

Received 20 October 2022; revised 6 May 2023; accepted 15 May 2023; date of publication 24 May 2023; date of current version 23 June 2023.

Digital Object Identifier 10.1109/TQE.2023.3279379

# Branch-Assisted Sign-Flipping Belief Propagation Decoding for Topological Quantum Codes Based on Hypergraph Product Structure

TZU-HSUAN HUANG<sup>1</sup>, TING-AN HU<sup>1</sup>,  
AND YEONG-LUH UENG<sup>2</sup>  (Senior Member, IEEE)

<sup>1</sup>Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan

<sup>2</sup>Department of Electrical Engineering and the Institute of Communications Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan

Corresponding author: Yeong-Luh Ueng (e-mail: ylueng@ee.nthu.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 110-2221-E-007-029-MY3.

**ABSTRACT** Topological codes, a kind of quantum error correction code, have been used for current quantum computers due to their local qubit layout and high threshold. With nearly linear complexity, syndrome-based belief propagation (BP) can be considered as a decoding candidate for topological codes. However, such highly degenerate codes will lead to multiple low-weight errors where the syndrome is identical so that the BP decoding is not able to distinguish it, resulting in degradation in performance. In this article, we propose a branch-assisted sign-flipping belief propagation (BSFBP) decoding method for topological codes based on the hypergraph product structure. In our algorithm, we introduce the criteria to enter the new decoding path branched from BP combined with a syndrome residual, which is obtained from the syndrome-pruning process. A sign-flipping process is also conducted to disturb the log-likelihood ratio of the selected variable nodes, which provides diversity in the syndrome residual. Simulation results show that using the proposed BSFBP decoding is able to outperform the BP decoding by about two orders of magnitude.

**INDEX TERMS** Belief propagation (BP) decoding, hypergraph product (HGP) codes, topological codes.

## I. INTRODUCTION

As transistors reduce in size, the quantum effect will be increasingly important. For example, along with the short length of the gate, the quantum tunneling effect will decrease performance. Thus, quantum computers become another potential bottleneck route for Moore's law. In a quantum computer, the information will be disturbed by noise. Thus, quantum error correction (QEC) codes are necessary for the information-passing process.

The original famous QEC code was the 9-qubit Shor code, which is able to correct both a single bit-flip error and a single phase-flip error [1]. Later, a widely used type of QEC code, called the Calderbank–Shor–Steane (CSS) code, was constructed using two related classical codes under a variety of constraints [2], [3]. Since then, many QEC codes have been constructed using existing classical codes, such as quantum low-density parity-check (LDPC) codes [4] and quantum expander codes [5].

Another type of code is the topological code, such as the toric code [6] and surface codes [7], [8], which are defined on a spin-lattice. Because of the locality of the stabilizer measurements for topological codes, they have become beneficial to fault-tolerant large-scale quantum implementation [9] and have been used in ongoing research [10], [11]. In fact, almost all the QEC codes, from the earliest Shor code to current topological codes, are classified as CSS codes. A subtype of CSS construction, called hypergraph product (HGP) construction, can also be used to construct topological codes [12]. Furthermore, HGP construction preserves the merits of topological codes and has a better minimum-distance growth compared to other constructions, such as hyperbolic construction [13]. Thus, our article will mainly focus on such topological codes.

Since QEC codes can be regarded as a classical binary code [14] or as a classical 4-ary additive code with some constraints [15], many decoding algorithms for classical codes

can be applied. One of the main decoding algorithms that have nearly linear complexity is belief propagation (BP) [16]. BP decoding, modified with the syndrome as its input due to the quantum state collapse, has been used in QLDPC codes, resulting in remarkable performance [17], [18], [19], [20].

In addition, binary BP decoding can be used as an auxiliary to yield the information of qubits for other decoding algorithms. In this way, an output of binary BP decoding can be a reliable input to the main decoders. For example, Grospeilier et al. [21] use the soft output from the binary BP decoding for the small-set-flip (SSF) decoding, where the regular quantum expander codes are considered [5]. Then, the SSF will search all the supports of the generators in the stabilizer group and add to the estimated error by the one that is able to reduce the weight of the corresponding syndrome [22]. Recently, inspired by ordered-statistics decoding (OSD) for classical linear block codes [23], Panteleev and Kalachev [24] proposed a quantum-version OSD using QLDPC codes, including OSD with order 0 (OSD-0) as well as a higher order version. A modified higher order OSD is proposed via a combination sweep method (OSD-CS) [25], and the authors also show that the quantum-version OSD can be used for topological codes. However, BP decoding concatenated with OSD will result in additional complexity and latency when compared to pure BP decoding, which is a disadvantage for practical applications [26].

Decoders intended for topological codes are typically based on the minimum weight perfect matching (MWPM) method [6], [27], which has the complexity  $O(N^3)$  at the worst case. Recently, some decoders based on MWPM have been proposed especially for topological codes, which can reduce the complexity approximately to  $O(N \log N)$  [28], [29]. Although it has been shown that MWPM provides outstanding performance for topological codes, the requirement of pair-like syndromes limits its universality [30]. For example, to date, the MWPM decoder cannot be applied directly to some higher dimensional topological codes that include parity-check matrices that have column weights larger than 2 [31]. Instead, BP decoding, which has a complexity that is approximately linear to the code length, is a more general decoder for different types of codes; thus, it is considered here.

However, due to the highly degenerate nature of topological codes, BP will suffer from performance degradation. Degeneracy occurs when the codes contain many low-weight stabilizers compared to their minimum distance [32], [33], [34]. For the highly degenerate topological code, we find that its parity-check matrix will include lower row and column weights, leading to multiple low-weight errors, which have syndromes identical to the measured syndrome and will cause the BP decoding to fail. Since the difference in performance between using nonbinary BP and binary BP decoding is not significant for topological codes [32], we will focus on improving the performance of binary BP decoding under the consideration of complexity.

In this article, we propose a decoding algorithm termed branch-assisted sign-flipping belief propagation (BSFBP) decoding. In our algorithm, criteria are first proposed to identify the specific estimated syndrome. Then, the syndrome residual is obtained from the proposed syndrome-pruning process and is viewed as the syndrome input for the new decoding path branched from the syndrome-based BP. This branch-assisted process is able to mitigate the effect of those indistinguishable error patterns that would cause the syndrome-based BP to fail. We also propose a sign-flipping (SF) process to disturb the log-likelihood ratio (LLR) of the selected variable node. In this way, the process is able to provide a new decoding path that has diverse estimated syndromes, thereby improving the performance. BSFBP is able to outperform BP decoding by about two orders of magnitude. Our proposed decoding method can also be an auxiliary for OSD, providing a significant performance gain, which can compete with the MWPM for the considered 2-D topological codes. We also show that our BSFBP can be applied to those codes where the MWPM is limited, such as 3-D topological codes based on the homological product [35] and other types of codes based on HGP structures.

The rest of this article is organized as follows. In Section II, the background to QEC codes and binary BP decoding is introduced. Section III provides a detailed description of our proposed algorithm for topological codes. Our simulation results are shown in Section IV. Finally, Section V concludes this article.

## II. PRELIMINARIES

Topological quantum codes, such as toric code [6] and surface code [7], [8], are constructed based on the lattice. The qubits for the topological codes are located on the edge, while stabilizer operators are related to the vertices and the plaquettes of the lattice. Thanks to their tolerance to local errors, they have been used for fault-tolerant quantum computing [9]. An HGP structure can also be used to construct a class of topological codes that have growing minimum distance [12], which will be described as follows.

### A. QEC CODES IN HGP STRUCTURES

Pauli matrices are useful for describing the operations acting on a single qubit in quantum computation [36]. They are defined as follows:

$$\begin{aligned} I &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & X &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ Y &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} & Z &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \end{aligned} \quad (1)$$

where  $X$ ,  $Y$ , and  $Z$  anticommute with each other. A single qubit state  $|\phi\rangle$  can be represented by a linear superposition of two orthonormal basis vectors  $|0\rangle = (1 \ 0)^T$ ,  $|1\rangle = (0 \ 1)^T$ , where  $T$  is the transpose operator. That is,  $|\phi\rangle =$

$\alpha_1 |0\rangle + \alpha_2 |1\rangle \in \mathcal{H}_2$ , where  $\alpha_1$  and  $\alpha_2$  are in the set of complex numbers  $\mathbb{C}$ ,  $|\alpha_1|^2 + |\alpha_2|^2 = 1$ , and  $\mathcal{H}_2$  denotes the 2-D complex vector space.  $X$  can also be viewed as a bit-flip since  $X|\phi\rangle = \alpha_1|1\rangle + \alpha_2|0\rangle$ . Similarly,  $Z$  is a phase-flip since  $Z|\phi\rangle = \alpha_1|0\rangle - \alpha_2|1\rangle$ , and  $Y = iXZ$  represents both. The Pauli matrices together with the multiplicative factors  $\pm 1$  and  $\pm i$  form a Pauli group  $\mathcal{G}_1$  under the matrix multiplication. Thus, an  $n$ -fold tensor product of Pauli group  $\mathcal{G}_n$  is denoted as

$$\begin{aligned} \mathcal{G}_n &\equiv \mathcal{G}_1^{\otimes n} \\ &\equiv \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}^{\otimes n} \quad (2) \end{aligned}$$

for  $n$  qubits.

The stabilizer formalism is useful for describing QEC codes [37]. Given  $\mathcal{G}_n$ , an  $[[n, k, d]]$  quantum code is able to encode  $k$  qubits to an  $n$ -qubit codeword with a minimum weight of Pauli operations  $d$ , where the weight is the number of qubits operated by a nonidentity Pauli matrix. The double bracket is used for quantum codes to distinguish from the conventional  $[n, k, d]$  notation for classical codes. An  $[[n, k, d]]$  stabilizer code  $C_S$  is defined by its stabilizer group  $\mathcal{S} \subseteq \mathcal{G}_n$ , where  $-I \notin \mathcal{S}$  and all the elements commute mutually. A codeword  $|\psi\rangle \in C_S$  is the +1 eigenstate of all the stabilizers, i.e.,  $\mathcal{P}|\psi\rangle = |\psi\rangle$ ,  $\mathcal{P} \in \mathcal{S}$ . In addition,  $\mathcal{S}$  can be characterized by  $n - k$  generators  $g_i \in \mathcal{G}_n$ ,  $0 \leq i < n - k$ . That is, every element  $\mathcal{P} \in \mathcal{S}$  can be represented as a product of  $g_0, \dots, g_{n-k-1}$ . Hence, by the isomorphism between Pauli group and binary vector, i.e.,  $I \mapsto (0\ 0)$ ,  $X \mapsto (1\ 0)$ ,  $Z \mapsto (0\ 1)$ ,  $Y \mapsto (1\ 1)$ , these  $n - k$  generators  $g_i$  together can be represented in terms of parity-check matrix  $H_S$  for  $C_S$ , where each row  $i$  of  $H_S$  is the binary representation of  $g_i$  [38]. For example, a generator  $XZZXI$  with  $n = 5$  can be represented as a  $2n$ -binary vector  $(1\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0)$ , where the 1s in the leftmost and rightmost  $n$  bits correspond to the qubits, in which  $X$  operate and  $Z$  operate, respectively [14].

An  $[[n, k_X - k_Z, d]]$  CSS code [2], [3] is a kind of stabilizer code, which is constructed using two classical codes  $C_X[n, k_X, d_X]$  and  $C_Z[n, k_Z, d_Z]$ , where  $d \geq \min\{d_X, d_Z\}$ ,  $C_Z \subset C_X$ , and both  $C_X$  and  $C_Z^\perp$ , the dual of  $C_Z$ , are able to correct up to  $t$ -bit errors, i.e., both  $d_X, d_Z \geq 2t + 1$ . Given the parity-check matrix  $H_X$  of  $C_X$  and  $H_Z$  of  $C_Z^\perp$ , the CSS code constructed from  $C_X$  and  $C_Z$  has the form

$$H = \begin{pmatrix} H_Z & 0 \\ 0 & H_X \end{pmatrix} \quad (3)$$

where  $H_X \cdot H_Z^T = 0$ .

