Decoding Quantum Surface Codes via Belief-Propagation

(an extended work of DOI: 10.1109/JSAIT.2020.3011758)

Kao-Yueh Kuo and Ching-Yi Lai

Institute of Communications Engineering, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan

2021 Feb.

Motivation

- Quantum states (e.g., qubits) are sensitive and error-prone.
- **Quantum surface codes** (a kind of *stabilizer codes*) provide an implementable topological structure for quantum error-correction.
- Since the coherence decays quickly, a fast decoding is desired.
- For an N-qubit surface code, the often-mentioned decoding algorithms have a complexity $O(N^2)$ (by minimum-weight-matching (MWM)) or $O(N \log N)$ (by renormalization-group (RG)).
- We intend to use **belief-propagation (BP)**, which has a complexity $O(N\tau)$, where τ is the number of iteration, $\tau = O(\log\log N)$ for a well BP convergence.
- The **practical complexity** and **decoding performance** of BP, for decoding surface (or stabilizer) codes, need to be improved.

Stabilizer codes

- $\mathcal{G}_N$: the N-fold Pauli group.

$$\mathcal{G}_N = \{E = \omega E_1 \otimes \cdots \otimes E_N \mid \omega \in \{\pm 1, \pm i\}, E_n \in \{I, X, Y, Z\}\},$$

where $I = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $X = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, $Z = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$, and $Y = iXZ$.
- It suffices to discard ω and denote, e.g., if $N = 5$, $I \otimes Z \otimes X \otimes I \otimes Y = IZXIY = Z_2 X_3 Y_5$.
- $\{I, X, Y, Z\}^N \subset \mathcal{G}_N$: all positive-phase elements in $\mathcal{G}_N$, forming a basis for unitary matrices on $\mathbb{C}^{2^N}$.
- $\{S_m\}_{m=1}^{N-K} \subset \{I, X, Y, Z\}^N$: a set of $N - K$ **independent** and **commuting** generators, i.e.,
  - S_m cannot be generated by other $S_{m'}$'s; and $S_m S_{m'} = S_{m'} S_m$.
- $\mathcal{S} < \mathcal{G}_N$: a **stabilizer group** generated by S_m's.
  - An $F \in \mathcal{S}$ is called a **stabilizer**. $-I^{\otimes N}$ will not in $\mathcal{S}$.
- $\mathcal{C}(\mathcal{S})$: an $[[N, K]]$ **stabilizer code** (a 2^K-dim. subspace in $\mathbb{C}^{2^N}$),

$$\mathcal{C}(\mathcal{S}) = \{|\psi\rangle \in \mathbb{C}^{2^N} \mid F|\psi\rangle = |\psi\rangle \ \forall\ F \in \mathcal{S}\}.$$

Measurement

- $E = E_1 \cdots E_N \in \{I, X, Y, Z\}^N$: an (unknown) N-qubit Pauli error.
- $S = [S_{mn}] = \begin{bmatrix} S_1 \\ \vdots \\ S_M \end{bmatrix} \in \{I, X, Y, Z\}^{M \times N}$: a **check matrix**, where $M \geq N - K$ and

$$S_m = S_{m1} \cdots S_{mN} \in \{I, X, Y, Z\}^N$$

corresponds to the mth measurement $\left\{\frac{I+S_m}{2}, \frac{I-S_m}{2}\right\}$.
- For any two Pauli operators F, F', denote

$\langle F, F' \rangle = 0$ if F, F' commte, and $\langle F, F' \rangle = 1$ if F, F' anticommte.

