877-892, Aug. 1989.