HGP code [12] is also a class of CSS codes. Given two classical binary codes  $C_1[n_1, k_1, d_1]$  and  $C_2[n_2, k_2, d_2]$  that have parity-check matrices  $H_1 \in F_2^{m_1 \times n_1}$  and  $H_2 \in F_2^{m_2 \times n_2}$ , respectively, the HGP code has parameters  $[[n_1 n_2 + m_1 m_2, k_1 k_2 + k_2' k_1', d]]$ , where  $d = \min\{d_1, d_1', d_2, d_2'\}$  and  $k_i'(d_i')$  are the dimensions (minimum distances) of the transpose code of  $C_i$ , i.e., the code specified by the transpose of  $H_i$ , for  $i = 1, 2$ .  $H_X$  and  $H_Z$  of the CSS form in the HGP

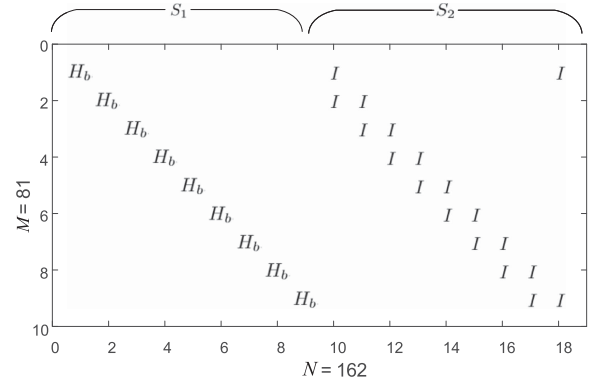


FIGURE 1. Parity-check matrix  $H_Z$  for the  $[[162,2,9]]$  toric code.

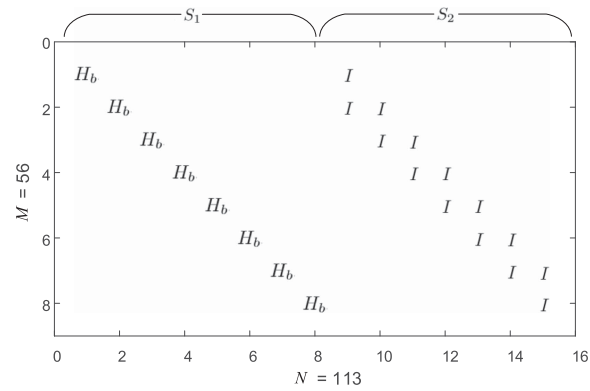


FIGURE 2. Parity-check matrix  $H_Z$  for the  $[[113,1,8]]$  surface code.

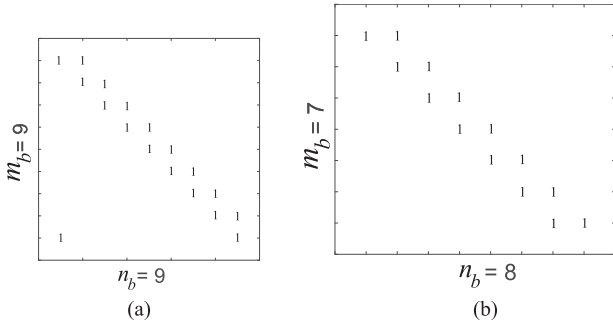
construction are

$$H_X = (H_1 \otimes I_{n_2} \quad I_{m_1} \otimes H_2^T) \quad (4)$$

$$H_Z = (I_{n_1} \otimes H_2 \quad H_1^T \otimes I_{m_2}) \quad (5)$$

where  $I_k$  denotes the  $k \times k$  identity matrix. It can be easily verified that HGP is a CSS code since  $H_X \cdot H_Z^T = 0$ . In this article, we consider the symmetric HGP for which  $H_1 = H_2 = H_b \in F_2^{m_b \times n_b}$ . The HGP structure can also be used to construct topological codes that have a minimum distance proportional to  $n^{1/2}$ , where  $n$  is the length of the quantum code. For example, the  $[[2n_b^2, 2, n_b]]$  toric code and the  $[[n_b^2 + (n_b - 1)^2, 1, n_b]]$  surface code considered in this article are obtained based on the HGP structure using the  $n_b \times n_b$  parity-check matrix  $H_b$  of an  $[n_b, 1, n_b]$  closed-loop repetition code and the full-rank  $(n_b - 1) \times n_b$  parity-check matrix  $H_b$  of a classical  $[n_b, 1, n_b]$  repetition code, respectively [25]. Figs. 1 and 2 show the structure of  $H_Z$  of the  $[[162,2,9]]$  toric code as well as for the  $[[113,1,8]]$  surface code with the parity-check matrices  $H_b$  shown in Fig. 3(a) and (b), respectively. Blanks in these figures represent zeros.

As mentioned in Section I, topological codes are highly degenerate, which means that the weights of their stabilizers, i.e., the row weights of their parity-check matrices, are also low. For example, Fig. 1 shows that the average row



**FIGURE 3.** Parity-check matrix  $H_b$  used in different topological codes. (a) Toric code ( $n_b = 9$ ). (b) Surface code ( $n_b = 8$ ).

and column weights are 4 and 2, respectively, for  $H_Z$  for the  $[[162,2,9]]$  toric code, while for  $H_Z$  of the  $[[113,1,8]]$  surface code, the average row and column weights are lower than 4 and 2, respectively. This fact also results in a low-average column weight and will have an adverse impact on the syndrome-based BP decoding algorithm.

### B. BINARY BP DECODING

BP is an efficient iterative decoding algorithm for classical LDPC codes [39], [40] and can also be applied to QEC codes [14]. For classical codes, the information of transmitted bits from the noisy channel is measured and treated as the intrinsic information for the BP decoding so as to estimate the transmitted bits. Unlike decoding on classical codes, transmitted information cannot be measured directly since the measurement will collapse a qubit to the specific classical state (0 or 1). Thus, there is no intrinsic information from the channel for quantum codes. Instead, it is only the syndrome that can be obtained from the measurement on the auxiliary qubits [36]. Thus, the syndrome-based BP decoding is used for QEC codes [41]. In this article, we will focus on the syndrome-based binary BP decoding, which we refer to as BP for simplicity.

In this article, we consider the discrete error model, where errors disturb each qubit independently and their effects can be described by the Pauli matrices, i.e., for a single qubit, the error operating on it can be described by  $X$  (bit flip),  $Z$  (phase flip), and  $Y$  (both). Furthermore, since  $Y$  can be decomposed into  $X$  and  $Z$ , under the consideration that  $X$  and  $Z$  are uncorrelated, the noise model can be further viewed as two independent channels: bit-flip and phase-flip channels [36].

The noisy model containing independent bit-flip and phase-flip channels can be analogized as two independent binary symmetric channels (BSCs) with the crossover probability  $p$  [14]. For example, based on the isomorphism between the Pauli operators and the binary vector, an  $n$ -tuple error  $\mathbf{e}$ , where elements are Pauli matrices, is able to transform into  $2n$ -bit vector  $(\mathbf{e}_x \mathbf{e}_z)$ , where  $\mathbf{e}_x$  and  $\mathbf{e}_z$  are two  $n$ -bit vectors which can be viewed as errors from two independent BSCs, respectively. Then, for a CSS code with the form (3), we obtain the syndrome  $\mathbf{s} = (\mathbf{s}_x \mathbf{s}_z)$ , where  $\mathbf{s}_x = H_Z \cdot \mathbf{e}_x$  and

$\mathbf{s}_z = H_X \cdot \mathbf{e}_z$ . Then,  $\mathbf{s}_x$  and  $\mathbf{s}_z$  are independently fed to the BP decoding algorithm. Eventually, the logical error rate  $P_{\text{CSS}}$  for the CSS code is  $P_X + P_Z - (P_X \cdot P_Z)$ , where  $P_X$  and  $P_Z$  denote the classical logical error rates for codes  $C_X$  and  $C_Z$ , respectively.

In this article, since we consider the symmetric HGP structure for which  $H_1 = H_2$  and the transmission is over the two independent BSCs that have an identical  $p$ , the decoding performance for  $H_Z$  and  $H_X$  are nearly identical. Thus, we can only focus on one type of error and one parity-check matrix, for instance,  $\mathbf{e}_x$  from the bit-flip channel and  $H_Z$ . In the following description, we will omit the subscript of  $\mathbf{e}_x$  and  $\mathbf{s}_x$ , denoted as  $\mathbf{e}$  and  $\mathbf{s}$ , respectively.

Given an  $M \times N$  matrix  $H_Z$  from the HGP structure, and the measured syndrome  $\mathbf{s} \in F_2^M$  as the input, where  $M = N - K$ , BP decoding can be conducted on the Tanner graph for  $H_Z$ , where each error location corresponds to a variable node and each syndrome location corresponds to a check node. Define  $j$  and  $i$  as the index for the variable node and the check node, where  $0 \leq j < N$  and  $0 \leq i < M$ , respectively. The index set of variable nodes neighboring the  $i$ th check node is defined as  $\mathcal{N}(i) = \{j | H_{Zij} = 1\}$ , and vice versa. Let  $L_j^k$  be the LLR for the  $j$ th variable node at the  $k$ th iteration.  $Q_{ij}^k$  denotes the message from the  $j$ th variable node to the  $i$ th check node at the  $k$ th iteration, and  $R_{ij}^k$  denotes the message from the  $i$ th check node to the  $j$ th variable node at the  $k$ th iteration. The maximum number of iterations of BP decoding is denoted as  $I_{\text{max}}$ . To reduce the computational complexity, we use normalized min-sum BP decoding [42], [43], which is shown as follows.

- 1) *Initialization*: Set  $k = 0$ . Since there is no intrinsic information from the channel, we initialize each variable node  $j$  with the prior LLR, i.e.,  $L_j^0 = \ln \frac{1-p}{p}$ . Initialize  $R_{ij}^{(0)}$  with  $L_j^0$  for all  $j$  and  $i \in \mathcal{N}(j)$ .
- 2) *Vertical message exchange*: For  $0 \leq i < M$ ,  $j \in \mathcal{N}(i)$ , compute

$$Q_{ij}^k = \begin{cases} R_{ij}^{(k)}, & k = 0 \\ L_j^{(k)} - R_{ij}^{(k-1)}, & k > 0. \end{cases} \quad (6)$$

- 3) *Horizontal message exchange*: For  $0 \leq i < M$ ,  $j \in \mathcal{N}(i)$ , compute

$$R_{ij}^k = (-1)^{s_i} \cdot \beta \cdot \prod_{j' \in \mathcal{N}(i), j' \neq j} \text{sgn}(Q_{ij'}^k) \cdot \min_{j' \in \mathcal{N}(i), j' \neq j} |Q_{ij'}^k| \quad (7)$$

where  $s_i$  denotes the  $i$ th element in the measured syndrome  $\mathbf{s}$  and  $\beta = 1 - 2^{-k}$  denotes a scaling factor [44].

- 4) *Elementwise LLR update*: For  $0 \leq j < N$ , update  $L_j^{k+1}$  as

$$L_j^{k+1} = L_j^0 + \sum_{i \in \mathcal{N}(j)} R_{ij}^k. \quad (8)$$

5) *Hard decision*: For  $0 \leq j < N$

$$\hat{e}_j^k = \begin{cases} 0, & L_j^{k+1} > 0 \\ 1, & L_j^{k+1} \leq 0 \end{cases} \quad (9)$$

where  $\hat{e}_j^k$  denotes the  $j$ th element of the estimated error  $\hat{\mathbf{e}}^k$  for the  $k$ th iteration. Then, the estimated syndrome at the  $k$  iteration is  $\hat{\mathbf{s}}^k = \hat{\mathbf{e}}^k \cdot H_Z^T$ .

