2023 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-7554-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISIT54713.2023.10206446

A Novel Soft-In Soft-Out Decoding Algorithm for VT Codes on Multiple Received DNA Strands

Zihui Yan, Guanjin Qu, and Huaming Wu
Center for Applied Mathematics, Tianjin University, China
Emails: {yanzh, guanjinqu, whming}@tju.edu.cn

Abstract—**In recent years, DNA-based data storage has received extensive attention as a promising technology due to its high density, long-term durability, and low power consumption. One of the most challenging issues in its field is coping with insertion, deletion, and substitution (IDS) errors introduced during DNA synthesis and sequencing. Many traditional codes have been introduced as error correction methods, among which VT codes have attracted widespread interest since it is asymptotically optimal. In this work, we provide a novel soft-input soft-output (SISO) decoding algorithm for VT codes and generalize it to decode multiple received sequences. Monte Carlo simulations show that the bit error rate of the SISO decoder is one order of magnitude lower as compared with that of the conventional hard-decision decoder. Additionally, the generalized decoding algorithm over multiple received sequences achieves significant performance gains compared to a single-sequence transmission case. We further provide two reduced-complexity strategies for our SISO decoding algorithm that greatly reduce the decoding complexity by truncating traces of small probability.**

I. Introduction

Synthetic deoxyribonucleic acid (DNA) molecules, which emerged as a long-term and high-density medium for data storage, have recently attracted a great deal of attention [1]–[7]. One of the significant challenges in DNA-based information coding schemes is the correction of the numerous errors generated during DNA synthesis and sequencing, particularly insertion, deletion, and substitution (IDS) errors [8]–[10]. In the context of this problem, one feature to be aware of is that the raw data will be separated into short strands and synthesized multiple times when stored [11], [12], and each synthesized strand will be independently sequenced multiple times via high-depth sequencing when accessed [13]. Obtaining consensus base calling through a multiple sequence alignment (MSA) of multiple copies and a majority decision on the alignment is a typical method for information recovery in this situation [14], [15]. However, it is considered to be the most space- and time-consuming step in the DNA data recovery process [16]. The time complexity of the algorithm proposed in [14] is $O(cn^2 + c^3n)$, where n and c denote the length and number of sequences, respectively.

Many attempts have been made to address the problem of reliable communication with IDS errors, such as convolutional codes [16]–[18], watermark codes [19], Time-varying block codes [20], and Varshamov-Tenengolts (VT) codes [21]. See [22] for a recent survey. Among these, we note that VT codes have proven to be a particularly applicable IDS error correction scheme for DNA storage due to its asymptotically optimal redundancy (i.e., it only requires $\lceil \log n \rceil + 1$ redundancy bits to correct a single IDS error).

The structure of VT codes was originally developed by Varshamov and Tenengol'ts for correcting a single asymmetric error [23]. Subsequently, Levenshtein proved that this structure is also effective in correcting a single IDS error and showed that it is asymptotically optimal [21]. Abdel-Gaffar and Ferreira first provided a systematic encoder for VT codes [24], which was later improved by Saowapa *et al.* [25]. Recently, VT codes have been applied to the DNA-based storage system. Cai *et al.* [26] provided a single IDS error correction code with order-optimal redundancy modified from VT codes. A segmented error correction code was designed in [27], where each segment is a VT codeword and can correct a single IDS error. These works focused on the encoding aspect and employed the conventional hard-decision decoding algorithm. However, the DNA synthesis and sequencing processes bring in stochastic IDS errors. In this context, multiple IDS errors are inevitably introduced into received sequences and the hard-decision decoder thus fails.

In this work, to generalize VT codes to the case of stochastic IDS errors, we for the first time focus on the decoding aspect. Our key improvement for VT codes is a soft-in soft-out (SISO) decoding algorithm. The main contributions are summarized as follows.

- We propose a SISO decoding algorithm for VT codes, i.e., a bit-wise maximum a posteriori (MAP) decoder based on the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm. This SISO decoding algorithm output the maximum posterior probability (APP) for each bit and can be applied to an arbitrary error rate.
- The SISO decoding algorithm can be simply improved to perform on multiple received sequences, which obtain higher performance gains from the sequencing redundancy, skipping the complex MSA process. In addition, in contrast to MSA technology that makes hard decisions on DNA bases based on a majority vote, our decoder outputs soft information. Thus, it is more applicable to concatenated coding schemes for DNA storage.
- We further provide two reduced-complexity strategies for our SISO decoding algorithm, i.e., the M-Reduced and T-Reduced decoding strategies. They calculate the traces

This work is supported by the National Key R&D Program of China (2020YFA0712100) and the National Natural Science Foundation of China (62071327). *(Corresponding author: Huaming Wu)*

only on the M states with the maximum value or on the states with values above a threshold, respectively, thus greatly reducing the computation over the trellis.

