

# Low-Complexity Linear Programming Based Decoding of Quantum LDPC codes

Sana Javed\*, Francisco Garcia-Herrero†, Bane Vasić ‡ and Mark F. Flanagan\*

\* School of Electrical and Electronic Engineering, University College Dublin, Belfield, Dublin 4, Ireland,

† Department of Computer Architecture and Automatics, Complutense University of Madrid, Madrid, Spain

‡ Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ, 85721 USA

Email: sana.javed@ucdconnect.ie, francg18@ucm.es, vasic@ece.arizona.edu, mark.flanagan@ieee.org

**Abstract**—This paper proposes two approaches for reducing the impact of the error floor phenomenon when decoding quantum low-density parity-check codes with belief propagation based algorithms. First, a low-complexity syndrome-based linear programming (SB-LP) decoding algorithm is proposed, and second, the proposed SB-LP is applied as a post-processing step after syndrome-based min-sum (SB-MS) decoding. For the latter case, a new early stopping criterion is introduced to decide when to activate the SB-LP algorithm, avoiding executing a predefined maximum number of iterations for the SB-MS decoder. Simulation results show, for a sample hypergraph code, that the proposed decoder can lower the error floor by two to three orders of magnitude compared to SB-MS for the same total number of decoding iterations.

**Keywords**—Quantum error correction, quantum LDPC codes, linear programming based decoding.

## I. INTRODUCTION

Quantum algorithms that involve a large number of qubits (between thousands and millions) play a pivotal role in advancing quantum computing, unlocking the ability to tackle intricate problems that are beyond the reach of classical computers [1]. However, noise in quantum systems, generated by factors such as decoherence, crosstalk, and environmental interference, represents a significant hurdle [2], making it challenging to maintain the integrity of quantum computation, especially when the number of qubits scales [1]– [3].

To reduce the effects of noise, many new codes and decoders for quantum error correction have been designed over the past few decades [2]. Some of the most intensively studied codes are surface codes [4], initially suggested for noisy intermediate-scale quantum (NISQ) devices. Unfortunately, these codes have some limitations when the objective is to protect a large number of logical qubits, as they have

The work of S. Javed and M. F. Flanagan was supported by Science Foundation Ireland under the US-Ireland R&D Partnership Programme (CoQCREATE, Grant Number SFI/21/US-C2C/3750). The work of F. Garcia-Herrero was supported by the QuantERA grant EQUIP (Spain MCIN/AEI/10.13039/501100011033, grant PCI2022-132922), funded by Agencia Estatal de Investigación, Ministerio de Ciencia e Innovación, Gobierno de España and by the European Union “NextGenerationEU/PRTR”. Bane Vasić acknowledges the support of the NSF under grants CIF-1855879, CCF-2100013, CIF-2106189, CCSS-2027844, CCSS-2052751, CIF-2106189, and ERC-1941583 (CoQCREATE), as well as the support of NASA through the SURP Program. Bane Vasić has disclosed an outside interest in Codelucida to the University of Arizona. Conflicts of interest resulting from this interest are being managed by The University of Arizona in accordance with its policies.

a low code rate and consequently an impractical overhead in terms of physical qubits.

As an alternative with a higher code rate, quantum low-density parity-check (QLDPC) codes offer benefits by reducing the qubit overhead and also improving the minimum distance [5], [6]. Despite their potential, there are still open problems such as the fault-tolerant implementation of the encoding process on the quantum side and the design of efficient decoders that can obtain very low logical error rates [7]. This paper will focus on addressing the latter limitation.

One of the main problems in performing efficient decoding for a QLDPC code is the existence of short cycles within the code’s Tanner graph, which result in an error floor when belief propagation (BP) based decoders are applied. This error floor effect means that an improvement in the physical error rate of the qubits does not translate into an improvement in the logical error rate [8]. To mitigate this problem, two main categories of approach have been followed: the use of a new standalone decoder based on alternative approaches to standard BP, and applying a post-processing technique after BP decoding.

For the first category of approach, some more powerful decoding techniques such as the syndrome-based generalized belief propagation (GBP) algorithm [9] have been proposed to improve the decoding performance. Unfortunately, these require a high computational complexity, so additional simplifications are necessary to obtain practical real-time solutions. Some neural network based decoders for QLDPC codes have also been explored [10]. However, training and deploying these algorithms for large QLDPC codes can be computationally intensive and resource-demanding when the number of qubits grows.