- a) If the decoding is able to converge to the measured syndrome, i.e.,  $\hat{\mathbf{s}}^k = \mathbf{s}$ , declare as a success.
- b) Otherwise, set  $k = k + 1$  and repeat from Step 2. If  $k = I_{\max}$ , then declare as a failure.

Note that for BP decoding on classical codes, to compute  $R_{ij}^k$ , we do not need to compute  $(-1)^{s_i}$  as in (7) since the prior LLR  $L_j^0$  contains the transmitted information from the channel. While using syndrome-based BP decoding on quantum codes, we cannot directly measure the transmitted information. Thus, we set  $L_j^0$  for each variable node by using the given cross probability  $p$  of BSC. To compute  $R_{ij}^k$ , the  $i$ th element of the measured syndrome  $s_i$  is aided to update the LLR for the  $j$ th variable node. Eventually, the  $j$ th element of the estimated error is obtained by summing the prior LLR together with all the information sent to the  $j$ th variable node.

### III. PROPOSED ALGORITHM

In this section, we will first show the structure of the parity check matrix in HGP form. Then, from this structure, we find that except for weight-1, for other low-weight errors, i.e., where the weight is equal to or larger than 2, another error exists with the same weight and syndrome as the given error and the corresponding syndrome. Thus, the BP decoding algorithm cannot converge to the measured syndrome. To improve the BP decoding on topological codes, we propose a BSFBP decoding approach inspired by observations through the BP decoding process of those errors. Although the proposed decoder is based on 2-D topological codes, it can also be applied to the 3-D versions, which we will show in Section IV.

#### A. HGP STRUCTURE FOR TOPOLOGICAL CODES

Although the locality nature of topological codes constructed from the HGP form is beneficial to the real implementation, it also leads to lower row and column weights from the perspective of the parity-check matrix, which will cause ambiguity between different errors with the same weights.

Considering  $H_Z$  constructed based on an  $m_b \times n_b$  parity-check matrix  $H_b$  using the symmetric HGP structure, i.e.,  $M = m_b \cdot n_b$ ,  $N = n_b^2 + m_b^2$ , we denote the first  $n_b^2$  columns as section  $S_1$ , which corresponds to  $I \otimes H_b$ , and the final  $m_b^2$  columns as section  $S_2$ , which corresponds to  $H_b^T \otimes I$ . For example, in Fig. 1, the first 81 columns from left to right together represent section  $S_1$  of  $H_Z$  for the [[162,2,9]] toric code and the remaining 81 columns together represent section  $S_2$ . Furthermore,  $S_1$  can be divided into  $n_b$   $H_b$ -type

block columns where each  $H_b$ -type block column contains  $n_b$  columns. Likewise,  $S_2$  can be divided into  $m_b$   $I$ -type block columns where each  $I$ -type block column contains  $m_b$  columns. For the [[113,1,8]] surface code shown in Fig. 2, all the representations are the same as for the toric code, except  $H_b$ , as shown in Fig. 3(b).

Now, we will demonstrate the correspondence between the error and the syndrome by considering the weight-1 and weight-2 errors. In Statement 1, we show that each weight-1 error yields a unique syndrome. We denote the set of the nonzero element location for a given vector  $\mathbf{v}$  as  $\mathcal{I}_v$ .

*Statement 1*: For topological codes based on the symmetric HGP structure, given an arbitrary weight-1 error  $\mathbf{e}$  with syndrome  $\mathbf{s}$ , there is no other weight-1 error having the identical syndrome  $\mathbf{s}$ .

*Proof*: Suppose an error  $\mathbf{e}$  with  $\mathcal{I}_e = \{j\}$ , where  $j$  is located at  $S_1$  and its corresponding syndrome  $\mathbf{s}$  with  $\mathcal{I}_s = \{j, j-1 \pmod{m_b}\}$ ; if  $m_b \mid j$ , then  $\mathcal{I}_s = \{j, j-1 \pmod{m_b} + m_b\}$ .

- 1) No other weight-1 error  $\mathbf{e}'$  where  $\mathcal{I}_{e'} = \{j'\}$ , where  $j' \neq j$  and  $j'$  and  $j$  are in the same block column in  $S_1$ , has the identical syndrome  $\mathbf{s}$  since  $H_b$  is the parity-check matrix for the repetition code.
- 2) No other weight-1 error  $\mathbf{e}'$  where  $\mathcal{I}_{e'} = \{j'\}$ , where  $j' \neq j$  and  $j'$  and  $j$  are in different block columns in  $S_1$ , has the identical syndrome  $\mathbf{s}$  since these two syndromes do not have overlapping locations of nonzero elements.
- 3) No other weight-1 error  $\mathbf{e}'$  where  $\mathcal{I}_{e'} = \{j'\}$ , where  $j'$  is in  $S_2$ , has the identical syndrome  $\mathbf{s}$ . This is evident from Figs. 1 and 2. ■

The case where an error has a nonzero element located at  $S_2$  with syndrome  $\mathbf{s}$  is similar since no other weight-1 error in  $S_1$  or  $S_2$  has a corresponding syndrome identical to  $\mathbf{s}$ . That is, given a weight-1 error, it corresponds to a unique syndrome.

In contrast, given a weight-2 error, another weight-2 error might exist where these two errors have an identical syndrome. For example, a special case is given as follows.

*Statement 2*: Given a weight-2 error  $\mathbf{e}$  where  $\mathcal{I}_e = \{j, j+1\}$ , where  $j$  and  $j+1$  are in the same block column in  $S_2$ , a weight-2 error exists, which has an identical syndrome to  $\mathbf{e}$ .

*Proof*: For a toric code, where  $m_b = n_b$ , the weight-2 error  $\mathbf{e}$  will yield the corresponding syndrome  $\mathbf{s}$  where  $\mathcal{I}_s = \{\bar{j}, \bar{j} + n_b, \bar{j} + 1, \bar{j} + 1 + n_b\}$ , where  $\bar{j} = j - n_b^2$ . We are able to identify an error  $\mathbf{e}'$  where  $\mathcal{I}_{e'} = \{j', j' + n_b\}$ , where  $j'$  and  $j' + n_b$  are in different block columns in  $S_1$ . The corresponding syndrome  $\mathbf{s}'$  has an identical  $\mathcal{I}_{s'} = \{j', j' - 1, j' + n_b, j' - 1 + n_b\}$  to the syndrome of  $\mathbf{e}$  if we select  $j' = \bar{j} + 1$ . For a surface code, where  $m_b = n_b - 1$ , the weight-2 error  $\mathbf{e}$  will yield the corresponding syndrome  $\mathbf{s}$ , where  $\mathcal{I}_s = \{\bar{j}, \bar{j} + m_b, \bar{j} + 1, \bar{j} + 1 + m_b\}$ . We are able to identify an error  $\mathbf{e}'$  where  $\mathcal{I}_{e'} = \{j', j' + n_b\}$ , where  $j'$  and  $j' + n_b$  are in the different block columns in  $S_1$ . Its corresponding syndrome  $\mathbf{s}'$  contains  $\mathcal{I}_{s'} = \{\bar{j}', \bar{j}' - 1, \bar{j}' - 1 +$

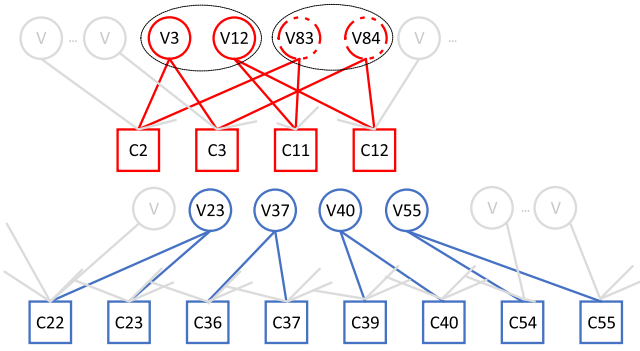


FIGURE 4. Tanner graph for the  $[[162,2,9]]$  toric code.

$n_b - 1, \bar{j}' + n_b - 1\}$ , where  $\bar{j}' = j' - \lceil \frac{j'}{n_b} \rceil$ . Pick  $\bar{j}' = \bar{j} + 1$ ; then, the two syndromes are identical since  $m_b = n_b - 1$ . ■

For the case of larger weights such as a weight-3 case, we can combine a weight-1 error pattern with a weight-2 error pattern described above. Then, another weight-3 error exists with the identical syndrome based on the statements above.

### B. BP DECODING FOR TOPOLOGICAL CODES

In this article, we will consider the  $[[162,2,9]]$  and  $[[242,2,11]]$  toric codes together with the  $[[113,1,8]]$  and  $[[181,1,10]]$  surface codes from [25] in order to demonstrate the error rate performance. For the specific topological codes mentioned above, using the weight-1 case, though no strict proof of the convergence to the measured syndrome exists for the BP decoding, the simulation achieved by tracing all weight-1 errors shows that BP decoding is able to converge to the measured syndromes. However, when the weight is greater than or equal to 2, as described in Section III-A, for such topological codes with both low row-weight and column-weight of the parity-check matrices, many pairs of errors that have an identical syndrome will exist such that the BP decoding cannot distinguish from the syndrome and will not converge to the measured syndrome.

We will show an example of the BP decoding process for the  $[[162,2,9]]$  toric code. Given the error  $\mathbf{e}$  where  $\mathcal{I}_e = \{3, 12, 23, 37, 40, 55\}$  (solid circles), its corresponding syndrome  $\mathbf{s}$  has  $\mathcal{I}_s = \{2, 3, 11, 12, 22, 23, 36, 37, 39, 40, 54, 55\}$  (solid squares). Fig. 4 shows the related section of the Tanner graph for this code, where variable node  $v_j$  corresponds to the  $j$ th error location and check node  $c_i$  corresponds to the  $i$ th syndrome location. We will use both names interchangeably. Furthermore, the error (or syndrome) pattern is denoted as the set of locations of a fraction of the nonzero elements of the error (or syndrome). Then, it can be shown from Fig. 4 that the weight-2 error pattern  $\{83, 84\}$ , shown as red dashed circles, and the weight-2 error pattern  $\{3, 12\}$ , shown as red solid circles, yield the identical syndrome pattern  $\{2, 3, 11, 12\}$ . These error patterns that have identical weights yielding an identical syndrome pattern, denoted in red, are indistinguishable to the BP decoder. Otherwise, if the given syndrome pattern corresponds to a unique

TABLE 1. Estimated Error and Syndrome for Each Iteration Using the BP Decoding

$\mathcal{I}_e$ of real error $\mathbf{e}$	$\mathcal{I}_s$ of measured syndrome $\mathbf{s}$
3, 12, 23, 37, 40, 55	2, 3, 11, 12, 22, 23, 36, 37, 39, 40, 54, 55
Iter. $k$	$\mathcal{I}_{\hat{\mathbf{e}}^k}$ of estimated error $\hat{\mathbf{e}}^k$
0th	3, 12, 23, 37, 40, 55, 83, 84
1st	13, 22, 23, 28, 37, 38, 39, 40, 49, 55, 64, ...
...	...
122nd	12, 37, 40, 84
123rd	12, 23, 37, 40, 55
124th	3, 12, 23, 37, 40, 55, 83
125th	23, 37, 40, 55
126th	3, 12, 23, 37, 40, 55, 84
127th	23, 37, 40, 55
...	...

error pattern with the minimum weight, then we call such an error pattern distinguishable and it is denoted in blue. At the zeroth iteration in BP decoding, all the indistinguishable error patterns corresponding to the identical syndrome occur, as shown in Table 1. This can be deduced from the BP decoding. For instance,  $R_{i,j}^0 = -\beta \ln \frac{1-p}{p}$  since  $s_i = 1$  and  $Q_{ij} = \ln \frac{1-p}{p}$  in (7) for  $i = 2, 3, 11, 12, \forall j \in \mathcal{N}(i)$ . Then, in (8),  $L_j^1 = -2\beta \ln \frac{1-p}{p} + \ln \frac{1-p}{p} < 0$  for  $j = 3, 12, 83, 84$ . Thus, the weight for the estimated syndrome  $\mathbf{s}^0$  is less than the measured syndrome  $\mathbf{s}$  since the syndrome pattern  $\{2, 3, 11, 12\}$  corresponding to an indistinguishable error pattern will become zero due to the addition under mod 2 if all the indistinguishable error patterns occur.