II. The Channel and VT codes

Notations. Row vectors are denoted by boldface letters (i.e., $\boldsymbol{u}$). The subvector collecting entries $u_i, u_{i+1}, \ldots, u_j$ for the vector $\boldsymbol{u} = \{u_1, u_2, \ldots, u_n\}$ is represented by $\boldsymbol{u}_i^j = \{u_i, u_{i+1}, \ldots, u_j\}$. If $i > j$, then $\boldsymbol{u}_i^j$ equals the string ϵ, which is empty. The conditional probability of A is denoted by the notation $Pr[A]$, while the likelihood of A given the occurrence of B is denoted by $Pr[A|B]$.

A. DNA Storage Channel Model

DNA is a molecule made up of four nucleotides: A (Adenine), C (Cytosine), G (Guanine), and T (Thymine). Nevertheless, to simplify the exposition, we focus on the binary case. Considering the context of DNA synthesis and sequencing, a preliminary model of the DNA data storage channel is to view the input as a set of binary sequences of length n [28], and the output as a set of binary sequences of irregular length, typically referred to as an IDS channel [16]. In the context of multiple sequencing reads, the transition of a single DNA strand can be thought of as parallel and independent IDS channels, with c erroneous sequences as the outputs.

We consider the IDS channel depicted in Fig. 1 [19], [29]. Let $\boldsymbol{v} = \{v_1, v_2, \ldots, v_n\} \in \{0,1\}^n$ denote the channel input of length n. When v_i is ready to be transmitted over the channel, three events may occur:

- With probability p_{i}, an insertion error occurs where a uniformly random symbol is appended before v_i. And v_i is still not transmitted.
- With probability p_{d}, symbol v_i is deleted.
- With probability $p_{\mathrm{t}} = (1 - p_{\mathrm{d}} - p_{\mathrm{i}})$, v_i is transmitted. Then with probability p_{s}, v_i is substituted by $v_i \oplus 1$.

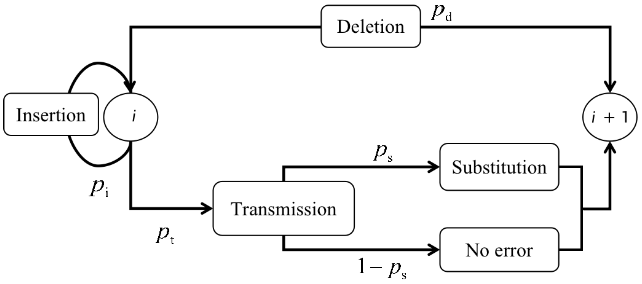

Fig. 1. A single use of the IDS channel.

It is assumed that insertion, deletion, and substitution errors are all independent and identically distributed (i.i.d.). For more clarity, the transition probabilities $Pr[\vec{r}_i|v_i]$ of a single use of the IDS channel for v_i are reported in Table I. Since the transmitter and the receiver have no information on the positions at which the errors occur, the overall output is the in-order concatenation of $\vec{r}_i$ without delimiters, such as $\boldsymbol{r} = (\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_n) = (r_1, r_2, \ldots, r_N) \in \{0,1\}^N$. Here, unobservable insertions and deletions cause the length of the received sequence N to fluctuate and perhaps differ from the length of the transmitted sequence n, where $(N - n)$ is equal to the number of insertion errors minus deletion errors.

Furthermore, we assume that $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_c$ denote the received sequences that correspond to c reads of the original strand $\boldsymbol{v}$. In Section III-C, we will show that our decoding algorithm can comprehensively utilize these multiple copies, thus effectively improving the decoding performance.

B. VT Code

In this work, we are interested in the original structure of VT codes, which refers to a class of binary algebraic block codewords that consists of all binary vectors of length n belonging to

$$VT_{a,m}(n) = \left\{ \boldsymbol{v} \in \{0,1\}^n : \sum_{i=1}^{n} i v_i \equiv a \pmod{m} \right\}, \quad (1)$$

where m is a predetermined integer, and a is an integer with $0 \le a \le m - 1$ which is usually called the syndrome or the remainder of the sequence $\boldsymbol{v}$. When $m \ge 2n$, this code can correct an IDS error [21].

We consider a systematic encoding scheme for VT codes [25]. We call it **Encoder VT** and briefly restate it as follows:

Encoder VT: For any message sequence $\boldsymbol{u} = \{u_1, u_2, \cdots, u_k\} \in \{0,1\}^k$, Encoder VT sticks these information bits into a codeword $\boldsymbol{v} = VT(\boldsymbol{u}) \in VT_{a,2n+1}(n)$, where $k = n - \lceil \log n \rceil - 1$. The encoder inserts “parity” bits at dyadic positions, i.e., v_{2^i}, for $0 \le i \le n - k - 2$ and v_n, and attaches message symbols to other positions, to ensure that $\sum_{i=1}^{n} i v_i \equiv a \pmod{2n+1}$.

