Sequential BP-based Decoding of QLDPC Codes

Mohsen Moradi (mmorad11@asu.edu), Salman Habib (shabib@tamut.edu),
Vahid Nourozi (nourozi@nmsu.edu), and David G. M. Mitchell (dgmm@nmsu.edu)
Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM 88003, USA

Motivation and Goal

Quantum LDPC (QLDPC) codes are promising for scalable quantum error correction, but conventional flooding BP can stall on QLDPC Tanner graphs because of short cycles, degeneracy, and symmetric message evolution.

Goal: improve BP decoding by changing only the update schedule.
Same code. Same BP equations. No post-processing.

QLDPC Syndrome Decoding

For the independent Pauli-X channel of a CSS code, an error vector $\boldsymbol{x} \in \mathbb{F}_2^n$ produces syndrome

$$\boldsymbol{s} = H_1\boldsymbol{x} \pmod 2.$$

Given a hard estimate $\hat{\boldsymbol{x}}$, define the residual mismatch

$$\boldsymbol{\delta} = \boldsymbol{s} \oplus H_1\hat{\boldsymbol{x}}, \qquad w = \|\boldsymbol{\delta}\|_1.$$

A check node is satisfied if its residual bit is zero. The decoder converges when $w = 0$; in quantum decoding, the final correction must also be in the correct logical coset.

Why Flooding BP Struggles

- **Short cycles:** stabilizer commutativity creates highly loopy Tanner graphs.
- **Degeneracy:** many Pauli errors can share the same syndrome.
- **Flooding synchrony:** simultaneous updates may reinforce oscillations or pseudo-codewords.

Sequential scheduling breaks harmful synchrony by immediately reusing fresh messages.

BP Message Updates

Let $m_{v\to c}$ and $m_{c\to v}$ denote VN-to-CN and CN-to-VN messages. Then,

$$\mu = \log\frac{1-p_x}{p_x},$$

$$m_{c\to v} = 2\,\mathrm{atanh}\left((-1)^{s(c)}\prod_{u\in\partial c\setminus v}\tanh\frac{m_{u\to c}}{2}\right),$$

$$m_{v\to c} = \mu + \sum_{c'\in\partial v\setminus c} m_{c'\to v}, \qquad m_v = \mu + \sum_{c\in\partial v} m_{c\to v}.$$

The hard decision is $\hat{x}_v = \mathbf{1}[m_v < 0]$.

Main Contributions

- Introduce **SCNS-BP** and **SVNS-BP** for QLDPC syndrome decoding.
- Show that sequential BP improves FER and reduces non-convergence near BP complexity.
- Combine sequential scheduling with BP guided decimation to obtain **SBPGD**.

Scheduling

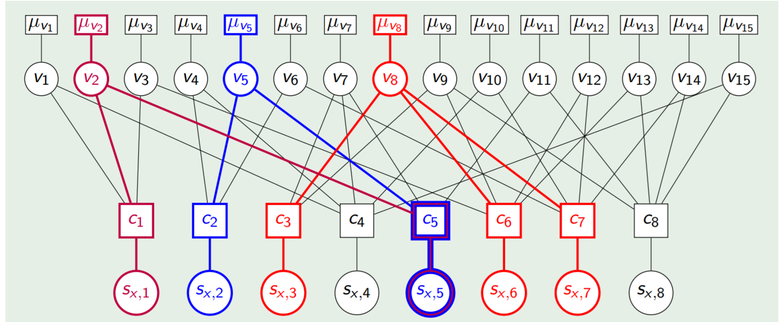

Figure: SVNS visualization: previous v_5, v_8 (blue, red) nodes and current scheduled v_2 (purple) node. Shared nodes/edge is $(c_5, s_{x,5})$.

SCNS-BP and SVNS-BP

SCNS-BP	Visit check nodes sequentially. After updating a CN, immediately refresh the beliefs/messages of its neighboring VNs.
SVNS-BP	Visit variable nodes sequentially. For the selected VN, update incident CN-to-VN messages, the VN belief, and outgoing VN-to-CN messages.

Both methods use the same BP equations as flooding BP; only the update order changes.

Sequential BPGD (SBPGD)

BPGD alternates short BP runs with guided decimation. SBPGD replaces the flooding BP subroutine by SCNS or SVNS.

Run T sequential BP iteration → Check syndrome → Decimate most reliable VN

Sequential schedules often create stronger reliabilities earlier, so SBPGD needs fewer decimation rounds on average.

Complexity and Message Traffic

The proposed schedules store the same edge messages and node beliefs as standard BP. The main cost change is the number and order of message updates before convergence.

Decoder	$p_x = .04$	$p_x = .05$	$p_x = .06$
Flooding BP	162376	221207	290289
SCNS-BP	28331	37073	56329
SVNS-BP	24272	33817	53736

Average CN-to-VN message counts for the $[[1922, 50, 16]]$ C2 code with $T = 100$.

Sequential-BP Results

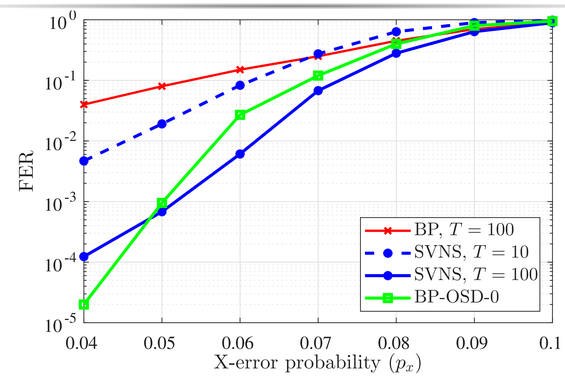

Figure: Sequential VN scheduling for $[[882, 24, 18 \le d \le 24]]$.

Summary: sequential schedules improve FER under the same iteration cap and reduce message traffic through faster convergence.

Learning Extension: RL-QSVNS

The learning extension keeps the SVNS update rule fixed but learns **which VN to update next** from residual-syndrome information.

State: local mismatch pattern around VN v_i. **Action:** choose next VN.
Reward: decrease in $w = \|\boldsymbol{\delta}\|_1$.

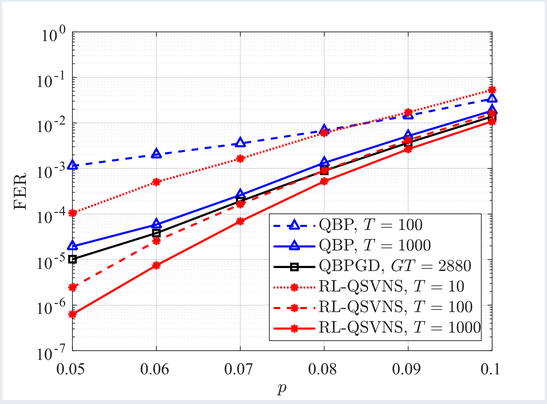

Figure: FER for the $[[288, 12, 18]]$ BB code.

Takeaway

- **Sequential BP** is a low-overhead way to reduce QLDPC BP non-convergence.
- **SBPGD** shows that better intermediate beliefs can reduce decimation effort.
- **RL-QSVNS** generalizes the same idea by learning a state-dependent VN schedule.