As iterations increase, messages will pass between variable nodes and check nodes, and until the 122nd iteration, most of the locations of the nonzero elements in  $\hat{\mathbf{e}}^k$  are identical to the locations of nonzero elements in  $\mathbf{e}$ , as shown in Table 1. One can see that pattern  $\{23, 37, 40, 55\}$  is distinguishable as it is fixed following the 123rd iteration. When the maximum number of iterations is reached, BP decoding will not converge to the measured syndrome due to those indistinguishable error patterns.

To improve the BP decoding, three phenomena can be used as aids. First, we find that most of the time, at the zeroth iteration, the hard-decision output will include all the indistinguishable error patterns. Second, we find that at some iterations, the estimated syndromes are close to the measured syndrome such as the 123rd, 124th, and 126th iteration, as shown in Table 1. Last but not least, we find that those distinguishable error patterns and their corresponding syndrome patterns might be auxiliary to decoding those indistinguishable patterns at the beginning of the decoding process due to the interactive message passing between them.

### C. NEW DECODING PATH BRANCHED FROM BP

Although indistinguishable error patterns may interfere with the BP decoding to converge to the measured syndrome, some estimated syndromes exist that are close to the measured syndrome for certain iterations. Thus, we thought it

might be feasible to perform another new path of the BP decoding, hereafter called the branch and the original path is denoted as the trunk, using the information from those estimated syndromes. We expect that the syndromes used for the branch correspond to less indistinguishable error patterns and are close to the measured syndromes. To achieve this goal, we introduce a syndrome-pruning process so as to obtain the syndrome residual from both the measured syndrome and the specific estimated syndrome, and view it as the input for the branch. In addition, to identify such a specific estimated syndrome, we need to design some criteria during the iterative decoding.

First, a benchmark is needed in order to compare the closeness of the estimated syndrome to the measured syndrome. Most of the time, at the zeroth iteration, the estimated error contains all the possible indistinguishable error patterns. During the decoding process, if some indistinguishable error patterns disappear at a specific iteration, the corresponding estimated syndrome will be close to the measured syndrome, compared to the estimated syndrome  $\hat{s}^0$ . Thus, it is reasonable to set  $\hat{s}^0$  as the zeroth benchmark  $\mathbf{b}$ . To identify an estimated syndrome that is closer to the measured syndrome than  $\hat{s}^0$ , we use the Hamming distance as a metric since the syndrome can be viewed as a binary vector. That is, the first criterion is introduced as

$$C.1 \quad w(\hat{s}^k \oplus \mathbf{s}) \leq w(\mathbf{b} \oplus \mathbf{s})$$

where  $w(\mathbf{c})$  is the Hamming weight for a vector  $\mathbf{c}$ , and  $\oplus$  is the addition under mod 2.

To further determine that the selected estimated syndrome is part of the measured syndrome and, thus, given the validity of syndrome pruning, we introduce the second criterion:

$$C.2 \quad \text{Set } \mathcal{I}_{\hat{s}^k} = \{i \in \{0, 1, \dots, M-1\} \mid s_i^k = 1\} \subseteq \mathcal{I}_{\mathbf{s}} = \{i \in \{0, 1, \dots, M-1\} \mid s_i = 1\}.$$

When the estimated syndrome for the  $k$ th iteration  $\hat{s}^k$ ,  $k \neq 0$ , satisfies both **C.1** and **C.2**, the syndrome-pruning process is conducted. The syndrome residual  $\tau_s$  is obtained by pruning  $\mathbf{s}$  using  $\hat{s}^k$ , i.e.,  $\tau_s = \mathbf{s} - \hat{s}^k$ . For this decoding branch,  $\tau_s$  is seen as the input for the BP decoding, which means that  $\tau_s$  replaces the  $\mathbf{s}$  in (7). Suppose that at the  $\ell$ th iteration in the branch, if the estimated syndrome corresponding to the error  $\hat{e}^\ell$  is identical to  $\tau_s$ , then we can declare that the decoding process is able to converge to the measured syndrome since  $H_Z \cdot (\hat{e}^k + \hat{e}^\ell) = \hat{s}^k + \tau_s = \mathbf{s}$ . Otherwise, when the maximum number of iterations in the branch is reached, the process will return to the trunk of the BP decoding. Meanwhile,  $\mathbf{b}$  is replaced by  $\hat{s}^k$ , which can not only result in a lower  $w(\mathbf{b} \oplus \mathbf{s})$  for the next branch but also cope with the case where not all indistinguishable error patterns are included at the zeroth iteration.

We will show how such branch-assisted belief propagation (BBP) works and how both criteria **C.1** and **C.2** and the syndrome-pruning process are able to obtain the syndrome residual by using Fig. 4 and Table 1. Considering BBP decoding, the zeroth estimated syndrome  $\hat{s}^0$  will be recorded as the

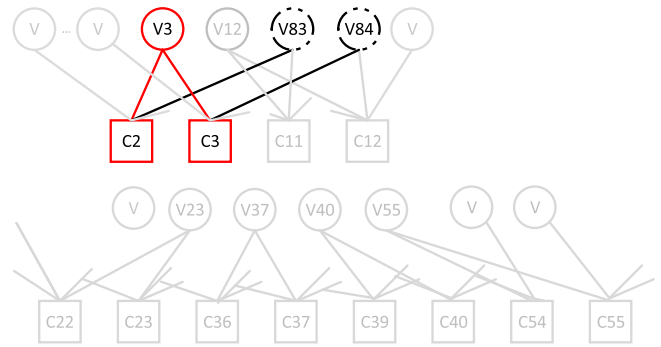


FIGURE 5. Example of the BBP decoding for the  $[[162,2,9]]$  toric code.

TABLE 2. Reduction Ratio of Unsolvable Weight-2 and -3 Errors for BBP Compared to BP

Codes	$[[162, 2, 9]]$	$[[242, 2, 11]]$	$[[113, 1, 8]]$	$[[181, 1, 10]]$
$w=2$	49.6%	69.7%	49.6%	51.2%
$w=3$	56.7%	78.0%	64.1%	63.6%

zeroth benchmark, where  $w(\mathbf{b} \oplus \mathbf{s}) = 4$ . At each iteration, only the estimated syndromes satisfying **C.1** will be identified. In Table 1, since  $w(\hat{s}^{123} \oplus \mathbf{s}) = 2 \leq 4$  and  $\hat{s}^{123}$  satisfies **C.2**, the estimated error  $\hat{e}^{123}$  is recorded. Then, the measured syndrome is pruned using the estimated syndrome  $\hat{s}^{123}$ , and the syndrome residual  $\tau_s$ , where  $\mathcal{I}_{\tau_s} = \{2, 3\}$ , will be seen as the input to the new decoding branch. Since  $\tau_s$  corresponds to a unique weight-1 error, as shown in Fig. 5, according to Statement 1 in Section III-A, the decoding algorithm is able to converge to a unique syndrome. Indeed, combining the recorded  $\hat{e}^{123}$  and  $\hat{e}_\tau$  where  $\mathcal{I}_{\hat{e}_\tau} = \{3\}$ , it can be shown that the channel error  $\mathbf{e}$  is found.

From the simulation of the considered topological codes, it can be shown that our BBP decoding algorithm is able to converge to the measured syndrome for those weight-2 and weight-3 errors that the BP decoding is able to converge. Thus, we will focus on those weight-2 and weight-3 errors that the BP decoding cannot solve, i.e., meaning that it cannot converge to the measured syndrome. Table 2 shows that the BBP decoding process is able to reduce the unsolvable weight-2 errors from the BP decoding for the toric codes and surface codes considered by roughly 50%. In addition, when extended to the case of weight-3 errors, our BBP algorithm is able to reduce the number of unsolvable errors by about 60%.

Now, we will discuss the difference in performance between the BP and BBP decoding processes based on the same complexity, i.e., the same average number of iterations. Following the settings used in [25], the maximum number of iterations for the BBP decoding is set to  $N$  for both the toric codes and the surface codes. To compare fairly, the comparison between the BP and BBP decoding will be under nearly the same average number of iterations. For the BBP decoding, the average number of iterations is computed using both the maximum iteration  $I'_{\max}$  for the trunk and  $I''_{\max}$  for the branch. Fig. 6 shows that for the  $[[162,2,9]]$  toric code, the

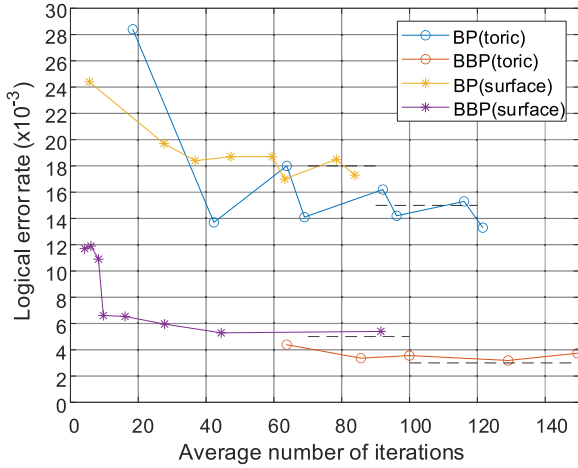


FIGURE 6. Logical error rate versus average number of iterations at  $p = 0.01$ .

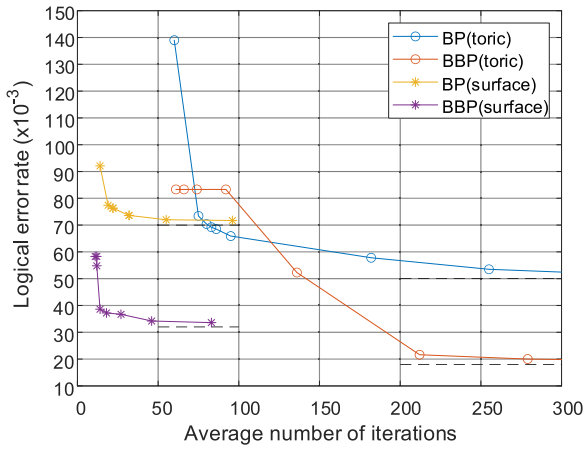


FIGURE 7. Logical error rate versus average number of iterations at  $p = 0.02$ .

logical error rate when using BBP decoding will converge to  $3 \times 10^{-3}$  when the average number of iterations is 80, while BP decoding only converges to  $1.5 \times 10^{-2}$  based on the same average number of iterations. For the  $[[113,1,8]]$  surface code illustrated in Fig. 2, the logical error rate for BBP decoding will decrease when the average number of iterations is 8. Fig. 7 shows that the comparison between the BP and BBP decoding for the case where there is a higher crossover probability, where similar comparison results can be observed.

In brief, the BBP algorithm introduces a new decoding path with syndrome residual  $\tau_s$  as its branch input. In this way,  $\tau_s$  in (7) will correspond to the error having less indistinguishable patterns, and the new decoding might eventually converge to  $\tau_s$ . It is interesting to explore the introduction of a subbranch within a branch to improve the BBP decoding. However, simulation shows that the performance when having an additional subbranch is similar to the performance

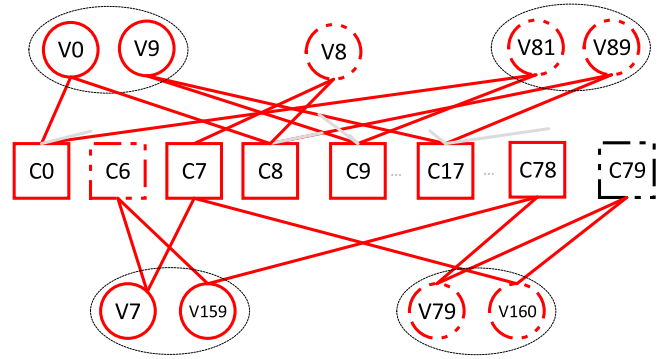


FIGURE 8. Example for the  $[[162,2,9]]$  toric code.