Example: Given a message sequence $\boldsymbol{u} = 11011$ and a fixed syndrome $a = 0$, we have $n = 10$, $k = 5$ and $m = 21$. As the codeword $\boldsymbol{v} = (v_1, v_2, \cdots, v_{10})$ should satisfy that $\sum_{i=1}^{10} i v_i \equiv \sum_{j=1}^{4} 2^{j-1} v_{2^{j-1}} + 1{\cdot}3 + 1{\cdot}5 + 0{\cdot}6 + 1{\cdot}7 + 1{\cdot}9 + 10 \cdot v_{10} \equiv 0 \pmod{21}$, we have $\sum_{j=1}^{4} 2^{j-1} v_{2^{j-1}} + 10 \cdot v_{10} = 18$. It indicates that $v_{10} = 1$ and $v_8 = 1, v_1 = v_2 = v_4 = 0$ by expending $18 - 10 = 8$ into the binary form $1 \cdot 2^3$. The final codeword is $\bar{0}\bar{0}1\bar{0}101\bar{1}1\bar{1}$, where bits with overbars are check bits.

For completeness, we summarize the hard-decision decoder of Encoder VT as follows:

Hard-Decision Decoder: For any received sequence $\boldsymbol{r} = (r_1, r_2, \ldots, r_N)$, if $N = n - 1/N = n + 1/N = n$, it is assumed that there is a deletion/insertion/substitution error. Without loss of generality, we assume that the received sequence is $\boldsymbol{r} = (v_1, \ldots, v_{t-1}, r, v_t, \ldots, v_n)$. Denote the syndrome of the received sequence as $\overline{a} = \sum_{i=1}^{N} i \cdot r_i$, it follows that

$$\overline{a} - a \equiv \sum_{i=j}^{n} v_i + jr \pmod{2n+1}.$$

If $\overline{a} - a \le \sum_{i=1}^{N} r_i$, we can see that $r = 0$ and this insertion position is before the $(\overline{a} - a)$-th “1” of the sequence from back to front. Otherwise, this insertion is “1” and its position

TABLE I
TRANSITION PROBABILITIES FOR THE IDS CHANNEL.

Input	Output / $Pr[\cdot\vert\cdot]$					
v_i	$\vec{r}_i = \emptyset$	$\vec{r}_i = 0$	$\vec{r}_i = 1$	$\vec{r}_i = 00$	$\vec{r}_i = 01$	$\dots$
0	p_{d}	$p_{\mathrm{t}}(1-p_{\mathrm{s}}) + \frac{1}{2}p_{\mathrm{i}}p_{\mathrm{d}}$	$p_{\mathrm{t}}p_{\mathrm{s}} + \frac{1}{2}p_{\mathrm{i}}p_{\mathrm{d}}$	$\frac{1}{2}p_{\mathrm{i}}p_{\mathrm{t}}(1-p_{\mathrm{s}}) + (\frac{1}{2}p_{\mathrm{i}})^2p_{\mathrm{d}}$	$\frac{1}{2}p_{\mathrm{i}}p_{\mathrm{t}}p_{\mathrm{s}} + (\frac{1}{2}p_{\mathrm{i}})^2p_{\mathrm{d}}$	$\dots$
1	p_{d}	$p_{\mathrm{t}}p_{\mathrm{s}} + \frac{1}{2}p_{\mathrm{i}}p_{\mathrm{d}}$	$p_{\mathrm{t}}(1-p_{\mathrm{s}}) + \frac{1}{2}p_{\mathrm{i}}p_{\mathrm{d}}$	$\frac{1}{2}p_{\mathrm{i}}p_{\mathrm{t}}p_{\mathrm{s}} + (\frac{1}{2}p_{\mathrm{i}})^2p_{\mathrm{d}}$	$\frac{1}{2}p_{\mathrm{i}}p_{\mathrm{t}}(1-p_{\mathrm{s}}) + (\frac{1}{2}p_{\mathrm{i}})^2p_{\mathrm{d}}$	$\dots$

is after $(\overline{a}-a-\sum_{i=1}^{N} r_i+1)$-th "0" of the sequence. A similar result could be obtained for a deletion and substitution error.

The hard-decision decoder is especially suited to the scenario where one IDS error occurs per sequence. If there are any additional errors, the decoder will fail. However, when errors in the IDS channel occur randomly, inevitably multiple errors will occur in a sequence. It makes sense to alter the decoder to make it compatible with stochastic errors. In the next section, we will introduce a SISO decoding algorithm for Encoder VT and prove that it outperforms the above hard-decision decoding algorithm.

III. SISO DECODING ALGORITHM

A. Graph Representation of the IDS Channel

We note that the core of the hard-decision decoder of VT codes is its algebraic structure in Eq. (1). Armed with this algebraic structure, we develop another decoding strategy that is motivated by the optimal bit-by-bit decoding of trellis-based codes (i.e., the BCJR algorithm). We follow the optimality criterion of the bit-wise MAP, which makes us interested in computing the APP that a specific bit in the transmitted codeword $\boldsymbol{v} = (v_1, v_2, \dots, v_n)$ equals 0/1 when given the received word $\boldsymbol{r} = (r_1, r_2, \dots, r_N)$.