For the second category, ordered statistics decoding (OSD) was proposed in [11] as a post-processing technique for BP. By seamlessly integrating OSD when BP decoding fails, this technique can significantly enhance the error correction capability for some specific QLDPC codes, providing improved error rates after post-processing. However, OSD decoders are computationally complex and introduce an extremely large latency due to the requirement, in one of the steps, of performing Gaussian elimination for the inversion of a dynamic submatrix of the code’s parity-check matrix; thus, an efficient hardware implementation that can finish decoding

within the required time interval (of hundreds of nanoseconds to several microseconds) is not feasible [12].

Stabilizer inactivation decoding for QLDPC codes was also recently introduced in [13] as an alternative post-processing method; this was shown to outperform OSD in lowering the error floor while also reducing the decoding complexity. However, it is a list decoding method (the list being computed according to the BP output), the complexity of which grows with the length of the list. Serial processing of the list can lead to information loss due to the short decoherence time, while parallel processing would require high power consumption. This highlights the importance of continuing the search for alternative decoding techniques.

Besides these previous algorithms, linear programming (LP) based decoding has also been recently explored as a solution for quantum codes. As LP decoding proved to be capable of reducing the error floor for classical LDPC codes [14], some works such as [15] and [16] adapted classical LP decoding algorithms to QLDPC codes, also providing theoretical performance guarantees. Nevertheless, these decoders require the solution of linear programs and thus these algorithms are not attractive from a complexity and latency perspective.

In this paper, a low-complexity iterative syndrome-based linear programming (SB-LP) decoder is proposed for decoding QLDPC codes, which can be used as a standalone decoding algorithm or as a post-processing step after syndrome-based min-sum (SB-MS) decoding to improve the behavior in the error floor region and the threshold of the code. Unlike the standalone SB-MS decoder, which exhibits a high error floor for several QLDPC code classes, the proposed combination of SB-MS with SB-LP improves the error floor with lower complexity than OSD and without the need for list decoding as required by stabilizer inactivation. We also propose a new early stopping criterion for the SB-MS decoder based on the syndrome Hamming distance, using which the algorithm can automatically detect misconvergence of the SB-MS decoder and use this to trigger the SB-LP post-processing.

## II. BACKGROUND

This section summarizes the depolarizing error model used in our work, and provides a brief description of the SB-MS decoder which is a *de facto* standard for low-complexity BP based decoding. In the following, all vectors are assumed to be row vectors.

### A. Quantum Noise Model

Quantum gates, crosstalk, and other processes applied to qubits (including maintaining them in an idle state) result in bit-flip errors, phase-flip errors, or a combination of both. The behavior of errors in a quantum processor can be described by using a depolarizing noise model [17]. Errors resulting from bit-flips are described by Pauli X operators and have a probability of occurrence denoted by  $p_X$ . Phase-flip errors are described by Pauli Z operators and have

a probability of occurrence  $p_Z$ . Errors involving both bit-flips and phase-flips are described by Pauli Y operators and occur with probability  $p_Y$ . We consider an i.i.d. symmetric depolarizing error model for the qubits, such that each qubit experiences an independent depolarizing error probability  $p$  and  $p_X = p_Y = p_Z = p/3$ . Bit-flip and phase-flip errors are represented by binary error vectors  $\mathbf{e}_X$  and  $\mathbf{e}_Z$ , respectively, in which the  $j$ -th entry is equal to 1 in the presence of a corresponding error on the  $j$ -th qubit, and is equal to 0 if there is no such error. An  $[[n, k]]$  QLDPC code, where  $k$  logical qubits are encoded using  $n$  physical qubits, can be characterized by two  $m \times n$  binary parity-check matrices  $\mathbf{H}_X$  and  $\mathbf{H}_Z$ ; the corresponding length- $m$  binary syndrome vectors  $\mathbf{s}_X = \mathbf{e}_X \mathbf{H}_Z^T$  and  $\mathbf{s}_Z = \mathbf{e}_Z \mathbf{H}_X^T$  are measured from the quantum system and can be used to correct X and Z errors, respectively [5]. For ease of exposition, in the rest of this paper we will focus on correction of only one type of error, and we will use the simpler notations  $\mathbf{H}$ ,  $\mathbf{e}$  and  $\mathbf{s}$  in place of  $\mathbf{H}_Z$ ,  $\mathbf{e}_X$  and  $\mathbf{s}_X$ , respectively. Also, we will assume the syndrome measurement to be error-free.