- $z = (z_1, \ldots, z_M) \in \{0, 1\}^M$: a **binary (error) syndrome**, where

$$z_m = \langle E, S_m \rangle = \sum_{n=1}^{N} \langle E_n, S_{mn} \rangle \mod 2.$$

Decoding

- $p_n = (p_n^I, p_n^X, p_n^Y, p_n^Z)$: the initial distribution of E_n according to some error model, e.g., for a **depolarizing channel** with error rate ϵ,

$$p_n = (1 - \epsilon,\ \epsilon/3,\ \epsilon/3,\ \epsilon/3).$$

- A decoding should have, given $S \in \{I, X, Y, Z\}^{M \times N}$, $z \in \{0, 1\}^M$, and optionally $\{p_n\}_{n=1}^N$,

$$\mathrm{Dec}(S, z, \{p_n\}_{n=1}^N) = \hat{E}$$

  s.t. $\hat{E} \in E\mathcal{S}$ with a probability as high as possible.
- $\hat{E}^\dagger$ will be applied as the correction.
- We may denote $\hat{E} = \mathrm{Dec}(z)$ by assuming S and $\{p_n\}_{n=1}^N$ fixed.

Correction radius

- $\mathrm{wt}(E)$: the **weight** (number of non-identity entries) of an $E \in \mathcal{G}_N$.
- d: the **(minimum) distance** of the code $\mathcal{C}(\mathcal{S})$ defined as

$$d = \min\{\mathrm{wt}(E) \mid E \in \{I, X, Y, Z\}^N \setminus \mathcal{S},\ \langle E, S_m \rangle = 0\ \forall\, m\}.$$

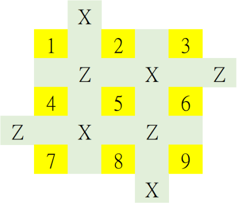
- $\gamma_d = t_d/N$: normalized correctable radius (with probability $= 1$) of **bounded-distance decoding** (BDD), where $t_d = \lfloor \frac{d-1}{2} \rfloor$.
- $\gamma_s = t_s/N$: normalized correctable radius (with probability ≈ 1) of **syndrome-based decoding** (SBD), based on collecting $\hat{E} = \mathrm{Dec}(z)$ for all $z \in \{0,1\}^M$.
  - For most channels, minimizing $\mathrm{wt}(E)$ can maximize $P(E)$.
  - If $\hat{E} = \underset{\substack{E \in \mathcal{G}_N: \\ \langle E, S_m \rangle = z_m\ \forall\, m}}{\arg\min}\ \mathrm{wt}(E)$, then $t_s \geq t_d$; but t_s is hard to compute.
- $\gamma = t/N$: normalized correctable radius (with probability ≈ 1) of SBD extended by **degeneracy**, based on collecting all errors in

$$\{E \in \mathcal{G}_N \mid E \in \hat{E}\mathcal{S},\ \hat{E} = \mathrm{Dec}(z),\ z \in \{0,1\}^M\}.$$

  - t is again hard to compute, but it could be $\gamma > 0$ even if $\gamma_d \to 0$.

Surface codes (due to Kitaev)

- $[[N = L^2, K = 1, d = L]]$ stabilizer codes, with odd $L \geq 3$.
- One qubit formation is encoded (protected) by an $L \times L$ lattice.
- For example, $L = 3$, an $[[N = 9, K = 1, d = 3]]$ code:



Check matrix

$$S = \begin{bmatrix} S_1 \\ \vdots \\ S_8 \end{bmatrix} = \begin{bmatrix} X & X & I & I & I & I & I & I & I \\ Z & Z & I & Z & Z & I & I & I & I \\ I & X & X & I & X & X & I & I & I \\ I & I & Z & I & I & Z & I & I & I \\ I & I & I & Z & I & I & Z & I & I \\ I & I & I & X & X & I & X & X & I \\ I & I & I & I & Z & Z & I & Z & Z \\ I & I & I & I & I & I & I & X & X \end{bmatrix}.$$

The weight of each measurement ≤ 4.