We define the state at time t as the combination of the syndrome value and the drift value of the sequence due to the deletion and insertion errors. To be specific, we use S_t to represent the syndrome state where $s_t \equiv \sum_{i=1}^{t} iv_i \pmod{2n+1}$, and D_t to represent the synchronization drift where d_t is defined as the number of insertion errors minus deletion errors until symbol v_t is received.

Using these symbols, and following the method of constructing factor graphs proposed in [30], we can decompose the MAP decoding criteria as:

$$\begin{aligned} &Pr[\boldsymbol{v}_1^n|\boldsymbol{r}_1^N, S_0 = s_0, D_0 = d_0]Pr[\boldsymbol{r}_1^N|S_0 = s_0, D_0 = d_0] \\ &= \prod_{t=1}^{n} \mathcal{F}[v_t, \boldsymbol{r}_{d_{t-1}+t}^{d_t+t}, s_t, d_t], \end{aligned} \tag{2}$$

where each factor is

$$\begin{aligned} &\mathcal{F}[v_t, \boldsymbol{r}_{d_{t-1}+t}^{d_t+t}, s_t, d_t] \\ &= Pr[v_t]Pr[\boldsymbol{r}_{d_{t-1}+t}^{d_t+t}|v_t, s_t, s_{t-1}, d_t, d_{t-1}]Pr[s_t, d_t|s_{t-1}, d_{t-1}]. \end{aligned} \tag{3}$$

For illustration, this factorization is represented in a factor graph (Fig. 2). For each factor in Eq. (3), there is a single trellis node (denoted by the letter "T"). For each variable on which the factor depends, there is a single edge connected to.

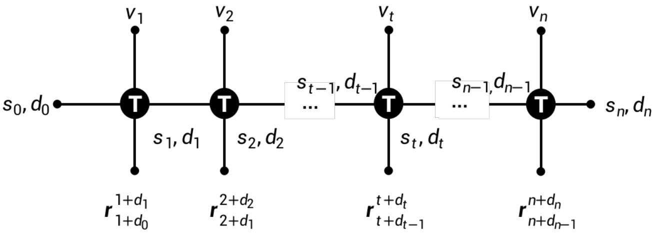

Fig. 2. Factor graph representation of the IDS channel.

Given the above graph representation, the trellis-based decoding algorithm can be used in the IDS channel model [31]. In this work, we employ an algorithm induced by the BCJR algorithm, which is optimal for the sense of minimizing the probability of bit errors [32]. Our algorithm is also analogous to the general forward-backward (FB) algorithm [33]. The main difference is that, contrary to the standard method discussed in [33], the IDS channel is not a finite state Markov chain.

B. SISO Decoder based on the BCJR algorithm

Without loss of generality, we now focus on the decoding of a bit v_t and the computing of the APP

$$Pr[v_t = 0/1|\boldsymbol{r}], \tag{4}$$

through the trellis shown in Fig. 2. We define the event that syndrome constrain $\sum_{i=1}^{n} iv_i \equiv a \pmod{2n+1}$ is satisfied as $\mathcal{S}$. Then the log-APP ratio, also known as the log-likelihood ratio (LLR), can be written as:

$$\begin{aligned} L(v_t) &= \log \frac{Pr[v_t = 0|\boldsymbol{r}, \mathcal{S}]}{Pr[v_t = 1|\boldsymbol{r}, \mathcal{S}]} \\ &= \log \frac{Pr[\boldsymbol{r}, \mathcal{S}|v_t = 0]}{Pr[\boldsymbol{r}, \mathcal{S}|v_t = 1]} + \log \frac{Pr[v_t = 0]}{Pr[v_t = 1]}. \end{aligned} \tag{5}$$

Considering the drift state and the syndrome state, we now generate the crucial results which facilitate the calculation of $Pr[\boldsymbol{r}, \mathcal{S}|v_t = b]$ as follows.

Theorem 1. *The probability density function (pdf) $Pr[\boldsymbol{r}, \mathcal{S}|v_t = b]$ is factored as*

$$\begin{aligned} &Pr[\boldsymbol{r}, \mathcal{S}|v_t = b] \\ &= \sum_{(d',d)\in\mathcal{D}_t, (s',s)\in\Sigma_t^b} Pr[s', d', s, d, \boldsymbol{r}] \qquad (6) \\ &= \sum_{(d',d)\in\mathcal{D}_t, (s',s)\in\Sigma_t^b} \alpha_{t-1}(s', d')\gamma_t(s', d', s, d)\beta_t(s, d), \qquad (7) \end{aligned}$$

where $\alpha_{t-1}(s', d') = Pr[\boldsymbol{r}_1^{d'+t-1}, s', d']$, $\beta_t(s,d) = Pr[\boldsymbol{r}_{d+t+1}^N|s,d]$, $\gamma_t(s',d',s,d) = Pr[\boldsymbol{r}_{d'+t}^{d+t}, s, d|s', d']$, *and* d_t *is the drift at time* t, $\mathcal{D}_t$ *is the set of pairs* (d', d) *for the drift accumulation* $(d_{t-1} = d') \to (d_t = d)$ *which satisfies* $d' \le d+1$, *and* Σ_t^b *is the set of pairs* (s', s) *for the syndrome states at time* t *which satisfies* $s' + tv_t \equiv s \pmod{2n+1}$.