### B. Tanner Graph

The Tanner graph serves as a graphical representation of the  $m \times n$  parity-check matrix  $\mathbf{H} = (H_{ij})$ . This graph consists of  $n$  variable nodes (VNs) and  $m$  check nodes (CNs), connected by a set of edges  $\mathcal{E}$ . For each  $i \in \{1, 2, \dots, m\}$  and  $j \in \{1, 2, \dots, n\}$ , the  $i$ -th CN is connected to the  $j$ -th VN by an edge (i.e.,  $(i, j) \in \mathcal{E}$ ) if and only if  $H_{ij} = 1$ .  $\mathcal{N}_i$  denotes the set of neighbors of the  $i$ -th CN, while  $\mathcal{N}_j$  denotes the set of neighbors of the  $j$ -th VN. For ease of exposition, we assume that the Tanner graph is *regular*, i.e., all CNs have the same degree  $d_c$  and all VNs have the same degree  $d_v$ . In the QLDPC decoding context, VNs correspond to entries of the estimated error pattern  $\mathbf{e} = (e_1 \ e_2 \ \dots \ e_n)$  while CNs correspond to the entries of the syndrome vector  $\mathbf{s} = (s_1 \ s_2 \ \dots \ s_m)$ .

### C. Syndrome-Based Min-Sum (SB-MS) Algorithm

The SB-MS algorithm is a low-complexity approximation to BP which replaces the complex CN operation with a simpler minimum operation [18]. Its operation is given in Algorithm 1.

The SB-MS decoder seeks to find an error pattern  $\hat{\mathbf{e}}$  that matches the measured syndrome  $\mathbf{s}$ . It does this by iteratively exchanging messages between the VNs and CNs (lines 4 to 7), making hard decisions to produce an estimate of the error vector  $\hat{\mathbf{e}}$  (line 8), and generating the corresponding syndrome  $\hat{\mathbf{s}}$  (line 9). The algorithm terminates when either this syndrome matches the measured syndrome or the maximum number of iterations  $I_{\max}^{(\text{MS})}$  is reached.

Here  $u_{i,j}$  represents the message from the  $i$ -th CN to the  $j$ -th VN, and  $v_{i,j}$  denotes the message sent from the  $j$ -th VN to the  $i$ -th CN. The hard decision operation is defined as

$$\text{HD}(x) = \begin{cases} 0 & \text{if } x > 0, \\ 1 & \text{otherwise.} \end{cases}$$

and we also define the function

$$\text{sgn}(x) = 1 - 2\text{HD}(x) = \begin{cases} 1 & \text{if } x > 0, \\ -1 & \text{otherwise.} \end{cases}$$

$\lambda_j = \log\left(\frac{P(e_j=0)}{P(e_j=1)}\right)$  represents the *a priori* log-likelihood ratio (LLR) of  $e_j$ , while  $\alpha$  is a scaling factor that accelerates the convergence of the algorithm.

---

**Algorithm 1:** Syndrome-Based Min-Sum (SB-MS) Algorithm

---

1: Input: syndrome  $\mathbf{s}$

2: Initialize parameters:

$$\lambda_j = \ln\left(\frac{1-(2p/3)}{2p/3}\right), \forall j \in \{1, 2, \dots, n\}$$

$$u_{i,j} = 0, \forall (i, j) \in \mathcal{E}$$

$$I = 0, \hat{\mathbf{s}} = \mathbf{0}$$

3: **while**  $\mathbf{s} \neq \hat{\mathbf{s}}$  and  $I \leq I_{\max}^{(\text{MS})}$  **do**

4:   **for** all  $(i, j) \in \mathcal{E}$  (in parallel) **do**

$$v_{i,j} = \lambda_j + \alpha \sum_{i' \in \mathcal{N}_j \setminus \{i\}} u_{i',j}$$

5:   **end for**

6:   **for** all  $(i, j) \in \mathcal{E}$  (in parallel) **do**

$$u_{i,j} = \text{sgn}(s_i) \cdot \prod_{j' \in \mathcal{N}_i \setminus \{j\}} \text{sgn}(v_{i,j'}) \cdot \min_{j' \in \mathcal{N}_i \setminus \{j\}} |v_{i,j'}|$$

7:   **end for**

$$8: \hat{e}_j = \text{HD}(\lambda_j + \alpha \sum_{i \in \mathcal{N}_j} u_{i,j}), \forall j \in \{1, 2, \dots, n\}$$

9:

$$\hat{\mathbf{s}} = \hat{\mathbf{e}} \mathbf{H}^T$$

$$10: I = I + 1$$

11: **end while**

12: **return**  $\hat{\mathbf{e}}$

---

In the case of certain QLDPC codes, the SB-MS decoder is unable to improve the error rate of the qubits even as the depolarizing error probability improves; this results in an error floor. The presence of this error floor can be attributed to the presence of short cycles within the code's Tanner graph which the SB-MS decoding algorithm struggles to handle. These cycles lead to oscillations between different candidate error vectors during the decoding process so that the SB-MS decoder cannot converge to a unique error pattern that matches the syndrome. In the next section, we propose an approach to reduce or eliminate this error floor phenomenon.