TABLE 3. Estimated Error and Syndrome for Each Iteration Using the BBP Decoding

$\mathcal{I}_e$ of real error $e$	$\mathcal{I}_s$ of measured syndrome $s$		
0, 7, 9, 159	0, 7, 8, 9, 17, 78		
Iter. $k$	$\mathcal{I}_{e^k}$ of estimated error $\hat{e}^k$	$\mathcal{I}_{s^k}$ of estimated syndrome $\hat{s}^k$	Enter the branch ?
0th	0, 8, 9, 81, 89	7, 8	No
...	...	...	...
122nd	8, 9	7, 8, 9, 17	Yes
123rd	8, 9, 81	0, 7, 8, 17	Yes
124th	9, 79, 81, 159	0, 6, 17, 79	No
125th	7, 8, 9, 81, 160	0, 6, 7, 8, 17, 79	No
126th	9, 79, 159	6, 9, 17, 79	No
...	...	...	...

when using the original BBP decoding for the considered topological codes.

For those errors that cannot be solved by the branch, the syndrome residuals still correspond to some indistinguishable error patterns. Here, we give an example as follows. Consider the  $[[162,2,9]]$  toric code, an error  $e$  where  $\mathcal{I}_e = \{0, 7, 9, 159\}$  has the corresponding measured syndrome  $s$  where  $\mathcal{I}_s = \{0, 7, 8, 9, 17, 78\}$ , as shown in Fig. 8. In this example, error pattern  $\{7, 159\}$  and another pattern  $\{79, 160\}$  have the identical corresponding syndrome pattern  $\{7, 78\}$ , and error patterns  $\{0, 9\}$  and  $\{81, 89\}$  have the identical corresponding syndrome pattern  $\{0, 8, 9, 17\}$ . While decoding syndrome pattern  $\{7, 8\}$ , error pattern  $\{8\}$  is a more reasonable candidate for the decoding since it has a lower weight compared to those real error patterns mentioned above. Thus, the observation at the zeroth iteration will violate the assumption of the BBP decoding, as shown in Table 3, since  $L_7^1 = L_{159}^1 = L_{79}^1 = L_{160}^1 = -\beta \ln \frac{1-p}{p} + \ln \frac{1-p}{p} > 0$  in (8). Table 3 shows that, after sufficient iterations, the nonzero elements of the estimated error all belong to one of the indistinguishable error patterns, and even the BBP decoding will fail to converge to the measured syndrome.

#### D. SF PROCESS INTRODUCED IN THE TRUNK

To decode such an error described above, a heuristic method is to perturb the bit LLR in an indistinguishable error pattern. Another method inspired by the third observation in Section III is to introduce an error pattern not containing



**TABLE 4. Estimated Error and Syndrome for Each Iteration Using the BSFBP Decoding**

$\mathcal{I}_e$ of real Error $e$		$\mathcal{I}_s$ of measured Syndrome $s$		
0, 7, 9, 159		0, 7, 8, 9, 17, 78		
Iter. $k$	$\mathcal{I}_{e^k}$ of estimated error $e^k$	$\mathcal{I}_{s^k}$ of estimated syndrome $s^k$	Enter the branch ?	Target $v_j$
0th	0, 8, 9, 81, 89	7, 8	No	81
...	...	...	...	...
53rd	8, 9, 79, 81, 159	0, 6, 7, 8, 17, 79	No	79
54th	7, 8, 9, 81, 159	0, 8, 17, 78	Yes	10
55th	7, 79, 82, 159	1, 7, 10, 79	No	79
56th	0, 1, 7, 89, 159	1, 7, 17, 78	No	81
57th	7, 159	7, 78	No	9
58th	7, 9, 159	7, 9, 17, 78	Yes	—

any location of the indistinguishable patterns. Then, due to the interactive message passing, there will be more different estimated syndromes than the original BBP decoding, which will also provide more diverse syndrome residuals for the BBP decoding. Thus, we propose an SF process for our BBP decoding algorithm where we change the sign of the posterior LLR for the selected variable node denoted in (8).

To decide which variable node's LLR sign should be flipped, we will first use the difference between the estimated syndrome  $\hat{s}^k$  at the  $k$ th iteration and the measured syndrome  $s$  in the trunk. At each iteration in the decoding process, the indices  $i$  of the mismatched check nodes, i.e.,  $\hat{s}_i^k \neq s_i$ , are recorded. The process is shown as follows.

- 1) After computing  $\hat{s}^k$  in the trunk, if  $\hat{s}^k \neq s$ , we record the different indices between  $\hat{s}^k$  and  $s$  as set  $U = \mathcal{I}_{\hat{s}^k \oplus s} = \{i | \hat{s}_i^k \neq s_i, 0 \leq i < M\}$ .
- 2) Find a variable node  $v_j$  based on  $U$  and flip the sign of  $L_j^{k+1}$  in (8) and move to the next iteration in the trunk.

In Step 2), we consider three different strategies to select the variable node whose LLR will be flipped. These strategies depend on the attributes of the chosen variable node. One reasonable strategy is to select the variable node involved in the largest number of mismatched check nodes, i.e., this variable node is likely to result in an estimated syndrome close to the measured syndrome. Since our SF process operates within the iterative decoding process, another strategy is to impose a disturbance on a randomly selected variable node related to the mismatched check node, which will provide diverse syndrome residuals and enhance the probability of convergence to the measured syndrome by the BBP decoding algorithm. Finally, we also consider a balanced strategy by comparing the LLR for those variable nodes neighboring the mismatched check nodes.

For now, we will show how our BSFBP works for the example above by using the so-called Global selection strategy. For example, compared to Table 3, Table 4 shows that at the 55th iteration, the SF process flips the LLR for variable node  $v_{10}$ , which results in an estimated error containing a distinguishable pattern {82}. At first glance, it seems that the flipping procedure moves the estimated syndrome away from

**TABLE 5. Reduction Ratio <sup>a</sup> for Unsolvable Weight-2 and Weight-3 Errors for BSFBP Decoding Compared to BP Decoding**

Codes	Strat.	[[162, 2, 9]]	[[242, 2, 11]]	[[113, 1, 8]]	[[181, 1, 10]]
$w=2$	1	99.17%	99.68%	90.27%	91.71%
	2	99.59%	99.68%	99.56%	94.08%
	3	99.17%	99.35%	87.17%	93.84%
$w=3$	1	99.48%	99.79%	85.55%	91.69%
	2	99.54%	99.75%	85.70%	92.07%
	3	99.62%	99.78%	85.78%	92.25%

<sup>a</sup>Since the SF process involves random selection, it cannot guarantee the successful decoding of every error that a single simulation of BP can converge on. Thus, we show the average reduction ratio in this table.

the measured syndrome. Nevertheless, the sign of the posterior LLR for  $v_8$  is also indirectly flipped to positive in the next iteration, and the magnitude of its LLR is increased in the following iterations. At the 57th iteration, the decoder successfully identifies the error pattern {7, 159} instead of error pattern {8}, and the syndrome pattern {7, 78} is fixed. Then, at the 58th iteration, syndrome residual  $\tau_s$  where  $\mathcal{I}_{\tau_s} = \{0, 8\}$  corresponds to a weight-1 error  $\hat{e}_\tau$ , where  $\mathcal{I}_{\hat{e}_\tau} = \{0\}$ , which can be solved by the BBP decoding algorithm according to Statement 1 in Section III-A.

Three variable node selection strategies are described as follows.

- S.1 *Global selection strategy*: This process will perform a global search from all the variable nodes neighboring all the mismatched check nodes and select the variable node which has the most neighboring check nodes with indices belonging to  $U$ , i.e., select the variable node  $v_j$ , where  $j \in \mathcal{N}(i), \forall i \in U$  such that  $|\mathcal{N}(j) \cap U|$  is the maximum for all  $E \in P(U)$ , where  $|\cdot|$  denotes the cardinality of the set and  $P(U)$  denotes the power set of  $U$ . This strategy needs to compare at most  $d_{c_i} \cdot d_{v_i}$  candidate variable nodes.
- S.2 *Reliability-based selection strategy*: The process will select the variable node with the least reliability, which is neighboring a randomly selected check node whose indices belong to  $U$ , i.e., randomly select a check node  $c_i$ , where  $i \in U$ . Select a variable node  $v_j$ , where the reliability  $|L_j^k| \leq |L_{j'}^k|$  for all  $j, j' \in \mathcal{N}(i), j' \neq j$ . In this way, this strategy only needs to compare  $d_{c_i}$  candidate variable nodes.
- S.3 *Random selection strategy*: The process will randomly select a variable node neighboring a randomly selected check node whose indices belong to  $U$ , i.e., select a variable node  $v_j, j \in \mathcal{N}(i), i \in U$ . The complexity is the lowest among these three strategies.

We show the reduction ratio of unsolvable weight-2 and weight-3 errors for the BSFBP decoding compared to the BP decoding in Table 5, which shows that the BSFBP decoding can reduce the number of unsolvable weight-2 and weight-3 errors for the considered toric codes across all selection strategies by up to 99%. For the considered surface codes, the reduction ratio is about 90% for both weight-2 and weight-3 errors compared to the BP decoding process. Overall, our

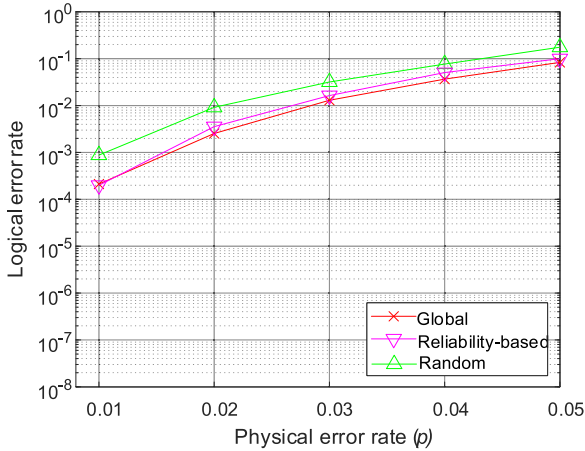


FIGURE 9. Performance comparison for BSFBP using different selection strategies based on the  $[[162,2,9]]$  toric code.

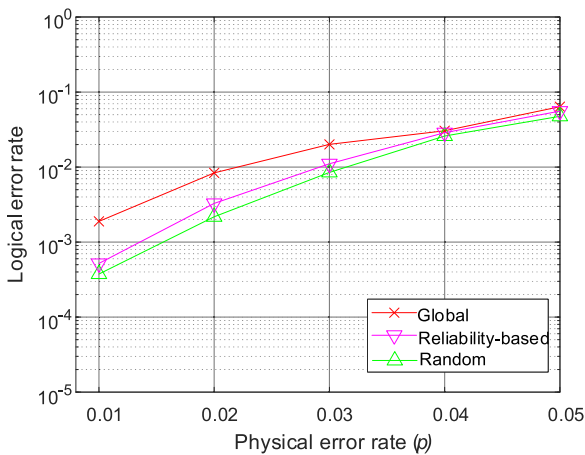


FIGURE 10. Performance comparison for BSFBP using different selection strategies based on the  $[[113,1,8]]$  surface code.

BSFBP decoding algorithm is able to converge to the measured syndrome between 99.48% and 99.99% of weight-2 and weight-3 errors depending on the codes considered.

Figs. 9 and 10 show the performance when using different selection strategies. These show that the relationship between strategies and performance depends on the codes. For example, the performance when using S.1 and S.2 for toric codes is almost identical, while, for the surface codes, the randomness of the selection seems to assist with the decoding. A possible explanation is that surface codes have an irregular parity-check matrix. Here, the term “regular” for the parity-check matrix refers to the fact that each row and column has a uniform weight. Otherwise, the parity-check matrix is called “irregular,” which is equivalent to saying that the degree of all check nodes or variable nodes is not equal. Unlike toric codes, where every variable node has two neighboring check nodes, i.e., degree 2, the considered surface codes have degree-2 and degree-1 variable nodes, i.e., some of the variable nodes are only neighboring one check node.