Proof: Based on Bayes' rule and total probability theorem, after incorporating the code structure of VT codes into the factor graph of the IDS channel, Eq. (6) can be calculated by summing the message passed from the trellis, specifically by summing the metrics of all possible drift state transitions and syndrome state transitions. Then through several applications of Bayes' rule, the factor representation Eq. (7) of $Pr[\boldsymbol{r}, \mathcal{S}|v_t = b]$ can be obtained, where each component can be calculated as follows.

The probability $\alpha_t(s,d)$ and $\beta_t(s,d)$ can be computed from the forward message and the backward message, respectively, in particular, via a "forward recursion" and a "backward recursion" as follows:

$$\begin{aligned}\alpha_t(s,d) &= \sum_{(s',s)\in\Sigma_t,(d',d)\in\mathcal{D}_t} \gamma_t(s',d',s,d)\alpha_{t-1}(s',d');\\ \beta_t(s',d') &= \sum_{(s',s)\in\Sigma_{t+1},(d',d)\in\mathcal{D}_{t+1}} \gamma_{t+1}(s',d',s,d)\beta_{t+1}(s,d),\end{aligned} \tag{8}$$

where Σ_t denotes all possible syndrome states at time j, and the initial conditions are

$$\alpha_0(s,d) = \begin{cases} 1 & s=0, d=0;\\ 0 & \text{otherwise,}\end{cases} \tag{9}$$

and

$$\beta_n(s,d) = \begin{cases} 1 & s=a, d=N-n;\\ 0 & \text{otherwise.}\end{cases} \tag{10}$$

Then, $\gamma_t(s',d',s,d)$ is calculated as follows:

$$\begin{aligned}\gamma_t(s',d',s,d) &= Pr[\boldsymbol{r}_{d'+t}^{d+t}, s, d|s', d']\\ &= Pr[s|s']Pr[\boldsymbol{r}_{d'+t}^{d+t}, d|s', d', s],\end{aligned} \tag{11}$$

where

$$Pr[s|s'] = \begin{cases} 1 & s' + jb \equiv s \pmod{2n+1};\\ 0 & \text{otherwise,}\end{cases} \tag{12}$$

and for $l \ge 0$,

$$\begin{aligned}&Pr[\boldsymbol{r}_{d'+t}^{d+t}, d|s', d', s]\\ &= \begin{cases} p_{\rm d} & d = d'-1;\\ \left(\frac{1}{2}p_{\rm i}\right)^l(p_{\rm t}p_{\rm s} + \frac{1}{2}p_{\rm i}p_{\rm d}) & \boldsymbol{r}_{d+t} \ne b, d = d'+l;\\ \left(\frac{1}{2}p_{\rm i}\right)^l(p_{\rm t}(1-p_{\rm s}) + \frac{1}{2}p_{\rm i}p_{\rm d}) & \boldsymbol{r}_{d+t} = b, d = d'+l;\\ 0 & \text{otherwise.}\end{cases}\end{aligned} \tag{13}$$

■

In light of the foregoing discussion, we may compute $L(v_t)$ via Theorem 1. All that remains at this point is to discuss the a priori probability of v_t,

$$\log\frac{Pr[v_t = 0]}{Pr[v_t = 1]}. \tag{14}$$

Since typically $Pr[v_t = 0] = Pr[v_t = 1]$, the a priori term is usually zero. And then we may use the L-value to obtain hard decisions: $\hat{v}_t = \text{sign}[L(v_t)]$. Afterward, the information bits are recovered from the inverse operation of the encoder.

An alternative interpretation of the decoder can be appreciated by considering the SISO decoder, which can accept extrinsic information from a component decoder (serves as a priori information Eq. (14)) and further, produce soft outputs. Thus, it can perform as the iterative (turbo) decoder.

C. Decoding for Multiple Received Sequences

We now show how to calculate the APP of a bit v_t when given multiple received sequences. Suppose that the binary sequence $\boldsymbol{v}$ is transmitted independently over c parallel IDS channels, and the outputs are $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_c$.