### III. PROPOSED METHOD

Linear programming based decoding, originally proposed for classical LDPC codes in [14], relies on a linear programming relaxation approach to achieve a high-performance decoding solution with theoretical guarantees on performance. In [19], Vontobel and Koetter proposed low-complexity iterative decoding algorithms that exploited the unique structure of the linear program to be solved (in particular, exploiting the sparsity of  $\mathbf{H}$ ). In the following, we propose an LP-based

decoder which is based on modifying Algorithm 1 in [19] to align with the unique features and requirements of QLDPC codes.

**A. Proposed Syndrome-based Linear Programming (SB-LP) Decoder**

The details of the proposed SB-LP decoder are given in Algorithm 2. This algorithm iteratively updates the estimated error vector  $\hat{\mathbf{e}}$ , guided by the measured syndrome  $\mathbf{s}$ , until convergence is achieved or the maximum number of iterations  $I_{\max}^{(\text{LP})}$  has been reached. Unlike the SB-MS decoder, the SB-LP decoder does not perform VN-to-CN and CN-to-VN updates alternately; instead, it simultaneously updates a single value  $\bar{u}_{i,j}$  for each edge  $(i, j) \in \mathcal{E}$  (line 9). This parallel update approach can significantly improve the decoding speed. Also,  $\alpha_1$  serves as a scaling factor designed to expedite the algorithm's convergence.

---

**Algorithm 2:** Proposed Syndrome-Based Linear Programming (SB-LP) Algorithm

---

1: Input: syndrome  $\mathbf{s}$

2: Initialize parameters:

$$\lambda_j = \ln\left(\frac{1-(2p/3)}{2p/3}\right), \forall j \in \{1, 2, \dots, n\}$$

$$\bar{u}_{i,j} = 0, \forall (i, j) \in \mathcal{E}$$

$$I = 0, \hat{\mathbf{s}} = \mathbf{0}$$

3: **while**  $\mathbf{s} \neq \hat{\mathbf{s}}$  and  $I \leq I_{\max}^{(\text{LP})}$  **do**

4:   **for** all  $(i, j) \in \mathcal{E}$  (in parallel) **do**

5:

$$S_{i,j} = \lambda_j + \sum_{i' \in \mathcal{N}_j \setminus \{i\}} \bar{u}_{i',j}$$

6:   **if**  $s_i = 0$  **then**

$$T_{i,j}^{(0)} = \max_{\mathbf{b} \in \mathcal{B}_i, b_j=0} \bar{\mathbf{u}}_{i,j} \tilde{\mathbf{b}}_j^T$$

$$T_{i,j}^{(1)} = \max_{\mathbf{b} \in \mathcal{B}_i, b_j=1} \bar{\mathbf{u}}_{i,j} \tilde{\mathbf{b}}_j^T$$

7:   **else**

$$T_{i,j}^{(0)} = \max_{\mathbf{c} \in \mathcal{C}_i, c_j=0} \bar{\mathbf{u}}_{i,j} \tilde{\mathbf{c}}_j^T$$

$$T_{i,j}^{(1)} = \max_{\mathbf{c} \in \mathcal{C}_i, c_j=1} \bar{\mathbf{u}}_{i,j} \tilde{\mathbf{c}}_j^T$$

8:   **end if**

$$9: \bar{u}_{i,j} = \frac{\alpha_1}{2} (T_{i,j}^{(0)} - T_{i,j}^{(1)} - S_{i,j})$$

10: **end for**

$$11: \hat{e}_j = \text{HD}(\lambda_j + \sum_{i' \in \mathcal{N}_j} \bar{u}_{i',j}), \forall j \in \{1, 2, \dots, n\}$$