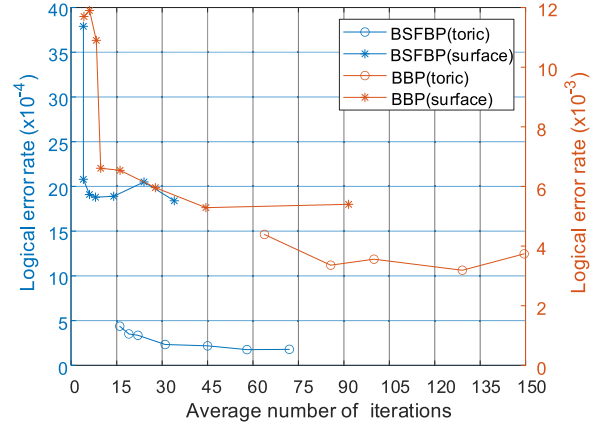


FIGURE 11. Logical error rate versus the average number of iterations at  $p = 0.01$ .

Since the criterion for the global selection S1 is to identify the variable node that has the most mismatched neighboring check nodes, this selection is affected by the degree of the variable node and is prone to choosing variable nodes that have higher degrees, while the random selection S3 randomly chooses a variable node regardless of the degree. Thus, for this case, using S3 provides a slightly better performance. However, it is obvious that the random selection is highly dependent on the maximum number of iterations, while the global selection is more stable. For example, if the maximum number of iterations is set to 30, the performance is almost the same when using either S1 or S3.

We also compare the performance between the BBP and BSFBP decoding based on Strategy S.1 under the same average number of iterations, as shown in Fig. 11. We show that the BSFBP decoding process not only reduces the logical error rate by about one order of magnitude compared to BBP decoding but also reduces the average number of iterations, which means that using the SF process is able to converge much faster.

### E. SUMMARY OF BSFBP

The details for the complete BSFBP decoding process are shown in Algorithm 1. First, based on the structure of topological codes, the specific estimated syndrome is identified using two criteria in Step 9. Then, the syndrome residual is obtained in Step 10 and used as the input to a new branch in the BSFBP decoding algorithm. To provide our decoding with diverse syndrome residuals, we also introduce a flipping element to the posterior LLR in the trunk at Step 21 based on flexible selection strategies. Note that in [17] and [18], a random perturbation and a perturbation based on the symbol of the stabilizers of QLDPC codes are used on the prior channel probabilities, respectively, to refine the initial prior probability for the nonbinary BP decoding. In contrast, our SF process aims to provide diverse syndrome residuals by perturbing the posterior LLR for the binary BP decoding. In summary, our BSFBP decoding is able to mitigate the

**Algorithm 1:** BSFBP Decoding.

- 1: Initialization: Given  $H_Z \in F_2^{M \times N}$ ,  $\mathbf{s} \in F_2^M$ , and channel crossover probability  $p$ , set the trunk iteration index  $k = 0$ . Initialize  $L_j^0 = \ln \frac{1-p}{p}$  and  $R_{ij}^{(0)}$  using  $L_j^0$  for all  $j$  and  $i \in \mathcal{N}(j)$ .
- 2: **while**  $k < I_{\max}^t$  **do**
- 3: Vertical message exchange as (6)
- 4: Horizontal message exchange as (7)
- 5: Elementwise LLR update as (8).
- 6: Hard decision as (9) and set  $\mathbf{b} = \hat{\mathbf{s}}^k$ .
- 7: **if**  $\hat{\mathbf{s}}^k = \hat{\mathbf{e}}^k \cdot H_Z^T = \mathbf{s}$  **then**
- 8: Declare a decoding success.
- 9: **else if**  $\hat{\mathbf{s}}^k$  satisfies both C.1 and C.2 **then**
- 10: Syndrome-pruning process:  $\tau_s = \mathbf{s} \oplus \hat{\mathbf{s}}^k$ .
- 11: Initialization: Use  $\tau_s$  as the syndrome input and set the iteration index  $\ell = 0$  for the branch.
- 12: **while**  $\ell < I_{\max}^b$  **do**
- 13: Repeat (6)–(8) using syndrome input  $\tau_s$  and the iteration index  $\ell$ .
- 14: **if** Hard decision:  $\hat{\tau}_s^\ell = \tau_s$  **then**
- 15: Declare a decoding success.
- 16: **end if**
- 17: Set  $\ell = \ell + 1$  and repeat from Step 13
- 18: **end while**
- 19: When  $\ell = I_{\max}^b$ , set  $\mathbf{b} = \hat{\mathbf{s}}^k$ . Move to Step 21.
- 20: **end if**
- 21: SF process:  $L_j^{k+1} = -L_j^{k+1}$  for the target  $v_j$  from one of the selected strategies.
- 22: Set  $k = k + 1$  and repeat from Step 3.
- 23: **end while**
- 24: Declare a failure when  $k = I_{\max}^t$ .

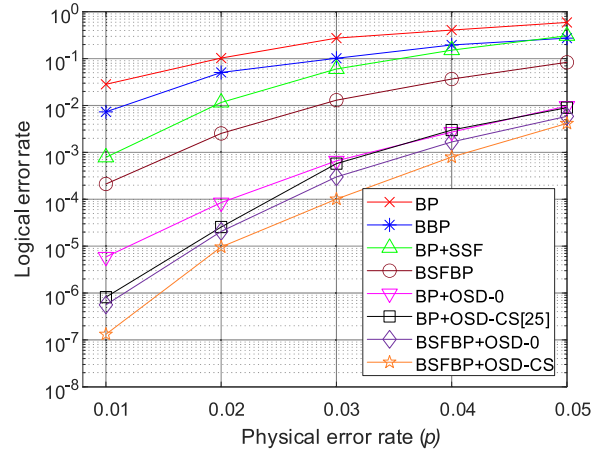
effects of those indistinguishable error patterns and improve the performance of the considered topological codes.

**IV. EXPERIMENTAL RESULTS**

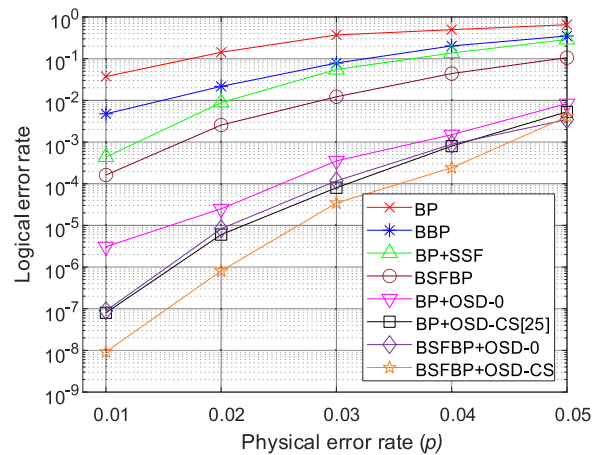
**A. SIMULATION ON TOPOLOGICAL CODES**

From the discussion presented in Section II, it is sufficient to simulate only one type of error. Thus, only the bit-flip channel and  $H_Z$  will be considered for 2-D codes as in [25], while for 3-D codes, the phase-flip channel and  $H_X$  will be considered as in [45]. In the simulation, we will show the performance of our proposed decoding algorithm compared to other methods using binary BP decoding [21], [25] based on the BSC model. For 2-D topological codes, we use variable node selection strategy S.2 in the SF process for toric codes, while for surface codes, S.3 is used. For 3-D topological codes, we use S.2 for both codes due to the lower maximum number of iterations, as discussed in Section III-D.

We show the performances of the [[162,2,9]] and the [[242,2,11]] toric codes in Figs. 12 and 13, respectively. These shows that, since our BBP decoding algorithm uses the structure of topological codes, it is able to surpass the



**FIGURE 12.** Performance comparison for the [[162,2,9]] toric code using different decoding algorithms.



**FIGURE 13.** Performance comparison for the [[242,2,11]] toric code using different decoding algorithms.

performance of conventional BP decoding. The small-set-flip (SSF) can also be applied directly on regular HGP codes [22]. However, the performance of HGP codes with code lengths below 500 using SSF decoding is not significant, as shown in [46], and even worse than using BP decoding for the toric codes considered. Thus, we consider the BP decoding concatenated to SSF (BP+SSF). BP+SSF uses BP decoding to yield an estimated error and then searches for all the supports of the  $M$  stabilizer generators and adds to the error using the support vectors, which is able to repeatedly reduce the corresponding syndrome to zero. Consequently, most of the computation time is due to SSF. In this article, SSF will only operate when BP decoding reaches a predetermined fixed number of iterations, which is set to an average number of iterations. This method can be viewed as a variant of the hybrid decoder presented in [21] to reduce the time complexity. Figs. 12 and 13 show that our BSFBP decoding algorithm can outperform BP+SSF when the predetermined fixed iterations are set to 20 and 30 for the [[162,2,9]] and

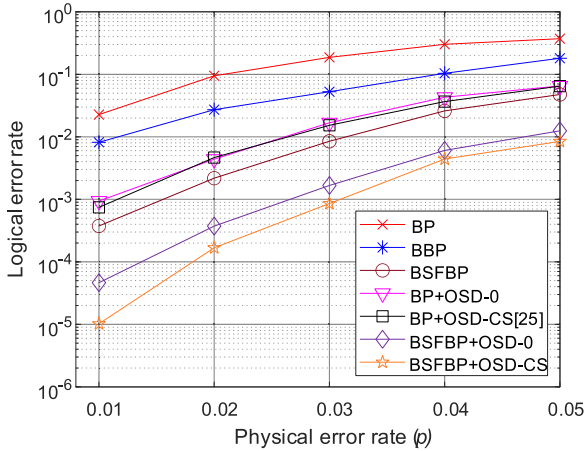


FIGURE 14. Performance comparison for the  $[[113,1,8]]$  surface code using different decoding algorithms.

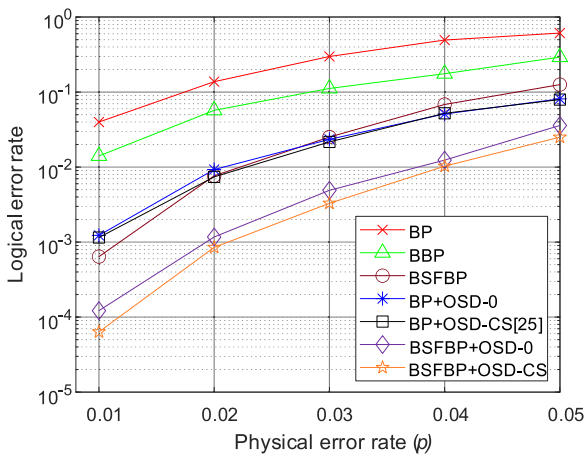


FIGURE 15. Performance comparison for the  $[[181,1,10]]$  surface code using different decoding algorithms.

the  $[[242,2,11]]$  toric codes, respectively. Both the figures show that the proposed BSFBP decoding algorithm is able to improve the logical error rate close to  $10^{-4}$  when  $p = 0.01$  and surpass the conventional BP decoding method by about two orders of magnitude.