Let $\sigma_t^b = \{\boldsymbol{v} : v_t = b\}$. In line with Eq. (5), we have

$$\begin{aligned}L(v_t) &= \log\frac{Pr[v_t = 0|\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_c]}{Pr[v_t = 1|\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_c]}\\ &= \log\frac{\sum_{\boldsymbol{v}\in\sigma_t^0} Pr[\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_c|\boldsymbol{v}]Pr[\boldsymbol{v}]}{\sum_{\boldsymbol{v}\in\sigma_t^1} Pr[\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_c|\boldsymbol{v}]Pr[\boldsymbol{v}]}\\ &\overset{(a)}{\simeq} \log\frac{\prod_{i=1}^c \sum_{\boldsymbol{v}\in\sigma_t^0} Pr[\boldsymbol{r}_i|\boldsymbol{v}]}{\prod_{i=1}^c \sum_{\boldsymbol{v}\in\sigma_t^1} Pr[\boldsymbol{r}_i|\boldsymbol{v}]} + \log\frac{\sum_{\boldsymbol{v}\in\sigma_t^0} Pr[\boldsymbol{v}]}{\sum_{\boldsymbol{v}\in\sigma_t^1} Pr[\boldsymbol{v}]}\\ &= \sum_{i=1}^c \log\frac{Pr[\boldsymbol{r}_i|v_t = 0]}{Pr[\boldsymbol{r}_i|v_t = 1]} + \log\frac{Pr[v_t = 0]}{Pr[v_t = 1]},\end{aligned} \tag{15}$$

where (a) follows the fact that $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_c$ are conditional independent when given $\boldsymbol{v}$ and $Pr[\boldsymbol{r}|\boldsymbol{v}]$ can be assumed to be sufficiently small for mismatched $\boldsymbol{v}$ and $\boldsymbol{r}$. Thus, the total LLR of a bit v_t can be calculated as the sum of L-values, each independently obtained from a received sequence. With Eq. (15), the decoder can accept all channel information, thus allowing significant gains from multiple sequence transmission cases.

D. Complexity Analysis

In this subsection, we evaluate the complexity of our SISO decoding algorithm. The complexity is mainly proportional to the number of state sets $\mathcal{D}$ and Σ, which determines the number of performed operations at each time unit. For $1 \le t \le n$, we have $|\Sigma_t| = 2(2n+1)$ since the current syndrome state s_t may in $[0, 2n]$ and the next syndrome state $s_{t+1} \equiv s_t$ or $s_t + t \pmod{2n+1}$. In line with [16], [19], we limit the drift transitions to a fixed interval, where $d_t \in [d_{\min}, d_{\max}]$, and the maximum insertion errors per bit to $I_{\max}$. We set $\Delta = d_{\max} - d_{\min} + 1$ to denote the total number of drift states. Thus, for $1 \le t \le n$, we have $|\mathcal{D}_t| = \Delta(I_{\max} + 1)$. As a result, the complexity to decode a single received sequence is $O(\Delta(I_{\max} + 1)n^2)$. In the case of c multiple copies, the complexity is simply c times that of the single sequence decoding, i.e., $O(c\Delta(I_{\max} + 1)n^2)$.

IV. Two Reduced-Complexity Strategies for the SISO Decoder

We now propose two lower-complexity strategies of our SISO decoding algorithm, namely, the M-reduced decoding strategy and the T-reduced strategy, which are inspired by the popular strategies for reducing memory in BCJR algorithms named M-BCJR and T-BCJR, respectively [34], [35]. We note that in the trellis-based recursive computation, the probabilities on most traces are very small. The main work of our proposed strategies is to truncate small probabilities and their correlations computations.

A. M-Reduced Decoding Strategy

In this strategy, the decoder traces only the M optimal forward states and does not evaluate the other states at each trellis stage. To be specific, the forward recursion computation of α_t by Eq. (8) will be performed only on M $(t-1)$-th states with the largest probabilities, while the other states are set to 0 and declared dead. The backward recursion is calculated only on the alive states after the forward recursion. At each time unit, the probabilities need to be normalized. Fig. 3 represents the state transmission diagram of the SISO decoding algorithm with a 3-dimensional trellis and characterizes the M-reduced trellis when $M = 2$.

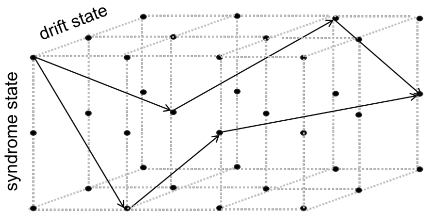

Fig. 3. M-reduced trellis with $M = 2$, maintaining the best two states at each time unit. The arrow indicates that the forward state survives and is used to compute the subsequent state and that the subsequent state survives the computation as well.

B. T-Reduced Decoding Strategy

In this strategy, the states are declared dead when their forward probabilities are less than a threshold T. That is when calculating α_t, only the alive state components of α_{t-1} that exceed the threshold T are used. Thus, the number of alive states at each time unit is varied (Fig. 4). The remaining steps are the same as in the M-reduced case.

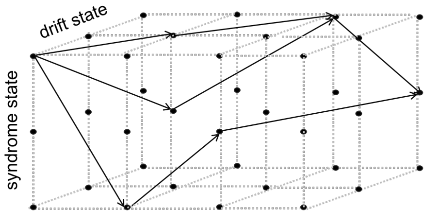

Fig. 4. T-reduced trellis.