12:

$$\hat{\mathbf{s}} = \hat{\mathbf{e}} \mathbf{H}^T$$

$$13: I = I + 1$$

14: **end while**

15: **return**  $\hat{\mathbf{e}}$

---

In lines 6 and 7,  $\mathcal{B}_i$  denotes the single parity-check code of length  $d_c$  consisting of all  $2^{|d_c|-1}$  binary sequences having even parity, while  $\mathcal{C}_i = \{0, 1\}^{|d_c|} \setminus \mathcal{B}_i$  represents the set of

all  $2^{|d_c|-1}$  binary sequences having odd parity (note that these vectors are indexed by the set  $\mathcal{N}_i$ ). Also,  $\tilde{\mathbf{b}}_j$  and  $\tilde{\mathbf{c}}_j$  correspond to the vectors  $\mathbf{b}$  and  $\mathbf{c}$  with the  $j$ -th element removed, respectively. Also, we define the check-perspective edge value vector  $\bar{\mathbf{u}}_{i,j} = (\bar{u}_{i,j'})_{j' \in \mathcal{N}_i \setminus \{j\}}$ .

With respect to Algorithm 1 in [19], which was proposed for decoding classical LDPC codes, we have introduced some modifications that enable the algorithm to decode QLDPC codes. First, we have modified the expressions for  $T_{i,j}^{(0)}$  and  $T_{i,j}^{(1)}$  in lines 6 and 7, as they depend on the  $i$ -th syndrome bit  $s_i$ ; in classical coding, we always have  $s_i = 0$ , while in QLDPC coding we may have  $s_i = 1$  which necessitates performing the corresponding minimization over the complement  $\mathcal{C}_i$  of the single parity check code  $\mathcal{B}_i$ . Second, we have replaced the “soft minimum” operator in [19] with the minimum operator in order to reduce the complexity (this was found to give negligible loss in performance). Finally, we have replaced the successive edge value update rule in [19] by a fully parallel update rule, and we have made some other minor simplifications including removing the (unnecessary) dual variables of the LP.

### B. SB-LP as Post-Processing after SB-MS Decoding

We can further improve the error floor behavior while reducing the overall decoding complexity by combining the SB-MS and SB-LP decoders. Algorithm 3 describes our proposed methodology for this. First, we note that the SB-MS decoder exhibits an oscillatory behavior when it becomes stuck in a trapping set of the QLDPC code [8]. In such cases, the decoder is not able to identify an error pattern that matches the measured syndrome, and hence the number of unmatched syndrome elements begins to oscillate as it seeks to converge to more than one error vector within the trapping set. If this behavior can be identified automatically by the decoding algorithm (through an appropriately designed early stopping criterion, as described in Subsection III-C), the SB-LP decoder can then be invoked to complete the decoding successfully (as the SB-MS decoder alone will usually not be capable of converging to a solution in such cases).

Moreover, to facilitate faster convergence of the iterative SB-LP decoder, we take advantage of the calculations provided by the SB-MS decoder in the first stage. Specifically, we use the *a posteriori* LLR values obtained after the SB-MS decoder satisfies the proposed stopping criterion (these are computed in line 8 of Algorithm 3 by adding the CN-to-VN and VN-to-CN messages) as initial values for the edge variables  $\bar{u}_{i,j}$  of the SB-LP decoder. This provides an improved initialization of the post-processing decoder.

The SB-MS decoder, which has a faster convergence than SB-LP in general, allows the decoding process to reach a good starting point, reducing considerably the number of errors; if it encounters issues during this process (as detected by the early stopping criterion), the SB-LP decoder is then invoked to remove the small set of residual errors, which are linked to trapping sets.

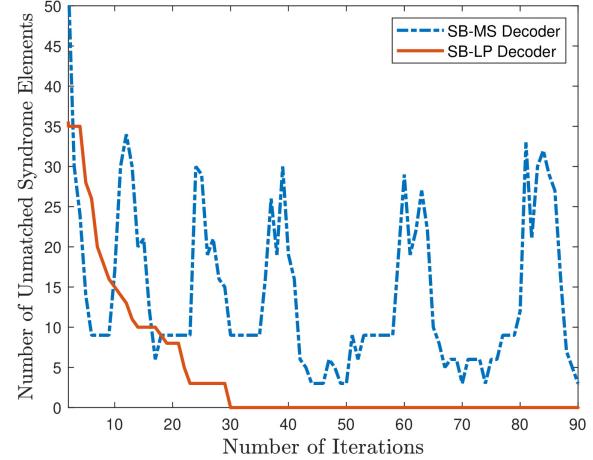


Fig. 1. Evolution of the number of unmatched syndrome elements with successive decoding iterations for a sample error vector. Plots are shown for the SB-MS (blue) and SB-LP (red) decoders for the [[882, 24]] QLDPC code B1 from [11].