If we use our BSFBP decoding method as an auxiliary to OSD, the performance can be improved significantly. We consider OSD-0 and the higher order OSD using a combination sweep method (OSD-CS), as described in [25]. OSD-0 supposes that nonzero elements of an error are more likely to exist in those locations of the error that have a lower reliability rather than in other locations where the reliability is higher. In contrast, OSD-CS will consider the case that some nonzero elements will exist in locations that have a higher reliability. Figs. 12 and 13 show that the performance of our BSFBP decoding process concatenated with OSD-0 is almost identical to the BP decoding method concatenated with OSD-CS. When the BSFBP decoding process is concatenated with OSD-CS, it is able to reach a logical error rate of about  $10^{-7}$

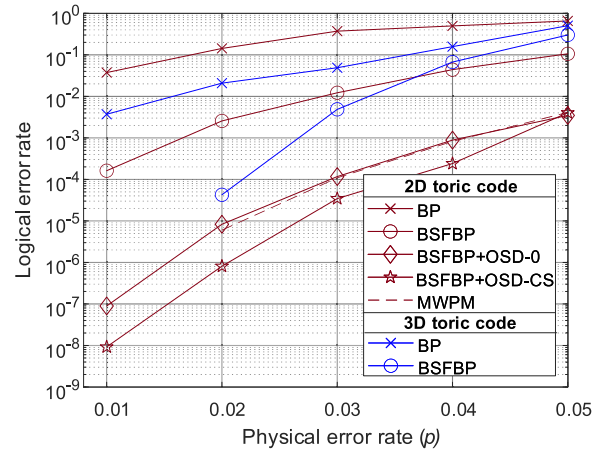


FIGURE 16. Performance of the 2-D toric code with distance 11 and the 3-D toric code with lattice size of 6. The maximum number of BP iterations for the 2-D and 3-D toric codes is  $N$  and 30, respectively.

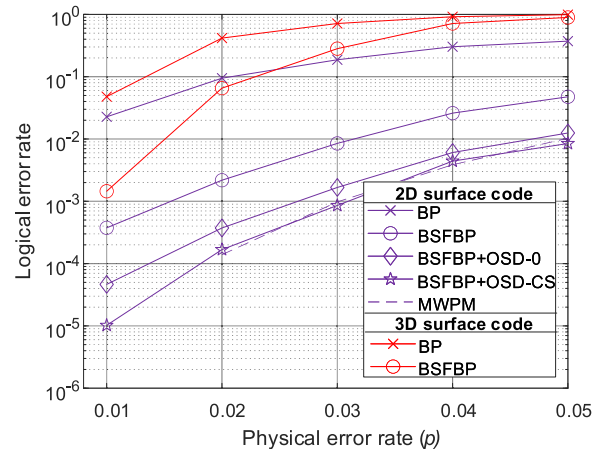
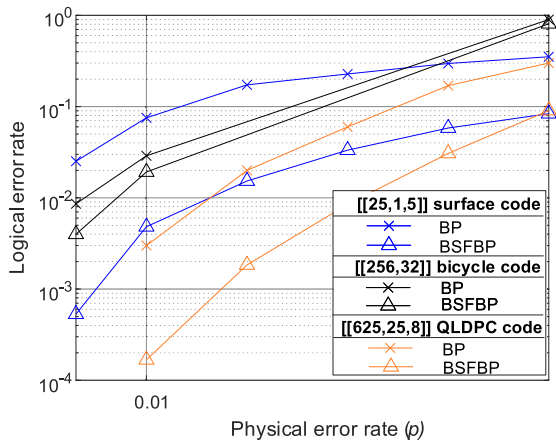


FIGURE 17. Performance of the 2-D surface code with distance 8 and 3-D surface codes with lattice size of 6. The maximum number of BP iterations for the 2-D and 3-D surface codes is  $N$  and 30, respectively.

when  $p = 0.01$  for the  $[[162,2,9]]$  toric code and  $10^{-8}$  for the  $[[242,2,11]]$  toric code.

Figs. 14 and 15 show the performance of the  $[[113,1,8]]$  and the  $[[181,1,10]]$  surface codes, respectively. By using the BSFBP decoding process, the performance improves by almost two orders of magnitude compared to the conventional BP decoding method. Since the BP+SSF decoding approach presented in [21] is devised for codes with a regular parity-check matrix, it is not considered for the surface code with the irregular parity-check matrix. In contrast, using BSFBP decoding combined with OSD-CS produces an error rate below  $10^{-5}$  and  $10^{-4}$  for the  $[[113,1,8]]$  and the  $[[181,1,10]]$  surface codes, which improves two and one orders of magnitude compared to the BP decoding concatenated with both OSD cases, respectively. Thus, this shows that BSFBP can be a better auxiliary to concatenate OSD. It also shows that our BSFBP decoding method itself can even outperform the



**FIGURE 18.** Performance comparison between BP and BSFBP for different codes. The maximum number of BP iterations is 12 for the  $[[256,32]]$  bicycle code and  $N$  for the  $[[25,1,5]]$  surface code and  $[[625,25,8]]$  QLDPC codes, respectively.

BP decoding concatenated with OSD, which indicates that it is possible to explore other modified BP decoders with lower complexity and better performance than OSD for the surface codes.

In Figs. 16 and 17, we show the performance for the toric code with a distance of 11 and the surface code with a distance of 8 when using the MWPM [31], respectively. The results show that by concatenating to OSD, it is possible to compete with the MWPM decoder for the considered 2-D topological codes. Although it has been shown that MWPM provides outstanding performance for topological codes, it requires codes that have a parity-check matrix containing column weights of, at most 2, which limits its universality [30], [31].

Consider the 3-D topological codes based on the homological product, which can be viewed as a generalization of the HGP construction [35]. These 3-D topological codes are popular for single-shot error correction, which means that the decoding process needs only measure the noisy syndrome in a single shot [45]. Both 3-D toric and surface codes are characterized by the  $X$ -stabilizer generators  $H_X$ ,  $Z$ -stabilizer generators  $H_Z$ , and a meta-check matrix  $M_c$ , which is used to correct syndrome noise. In this article, we will consider our decoder when applied to the 3-D toric and surface codes, both with a lattice size of 6 under the phase-flip channel, given the noiseless syndrome (without meta-checks). With a maximum number of iterations equal to 30, as in [47], Figs. 16 and 17 show that our BSFBP decoder can also operate on 3-D topological codes and even outperform BP by two orders of magnitude. For these codes, MWPM has restrictions on direct decoding since the column weights in  $H_X$  and  $H_Z$  are larger than 2.

## B. SIMULATION ON OTHER CODES

In this subsection, we will further explore the performance of other codes by using our proposed decoder in Fig. 18. First, the  $[[25,1,5]]$  surface code that is rotated by  $45^\circ$  is

considered [48]. This shows that our BSFBP is able to provide an improvement of about two orders of magnitude compared to BP. Then, we apply our decoder on  $[[256,32]]$  bicycle code with a maximum number of iterations equal to 12, as in [20]. The result shows that our decoder provides only a slight improvement. The reason for this is that the design of our BBP is based on the HGP structure; hence, only the SF process might provide an effect for the bicycle code. This can be indicated clearly by comparing it to the performance on the  $[[625,25,8]]$  QLDPC codes based on the HGP structure [25], which can improve performance by 1.5 orders of magnitude compared to BP.

## V. CONCLUSION

In this article, we proposed a BSFBP decoding method for topological codes based on the HGP structure. Our algorithm introduced the criteria and syndrome-pruning process to compute the specific syndrome residual. Then, a new decoding path using this syndrome residual as the input was then branched from BP. An SF process was also proposed on the original path of BP, which can provide more diverse syndrome residuals for decoding. For the considered topological codes, our BSFBP provided remarkable performance as a stand-alone decoder compared to BP. On the other hand, as an auxiliary decoder to OSD, we showed that it is possible to compete with the MWPM for the considered 2-D topological codes. We also showed that our BSFBP can be applied to those codes where the MWPM is limited, such as 3-D topological codes and other types of codes based on HGP structures.

Though for 2-D topological codes, it might be difficult to devise a BP-based decoder, which can compete with those state-of-the-art MWPM-based decoders such as in [28] and [29] based on similar performance and complexity, for single-shot codes such as 3-D toric and surface codes, the optimal decoder still needs to be explored in the future. Moreover, considering fault-tolerant computation, it would be interesting to compare the time overhead by applying our decoder on single-shot codes to hardware-efficient decoders such as those presented in [49], [50], and [51], when using multiple rounds of syndrome measurement on codes that have similar code parameters.

## REFERENCES

- [1] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, 1995, Art. no. R2493, doi: [10.1103/PhysRevA.52.R2493](https://doi.org/10.1103/PhysRevA.52.R2493).
- [2] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, pp. 1098–1105, 1996, doi: [10.1103/PhysRevA.54.1098](https://doi.org/10.1103/PhysRevA.54.1098).
- [3] A. M. Steane, "Error correcting codes in quantum theory," *Phys. Rev. Lett.*, vol. 77, pp. 793–797, 1996, doi: [10.1103/PhysRevLett.77.793](https://doi.org/10.1103/PhysRevLett.77.793).
- [4] M. S. Postol, "A proposed quantum low density parity check code," 2001, *arXiv:quant-ph/0108131*, doi: [10.48550/arXiv.quant-ph/0108131](https://doi.org/10.48550/arXiv.quant-ph/0108131).
- [5] A. Leverrier, J. P. Tillich, and G. Zémor, "Quantum expander codes," in *Proc. IEEE 56th Annu. Symp. Found. Comput. Sci.*, 2015, pp. 810–824, doi: [10.1109/FOCS.2015.55](https://doi.org/10.1109/FOCS.2015.55).