For both strategies, the decoding complexity is greatly reduced due to the use of a dynamically constructed trellis, which avoids the full computation of the original trellis. In practice, almost all the computations are consumed in the updating of the forward and backward probabilities of the states. Now it needs to be performed only on the alive states so that the effort will be saved on the dead states.

V. Performance Evaluation

In this section, we present the bit-error-rate (BER) performance of the SISO decoder. To have a more intuitive evaluation of the decoding performance, we directly used the L-values of the SISO decoder to obtain the hard decisions in the simulations. We set the independent variable as the total IDS error rate $pr = p_{\mathrm{i}} + p_{\mathrm{d}} + p_{\mathrm{s}}$ with $p_{\mathrm{i}} = p_{\mathrm{d}} = p_{\mathrm{s}}$, and the dependent variable as the BER of the message of length 60 bits (where the corresponding code word length is 68 bits).

We first compared our algorithm with Hard-Decision Decoder for the case of a single-sequence transmission (Fig. 5). These curves show that our SISO decoder provides a significant decoding performance gain compared to the hard-decision method. Besides, the dashed lines in Fig. 5 show the performance of the M-reduced and T-reduced decoding strategies. By reducing the number of survival states, the decoding complexity is decreased. When the threshold is $T = (pr/3)^4$, the decoding process truncates the traces that indicate more than 4 errors in the received sequence and reduces the computation to one-third of the states per trellis stage.

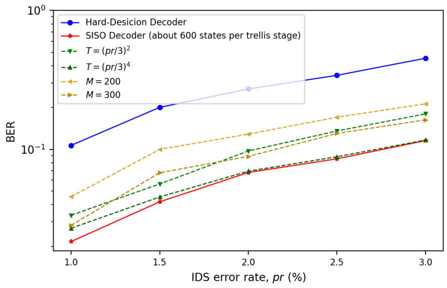

Fig. 5. BER performance over a single-sequence transmission.

We further investigated the coding performance on multiple sequences, as shown in Fig. 6. We observed a significant gain in performance when increasing the number of sequences.

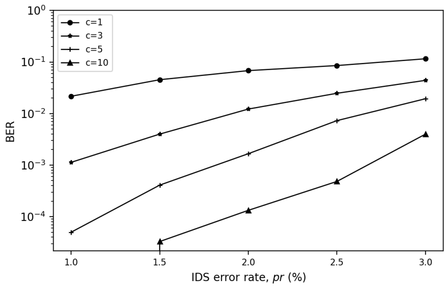

Fig. 6. BER performance over multiple received sequences.

VI. Conclusion

In this paper, we propose an innovative SISO decoding algorithm for VT codes and extend it to the case of multi-sequence decoding. We also propose two strategies to reduce the computational complexity for the algorithm. Simulations show that our decoding algorithm outperforms the traditional hard-decision approach. In summary, our work highlights the potential applications of VT codes in DNA storage and provides new approaches for the design of practical IDS error correction codes.

References

[1] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney, "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, vol. 494, no. 7435, pp. 77–80, jan 2013.

[2] Y. Erlich and D. Zielinski, "DNA fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. págs. 950–954, 2017.

[3] M. A. Sini and E. Yaakobi, "Reconstruction of sequences in DNA storage," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 290–294.

[4] L. Song, F. Geng, Z.-Y. Gong, X. Chen, J. Tang, C. Gong, L. Zhou, R. Xia, M.-Z. Han, J.-Y. Xu, B.-Z. Li, and Y.-J. Yuan, "Robust data storage in DNA by de Bruijn graph-based de novo strand assembly," *Nature Communications*, vol. 13, no. 1, p. 5361, 2022.

[5] A. Rasool, Q. Qu, Y. Wang, and Q. Jiang, "Bio-constrained codes with neural network for density-based DNA data storage," *Mathematics*, vol. 10, no. 5, 2022.

[6] G. Qu, Z. Yan, and H. Wu, "Clover: tree structure-based efficient DNA clustering for DNA-based data storage," *Briefings in Bioinformatics*, vol. 23, no. 5, p. bbac336, 2022.

[7] Z. Yan, C. Liang, and H. Wu, "Upper and lower bounds on the capacity of the DNA-based storage channel," *IEEE Communications Letters*, vol. 26, no. 11, pp. 2586–2590, 2022.

[8] S. M. H. T. Yazdi, H. M. Kiah, E. Garcia-Ruiz, J. Ma, H. Zhao, and O. Milenkovic, "DNA-based storage: Trends and methods," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 1, no. 3, pp. 230–248, 2015.

[9] R. Heckel, G. Mikutis, and R. N. Grass, "A characterization of the DNA data storage channel," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[10] X. Li, M. Chen, and H. Wu, "Multiple errors correction for position-limited DNA sequences with GC balance and no homopolymer for DNA-based data storage," *Briefings in Bioinformatics*, vol. 24, no. 1, p. bbac484, 2023.