---

### Algorithm 3: Combined SB-MS + SB-LP Decoding Algorithm

---

- 1: Input: syndrome  $\mathbf{s}$
- 2: Initialize parameters:  
 $\lambda_j = \ln\left(\frac{1-(2p/3)}{2p/3}\right)$ ,  $\forall j \in \{1, 2, \dots, n\}$   
 $u_{i,j} = 0, \forall (i, j) \in \mathcal{E}$   
 $I = 0, \hat{\mathbf{s}} = \mathbf{0}$
- 3: **do**
- 4:      $\hat{\mathbf{s}}_{\text{pre}} = \hat{\mathbf{s}}$
- 5:     Execute lines (4-11) of Algorithm 1
- 6:      $I = I + 1$
- 7:     **while**  $d_H(\hat{\mathbf{s}}_{\text{pre}}, \hat{\mathbf{s}}) > d_v$  and  $I \leq I_{\max}^{(\text{MS})}$
- 8:      $\bar{u}_{i,j} = u_{i,j} + v_{i,j}, \forall (i, j) \in \mathcal{E}$
- 9:      $I = 0$
- 10: Execute lines (3-15) of Algorithm 2

---

### C. Early Stopping Criterion for the SB-MS Decoder

As explained in the previous subsection, an appropriately designed early stopping criterion can reduce the number of decoding iterations by exiting the SB-MS decoder when it becomes trapped and at the same time provide an improved initialization to the SB-LP decoder. Fig. 1 shows an example of the evolution of the number of unmatched syndrome elements during SB-MS decoding in the case where the decoder becomes stuck in a trapping set. An oscillatory pattern is observed, indicating that the SB-MS decoder is unable to converge to an error pattern that matches the measured syndrome. In contrast, the proposed SB-LP decoder can converge in such cases; however, a high number of iterations is required.

From Fig. 1 we can conclude that operating the SB-MS decoder for a fixed number of iterations is not an efficient

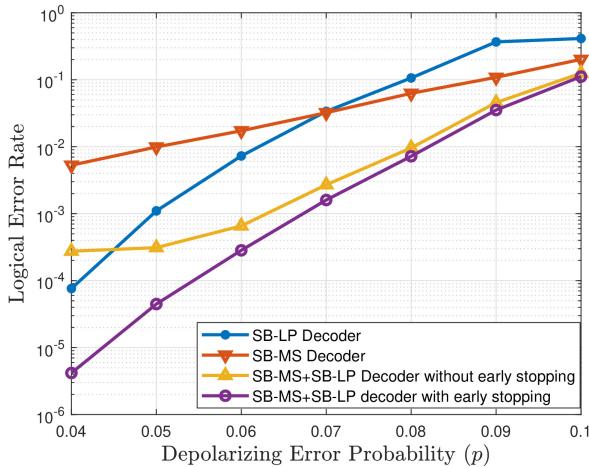


Fig. 2. Performance of the SB-MS decoder, the SB-LP decoder, and the combined SB-MS and SB-LP decoder with and without early stopping criterion, for the [[882, 24]] QLDPC code B1 from [11].

approach; instead, the SB-MS decoding procedure should terminate when such oscillatory behavior is detected. In Algorithm 3, we terminate the SB-MS decoding when the Hamming distance  $d_H(\hat{\mathbf{s}}, \hat{\mathbf{s}}_{\text{pre}})$  between the current syndrome estimate  $\hat{\mathbf{s}}$  and the previous syndrome estimate  $\hat{\mathbf{s}}_{\text{pre}}$  is less than or equal to the VN degree  $d_v$ ; this approach was found to be more robust than attempting to locate the first local minimum of the number of unmatched syndrome elements. By providing the SB-LP decoder with an improved initialization, this early stopping criterion leads to fewer required decoding iterations while also improving the logical error rate.

It is worth mentioning that this early stopping criterion can be applied when using SB-MS in conjunction with any other post-processing technique, such as OSD or stabilizer inactivation, to activate the post-processing exactly when required and reduce the total number of decoding iterations.

#### IV. SIMULATION RESULTS AND DISCUSSION

In this section, we compare the simulated performance of the proposed SB-LP decoder with that of SB-MS, and we also evaluate the performance of the combined SB-MS and SB-LP decoders. Simulations are performed for the [[882, 24]] QLDPC code designated as code B1 in [11] (similar results were found for the [[1922, 50]] QLDPC code C2 in [11]; these are omitted due to space limitations). The correction of bit-flip (Pauli X) errors is considered under the i.i.d. symmetric depolarizing noise model.