- [6] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Ann. Phys.*, vol. 303, pp. 2–30, 2003, doi: [10.1016/S0003-4916\(02\)00018-0](https://doi.org/10.1016/S0003-4916(02)00018-0).
- [7] S. B. Bravyi and A. Y. Kitaev, "Quantum codes on a lattice with boundary," 1998, *arXiv:quant-ph/9811052*, doi: [10.48550/arXiv.quant-ph/9811052](https://doi.org/10.48550/arXiv.quant-ph/9811052).
- [8] E. Dennis, A. Y. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *J. Math. Phys.*, vol. 43, pp. 4452–4505, 2002, doi: [10.1063/1.1499754](https://doi.org/10.1063/1.1499754).
- [9] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Phys. Rev. A*, vol. 86, 2012, Art. no. 032324, doi: [10.1103/PhysRevA.86.032324](https://doi.org/10.1103/PhysRevA.86.032324).
- [10] A. Córcoles et al., "Demonstration of a quantum error detection code using a square lattice of four superconducting qubits," *Nature Commun.*, vol. 6, 2015, Art. no. 6979, doi: [10.1038/ncomms7979](https://doi.org/10.1038/ncomms7979).
- [11] Google Quantum AI, "Exponential suppression of bit or phase errors with cyclic error correction," *Nature*, vol. 595, pp. 383–387, 2021, doi: [10.1038/s41586-021-03588-y](https://doi.org/10.1038/s41586-021-03588-y).
- [12] J.-P. Tillich and G. Zemor, "Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the block-length," *IEEE Trans. Inf. Theory*, vol. 60, pp. 1193–1202, Feb. 2014, doi: [10.1109/TIT.2013.2292061](https://doi.org/10.1109/TIT.2013.2292061).
- [13] D. Chandra et al., "Quantum topological error correction codes: The classical-to-quantum isomorphism perspective," *IEEE Access*, vol. 6, pp. 13729–13757, 2018, doi: [10.1109/ACCESS.2017.2784417](https://doi.org/10.1109/ACCESS.2017.2784417).
- [14] D. MacKay, G. Mitchison, and P. McFadden, "Sparse-graph codes for quantum error correction," *IEEE Trans. Inf. Theory*, vol. 50, pp. 2315–2330, Oct. 2004, doi: [10.1109/TIT.2004.834737](https://doi.org/10.1109/TIT.2004.834737).
- [15] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Slooan, "Quantum error correction via codes over GF(4)," *IEEE Trans. Inf. Theory*, vol. 44, no. 4, pp. 1369–1387, Jul. 1998, doi: [10.1109/18.681315](https://doi.org/10.1109/18.681315).
- [16] M. Davey and D. MacKay, "Low-density parity check codes over GF(q)," *IEEE Commun. Lett.*, vol. 2, pp. 165–167, Jun. 1998, doi: [10.1109/4234.681360](https://doi.org/10.1109/4234.681360).
- [17] D. Poulin and Y. Chung, "On the iterative decoding of sparse quantum codes," *Quantum Inf. Comput.*, vol. 8, pp. 987–1000, 2008, doi: [10.26421/QIC8.10-8](https://doi.org/10.26421/QIC8.10-8).
- [18] Y. J. Wang, B. C. Sanders, B. M. Bai, and X. M. Wang, "Enhanced feedback iterative decoding of sparse quantum codes," *IEEE Trans. Inf. Theory*, vol. 58, pp. 1231–1241, 2012, doi: [10.1109/TIT.2011.2169534](https://doi.org/10.1109/TIT.2011.2169534).
- [19] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "Fifteen years of quantum LDPC coding and improved decoding strategies," *IEEE Access*, vol. 3, pp. 2492–2519, 2015, doi: [10.1109/ACCESS.2015.2503267](https://doi.org/10.1109/ACCESS.2015.2503267).
- [20] K. Y. Kuo and C. Y. Lai, "Refined belief propagation decoding of sparse-graph quantum codes," *IEEE J. Sel. Area. Inf. Theory*, vol. 1, no. 2, pp. 487–498, Aug. 2020, doi: [10.1109/JSAIT.2020.3011758](https://doi.org/10.1109/JSAIT.2020.3011758).
- [21] A. Grospellier, L. Grou'ès, A. Krishna, and A. Leverrier, "Combining hard and soft decoders for hypergraph product codes," *Quantum*, vol. 5, 2021, Art. no. 432, doi: [10.22331/q-2021-04-15-432](https://doi.org/10.22331/q-2021-04-15-432).
- [22] O. Fawzi, A. Grospellier, and A. Leverrier, "Efficient decoding of random errors for quantum expander codes," in *Proc. 50th Annu. ACM SIGACT Symp. Theory Comput.*, 2018, pp. 521–534, doi: [10.1145/3188745.3188886](https://doi.org/10.1145/3188745.3188886).
- [23] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inf. Theory*, vol. 41, pp. 1379–1396, Sep. 1995, doi: [10.1109/18.412683](https://doi.org/10.1109/18.412683).
- [24] P. Panteleev and G. Kalachev, "Degenerate quantum LDPC codes with good finite length performance," *Quantum*, vol. 5, 2021, Art. no. 585, doi: [10.22331/q-2021-11-22-585](https://doi.org/10.22331/q-2021-11-22-585).
- [25] J. Roffe, D. R. White, S. Burton, and E. Campbell, "Decoding across the quantum low-density parity-check code landscape," *Phys. Rev. Res.*, vol. 2, 2020, Art. no. 043423, doi: [10.1103/PhysRevResearch.2.043423](https://doi.org/10.1103/PhysRevResearch.2.043423).
- [26] J. Valls, F. Garcia-Herrero, N. Raveendran, and B. Vasic, "Syndrome-based min-sum vs OSD-0 decoders: FPGA implementation and analysis for quantum LDPC codes," *IEEE Access*, vol. 9, pp. 138734–138743, 2021, doi: [10.1109/ACCESS.2021.3118544](https://doi.org/10.1109/ACCESS.2021.3118544).
- [27] J. Edmonds, "Paths, trees, and flowers," *Can. J. Math.*, vol. 17, pp. 449–467, 1965, doi: [10.4153/CJM-1965-045-4](https://doi.org/10.4153/CJM-1965-045-4), doi: [10.48550/arXiv.2101.04125](https://doi.org/10.48550/arXiv.2101.04125).
- [28] N. Delfosse and N. H. Nickerson, "Almost-linear time decoding algorithm for topological codes," *Quantum*, vol. 5, 2021, Art. no. 595, doi: [10.22331/q-2021-12-02-595](https://doi.org/10.22331/q-2021-12-02-595).
- [29] C. T. Chubb, "General tensor network decoding of 2D Pauli codes," 2021, *arXiv:2101.04125*, doi: [10.48550/arXiv.2101.04125](https://doi.org/10.48550/arXiv.2101.04125).
- [30] J. Roffe, L. Z. Cohen, A. O. Quintavalle, D. Chandra, and E. T. Campbell, "Bias-tailored quantum LDPC codes," *Quantum*, vol. 7, 2023, Art. no. 1005, doi: [10.22331/q-2023-05-15-1005](https://doi.org/10.22331/q-2023-05-15-1005).
- [31] O. Higgott, "PyMatching: A PYTHON package for decoding quantum codes with minimum-weight perfect matching," *ACM Trans. Quantum Comput.*, vol. 3, pp. 1–16, 2022, doi: [10.1145/3505637](https://doi.org/10.1145/3505637).
- [32] K. Y. Kuo and C. Y. Lai, "Exploiting degeneracy in belief propagation decoding of quantum codes," *NPJ Quantum Inf.*, vol. 8, 2022, Art. no. 111, doi: [10.1038/s41534-022-00623-2](https://doi.org/10.1038/s41534-022-00623-2).
- [33] E. Pelchat and D. Poulin, "Degenerate Viterbi decoding," *IEEE Trans. Inf. Theory*, vol. 59, no. 6, pp. 3915–3921, Jun. 2013, doi: [10.1109/TIT.2013.2246815](https://doi.org/10.1109/TIT.2013.2246815).
- [34] P. Fuentes, J. E. Martinez, P. M. Crespo, and J. Garcia-Frias, "Degeneracy and its impact on the decoding of sparse quantum codes," *IEEE Access*, vol. 9, pp. 89093–89119, 2021, doi: [10.1109/ACCESS.2021.3089829](https://doi.org/10.1109/ACCESS.2021.3089829).
- [35] S. Bravyi and M. B. Hastings, "Homological product codes," in *Proc. 46th Annu. ACM Symp. Theory Comput.*, 2014, pp. 273–282, doi: [10.1145/2591796.2591870](https://doi.org/10.1145/2591796.2591870).
- [36] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2000, doi: [10.1017/CBO9780511976667](https://doi.org/10.1017/CBO9780511976667).
- [37] D. Gottesman, "Stabilizer codes and quantum error correction," Ph.D. dissertation, California Inst. Technol., Pasadena, CA, USA, 1997.
- [38] R. Cleve, "Quantum stabilizer codes and classical linear codes," *Phys. Rev. A*, vol. 55, pp. 4054–4059, 1997, doi: [10.1103/PhysRevA.55.4054](https://doi.org/10.1103/PhysRevA.55.4054).
- [39] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, 1962, doi: [10.1109/TIT.1962.1057683](https://doi.org/10.1109/TIT.1962.1057683).
- [40] H. C. Lee, Y. L. Ueng, S. M. Yeh, and W. Y. Weng, "Two informed dynamic scheduling strategies for iterative LDPC decoders," *IEEE Trans. Commun.*, vol. 61, no. 3, pp. 886–896, Mar. 2013, doi: [10.1109/TCOMM.2013.012313.120172](https://doi.org/10.1109/TCOMM.2013.012313.120172).
- [41] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003. [Online]. Available: <https://www.cambridge.org/us/universitypress/subjects/computer-science/pattern-recognition-and-machine-learning/information-theory-inference-and-learning-algorithms?format=HB&isbn=9780521642989>
- [42] J. Chen and M. Fossorier, "Density evolution of two improved BP-based algorithms for LDPC decoding," *IEEE Commun. Lett.*, vol. 6, no. 5, pp. 208–210, May 2002, doi: [10.1109/4234.1001666](https://doi.org/10.1109/4234.1001666).
- [43] C. C. Cheng, J. D. Yang, H. C. Lee, C. H. Yang, and Y. L. Ueng, "A fully parallel LDPC decoder architecture using probabilistic min-sum algorithm for high-throughput applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 9, pp. 2738–2746, Sep. 2014, doi: [10.1109/TCSI.2014.2312479](https://doi.org/10.1109/TCSI.2014.2312479).
- [44] A. A. Emran and M. Elsbabrouy, "Simplified variable-scaled min-sum LDPC decoder for irregular LDPC codes," in *Proc. IEEE Consum. Commun. Netw. Conf.*, 2014, pp. 518–523, doi: [10.1109/CCNC.2014.6940497](https://doi.org/10.1109/CCNC.2014.6940497).
- [45] A. O. Quintavalle, M. Vasmer, J. Roffe, and E. T. Campbell, "Single-shot error correction of three-dimensional homological product codes," *PRX Quantum*, vol. 2, no. 2, 2021, Art. no. 020340, doi: [10.1103/PRXQuantum.2.020340](https://doi.org/10.1103/PRXQuantum.2.020340).
- [46] A. Grospellier and A. Krishna, "Numerical study of hypergraph product codes," 2018, *arXiv:1810.03681*, doi: [10.48550/arXiv.1810.03681](https://doi.org/10.48550/arXiv.1810.03681).
- [47] O. Higgott and N. P. Breuckmann, "Improved single-shot decoding of higher dimensional hypergraph product codes," *PRX Quantum*, vol. 4, 2023, Art. no. 020332, doi: [10.1103/PRXQuantum.4.020332](https://doi.org/10.1103/PRXQuantum.4.020332).
- [48] C. Horsman, A. G. Fowler, S. Devitt, and R. Van Meter, "Surface code quantum computing by lattice surgery," *New J. Phys.*, vol. 14, 2012, Art. no. 123011, doi: [10.1088/1367-2630/14/12/123011](https://doi.org/10.1088/1367-2630/14/12/123011).
- [49] P. Das et al., "A scalable decoder micro-architecture for fault-tolerant quantum computing," 2020, *arXiv:2001.06598*, doi: [10.48550/arXiv.2001.06598](https://doi.org/10.48550/arXiv.2001.06598).
- [50] A. Holmes, M. R. Jorak, G. Pasandi, Y. Ding, M. Pedram, and F. T. Chong, "NISQ+: Boosting quantum computing power by approximating quantum error correction," in *Proc. ACM/IEEE 47th Annu. Int. Symp. Comput. Archit.*, 2020, pp. 556–569, doi: [10.1109/isca45697.2020.00053](https://doi.org/10.1109/isca45697.2020.00053).
- [51] Y. Ueno, M. Kondo, M. Tanaka, Y. Suzuki, and Y. Tabuchi, "QE-COOL: On-line quantum error correction with a superconducting decoder for surface code," in *Proc. ACM/IEEE 58th Des. Autom. Conf.*, 2021, pp. 451–456, doi: [10.1109/DAC18074.2021.9586326](https://doi.org/10.1109/DAC18074.2021.9586326).



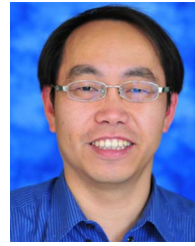
**Tzu-Hsuan Huang** received the B.S. degree in physics from National Yang Ming Chiao Tung University, Hsinchu, Taiwan, in 2015. He is currently working toward the Ph.D. degree in error correction codes in the Department of Electrical Engineering, National Tsing Hua University, Hsinchu.

His research interests include channel coding and quantum error correction codes.



**Ting-An Hu** received the B.S. degree in electrical engineering from Yuan Ze University, Jungli, Taiwan, in 2019. He is currently working toward the master's degree in the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan.

His research interests include channel coding and very large scale integration designs.



**Yeong-Luh Ueng** (Senior Member, IEEE) received the Ph.D. degree in communication engineering from National Taiwan University, Taipei, Taiwan, in 2001.

From 2001 to 2005, he was with a private communication technology company, where he focused on the design and development of various wireless chips. Since December 2005, he has been a Member of the faculty with National Tsing Hua University, Hsinchu, Taiwan, where he is currently a Full Professor with the Department of Electrical Engineering and the Institute of Communications Engineering. His research interests include coding theory, wireless communications, and communication integrated circuits.

Dr. Ueng was the recipient of the Wu Ta-You Memorial Award from the Ministry of Science and Technology (MOST) in 2016. In 2018, he was the recipient of the Outstanding Electrical Engineering Professor Award from the Chinese Electrical Engineering Association, as well as the Outstanding Research Award from MOST, Taiwan.