[11] B. H. Nguyen, C. N. Takahashi, G. Gupta, J. A. Smith, R. Rouse, P. Berndt, S. Yekhanin, D. P. Ward, S. D. Ang, P. Garvan *et al.*, "Scaling DNA data storage with nanoscale electrode wells," *Science advances*, vol. 7, no. 48, p. eabi6714, 2021.

[12] Z. Yan, C. Liang, and H. Wu, "A segmented-edit error-correcting code with re-synchronization function for DNA-based storage systems," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–13, 2022.

[13] C. Winston, L. Organick, D. Ward, L. Ceze, K. Strauss, and Y.-J. Chen, "A combinatorial PCR method for efficient, selective oligo retrieval from complex oligo pools," *ACS Synthetic Biology*, vol. 11, no. 5, pp. 1727–1734, 2022.

[14] R. C. Edgar, "MUSCLE: multiple sequence alignment with high accuracy and high throughput," *Nucleic acids research*, vol. 32, no. 5, pp. 1792–1797, 2004.

[15] K. Reinert, T. H. Dadi, M. Ehrhardt, H. Hauswedell, S. Mehringer, R. Rahn, J. Kim, C. Pockrandt, J. Winkler, E. Siragusa, G. Urgese, and D. Weese, "The SeqAn C++ template library for efficient sequence analysis: A resource for programmers," *Journal of Biotechnology*, vol. 261, pp. 157–168, 2017.

[16] I. Maarouf, A. Lenz, L. Welter, A. Wachter-Zeh, E. Rosnes, and A. G. i. Amat, "Concatenated codes for multiple reads of a DNA sequence," *IEEE Transactions on Information Theory*, pp. 1–1, 2022.

[17] V. Buttigieg and N. Farrugia, "Improved bit error rate performance of convolutional codes with synchronization errors," *2015 IEEE International Conference on Communications (ICC)*, pp. 4077–4082, 2015.

[18] W. Press, J. Hawkins, S. Jones, J. Schaub, and I. Finkelstein, "HEDGES error-correcting code for DNA storage corrects indels and allows sequence constraints," *Proceedings of the National Academy of Sciences*, vol. 117, no. 31, pp. 18 489–18 496, 2020.

[19] M. Davey and D. Mackay, "Reliable communication over channels with insertions, deletions, and substitutions," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 687–698, 2001.

[20] V. Buttigieg, S. Wesemeyer, and J. Briffa, "Time-varying block codes for synchronisation errors: maximum a posteriori decoder and practical issues," *The Journal of Engineering*, vol. 2014, 06 2014.

[21] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet physics. Doklady*, vol. 10, pp. 707–710, 1965.

[22] R. R. Garafutdinov, D. A. Chemeris, A. R. Sakhabutdinova, O. Y. Kiryanova, C. I. Mikhaylenko, and A. V. Chemeris, "Encoding of non-biological information for its long-term storage in DNA," *Biosystems*, vol. 215-216, p. 104664, 2022.

[23] R. Varšamov and G. Tenengolts, "A code which corrects single asymmetric errors," *Avtomat. i Telemeh*, vol. 26, no. 288-292, p. 4, 1965.

[24] K. Abdel-Ghaffar and H. Ferreira, "Systematic encoding of the Varshamov-Tenengol'ts codes and the Constantin-Rao codes," *IEEE Transactions on Information Theory*, vol. 44, pp. 340 – 345, 02 1998.

[25] K. Saowapa, H. Kaneko, and E. Fujiwara, "Systematic deletion/insertion error correcting codes with random error correction capability," in *Proceedings 1999 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (EFT'99)*, 1999, pp. 284–292.

[26] K. Cai, Y. M. Chee, R. Gabrys, H. M. Kiah, and T. T. Nguyen, "Correcting a single indel/edit for DNA-based data storage: Linear-time encoders and order-optimality," *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 3438–3451, 2021.

[27] K. Cai, H. M. Kiah, M. Motani, and T. T. Nguyen, "Coding for segmented edits with local weight constraints," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 1694–1699.

[28] N. Weinberger and N. Merhav, "The DNA storage channel: Capacity and error probability bounds," *IEEE Transactions on Information Theory*, vol. 68, no. 9, pp. 5657–5700, 2022.

[29] R. Hulett, S. Chandak, and M. Wootters, "On coding for an abstracted nanopore channel for DNA storage," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 2465–2470.

[30] A. Kavcic, X. Ma, and M. Mitzenmacher, "Binary intersymbol interference channels: Gallager codes, density evolution, and code performance bounds," *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1636–1652, 2003.

[31] S. R. Srinivasavaradhan, S. Gopi, H. D. Pfister, and S. Yekhanin, "Trellis BMA: Coded trace reconstruction on ids channels for DNA storage," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 2453–2458.

[32] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[33] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[34] G. Colavolpe, G. Ferrari, and R. Raheli, "Reduced-state BCJR-type algorithms," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 5, pp. 848–859, 2001.

[35] V. Franz and J. Anderson, "Concatenated decoding with a reduced-search BCJR algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 186–195, 1998.