Fig. 2 shows the performance curves for the different decoders, with a maximum number of iterations  $I_{\text{max}}^{(\text{MS})} = I_{\text{max}}^{(\text{LP})} = 100$  for the SB-MS and SB-LP algorithms respectively; the corresponding scaling factors are  $\alpha = 0.75$  and  $\alpha_1 = 0.9$ . For the case where the SB-MS and SB-LP decoders are combined (Algorithm 3), the first decoder has  $I_{\text{max}}^{(\text{MS})} = 25$  iterations while the second decoder is configured with  $I_{\text{max}}^{(\text{LP})} = 75$ . For each point in the figure, 10,000 logical errors were simulated in order to ensure that the results are statistically significant.

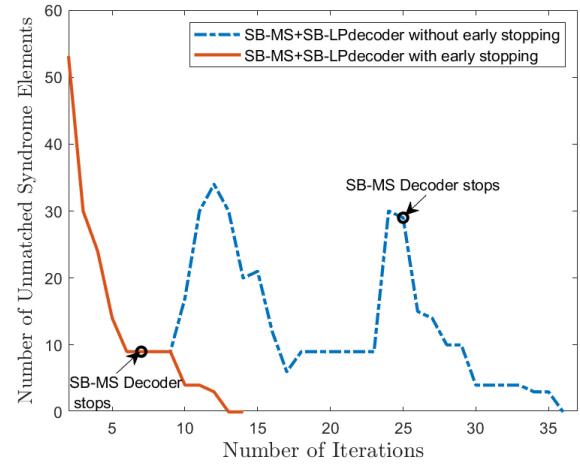


Fig. 3. Convergence behavior of the combined SB-MS and SB-LP decoder, with and without early stopping criterion, for the same sample error vector as in Fig.1.

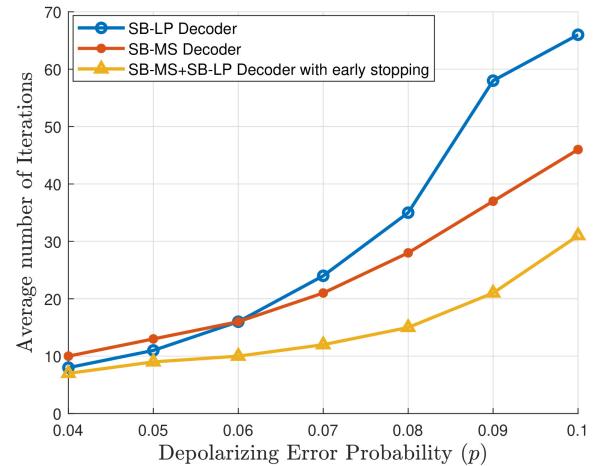


Fig. 4. Average number of iterations for the SB-MS decoder, the SB-LP decoder, and the combined SB-MS and SB-LP decoder with and without early stopping criterion, for the [[882, 24]] QLDPC code B1 from [11].

The results reveal that the SB-MS decoder introduces an error floor at a logical error of  $10^{-2}$ , and is significantly outperformed by the proposed SB-LP decoder. While the SB-LP decoder exhibits a worse threshold behavior, it improves the logical error rate for any depolarizing error probability lower than 0.07, providing almost two orders of magnitude of improvement in the logical error rate at a depolarizing error probability of 0.04.

The figure also shows the performance of the combination of the SB-MS and SB-LP algorithms, both with and without the early stopping criterion for SB-MS. Both solutions achieve a good logical error rate for high depolarizing error probabilities (greater than 0.06), but without the application of the early stopping criterion an error floor is introduced due to the poorer quality of the information passed by the SB-MS decoder to the SB-LP decoder.

Fig. 3 illustrates the effect of the early stopping criterion on the result achieved by the SB-MS algorithm. In the case where the early stopping criterion is not employed (dashed line), the SB-MS algorithm will stop when it reaches the maximum number of iterations, while in the case of early stopping (solid line) it can be observed that the SB-MS algorithm will stop near the first local minimum of the number of unmatched syndrome elements.

Note that, even if the SB-MS decoder is halted after a fixed number of iterations (25 in this example), convergence of the subsequent SB-LP algorithm can still be achieved. However, this comes at the cost of a high overall number of iterations (36 in this example); in particular, a high number of SB-LP decoder iterations is required because the initialization provided by the SB-MS algorithm is far from optimal. These additional iterations can be sufficient to prevent the decoder's convergence in some cases, introducing the undesirable error floor in Fig. 2. On the other hand, for the same error pattern, employing the early stopping criterion in SB-MS avoids the subsequent oscillations, reducing the number of iterations and providing the SB-LP decoder with a better initialization. For this example, only 7 SB-MS iterations are required to meet the stopping criterion, followed by 4 further iterations of the proposed SB-LP decoder.

Fig. 4 shows the average total number of iterations as a function of the depolarizing error probability  $p$ . It can be observed that the average number of iterations of the proposed combined SB-MS and SB-LP with early stopping criterion is always lower than that obtained with SB-MS or SB-LP alone. From the figure, two important conclusions can be drawn. The first is that for high values of depolarizing error probability, the combined SB-MS and SB-LP with early stopping criterion requires a significantly lower number of iterations for convergence than that of the SB-MS or SB-LP decoder alone (note that at these values of  $p$ , this combined decoder also provides the best logical error rate, as shown in Fig. 2). The second is that for low values of the depolarizing error probability, while there is not a substantial difference in the total number of iterations of the different decoders, the logical error rate of the combined decoder with early stopping criterion sees a significant improvement as shown in Fig. 2; while approximately 10 iterations are required for all three decoders, the combined decoder provides more than three orders of magnitude improvement in the logical error rate at  $p = 0.04$ .

## V. CONCLUSION

We have proposed a novel low-complexity syndrome-based decoder for QLDPC codes based on linear programming, which can be used either as a standalone decoder or as a post-processing decoding step for SB-MS decoding. We also proposed an early stopping criterion for the SB-MS decoder based on the syndrome Hamming distance that judiciously triggers the transition between the two decoding algorithms. The proposed approach was shown to be capable

of providing a significant improvement in the logical error rate performance as well as reducing the total number of decoding iterations.

## REFERENCES

- [1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.
- [2] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, pp. 1098–1105, Aug 1996.
- [3] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. Sloane, "Quantum error correction and orthogonal geometry," *Physical Review Letters*, vol. 78, no. 3, p. 405, 1997.
- [4] J. R. Wootton and D. Loss, "High Threshold Error Correction for the Surface Code," *Physical Review Letters*, vol. 109, no. 16, oct 2012.
- [5] 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.
- [6] P. Panteleev and G. Kalachev, "Asymptotically Good Quantum and Locally Testable Classical LDPC Codes," in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2022, 2022, p. 375–388.
- [7] N. P. Breuckmann and J. N. Eberhardt, "Quantum low-density parity-check codes," *PRX Quantum*, vol. 2, no. 4, p. 040101, 2021.
- [8] N. Raveendran and B. Vasić, "Trapping Sets of Quantum LDPC Codes," *Quantum*, vol. 5, p. 562, oct 2021.
- [9] N. Raveendran, M. Bahrami, and B. Vasic, "Syndrome-generalized belief propagation decoding for quantum memories," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.
- [10] Y.-H. Liu and D. Poulin, "Neural belief-propagation decoders for quantum error-correcting codes," *Physical review letters*, vol. 122, no. 20, p. 200501, 2019.
- [11] P. Panteleev and G. Kalachev, "Degenerate quantum LDPC codes with good finite length performance," *Quantum*, vol. 5, p. 585, nov 2021.
- [12] J. Valls, F. Garcia-Herrero, N. Raveendran, and B. Vasić, "Syndrome-based Min-sum vs OSD-0 decoders: FPGA Implementation and Analysis for Quantum LDPC codes," *IEEE Access*, vol. 9, pp. 138734–138743, 2021.
- [13] J. Du Crest, M. Mhalla, and V. Savin, "Stabilizer inactivation for message-passing decoding of quantum LDPC codes," in *2022 IEEE Information Theory Workshop (ITW)*. IEEE, 2022, pp. 488–493.
- [14] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 954–972, 2005.
- [15] J. X. Li and P. O. Vontobel, "LP decoding of quantum stabilizer codes," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 1306–1310.
- [16] O. Fawzi, L. Grouès, and A. Leverrier, "Linear programming decoder for hypergraph product quantum codes," in *2020 IEEE Information Theory Workshop (ITW)*. IEEE, 2021, pp. 1–5.
- [17] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge university press, 2010.
- [18] M. P. Fossorier, M. Mihaljević, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Transactions on communications*, vol. 47, no. 5, pp. 673–680, 1999.
- [19] P. O. Vontobel and R. Koetter, "Towards Low-Complexity Linear-Programming Decoding," in *4th International Symposium on Turbo Codes & Related Topics; 6th International ITG-Conference on Source and Channel Coding*, 2006, pp. 1–9.