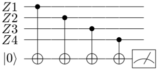
Logical basis states:

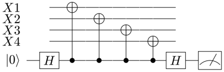
$$|0\rangle \mapsto |0_\mathrm{L}\rangle \propto |000000000\rangle + |110000000\rangle + |011011000\rangle + |101011000\rangle + |000110110\rangle + |110110110\rangle + |011101110\rangle + |101101110\rangle + |000000011\rangle + |110000011\rangle + |011011011\rangle + |101011011\rangle + |000110101\rangle + |110110101\rangle + |011101101\rangle + |101101101\rangle$$

$$|1\rangle \mapsto |1_\mathrm{L}\rangle \propto |111111111\rangle + |001111111\rangle + |100100111\rangle + |010100111\rangle + |111001001\rangle + |001001001\rangle + |100010001\rangle + |010010001\rangle + |111111100\rangle + |001111100\rangle + |100100100\rangle + |010100100\rangle + |111001010\rangle + |001001010\rangle + |100010010\rangle + |010010010\rangle$$

Example errors and syndromes

For $S_m = Z_1Z_2Z_3Z_4$ or $X_1X_2X_3X_4$, the measurement is as simple as





[MWM]: D. S. Wang, A. G. Fowler, A. M. Stephens, and L. C. L. Hollenberg, Threshold error rates for the toric and planar codes, Quant. Inf. Comput. 10, p. 456, 2010.

- $[[9,1,3]]$: since $d = 3$, it can correct any weight-one errors.

  if $E = Z_1$, then $z = (1,0,0,0,0,0,0,0)$

  $Z_9\,|0_L\rangle \propto$
  $|000000000\rangle + |110000000\rangle + |011011000\rangle + |101011000\rangle + |000110110\rangle + |110110110\rangle + |011101110\rangle +$
  $|101101110\rangle - |000000011\rangle - |110000011\rangle - |011011011\rangle - |101011011\rangle - |000110101\rangle - |110110101\rangle - |0$

  $Z_9\,|1_L\rangle \propto$
  $|111111111\rangle + |001111111\rangle + |100100111\rangle + |010100111\rangle + |111001001\rangle + |001001001\rangle + |100010001\rangle +$
  $|010010001\rangle - |111111100\rangle - |001111100\rangle - |100100100\rangle - |010100100\rangle - |111001010\rangle - |001001010\rangle - |1$

  if $E = X_1$, then $z = (0,1,0,0,0,0,0,0)$

  $X_9\,|0_L\rangle \propto |000000001\rangle + |110000001\rangle + |011011001\rangle + |101011001\rangle + |000110111\rangle + |110110111\rangle +$
  $|011101111\rangle + |101101111\rangle + |000000010\rangle + |110000010\rangle + |011011010\rangle + |101011010\rangle +$
  $|000110100\rangle + |110110100\rangle + |011101100\rangle + |101101100\rangle$

  $X_9\,|1_L\rangle \propto |111111110\rangle + |001111110\rangle + |100100110\rangle + |010100110\rangle + |111001000\rangle + |001001000\rangle +$
  $|100010000\rangle + |010010000\rangle + |111111101\rangle + |001111101\rangle + |100100101\rangle + |010100101\rangle +$
  $|111001011\rangle + |001001011\rangle + |100010011\rangle + |010010011\rangle$

  if $E = Y_1$, then $z = (1,1,0,0,0,0,0,0)$

  $\cdots$.

The asymptotic correction radius of surface codes

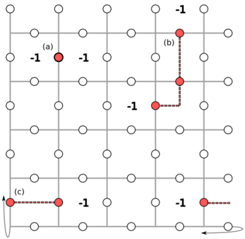
- Let ρ be a codeword state and $E\rho E^\dagger$ be a noisy state, $E \in \mathcal{G}_N$.

$$\rho = \alpha \left|1_{\mathrm{L}}\right\rangle \left\langle 1_{\mathrm{L}}\right| + \beta \left|0_{\mathrm{L}}\right\rangle \left\langle 0_{\mathrm{L}}\right| \quad \xmapsto{E} \quad E\rho E^\dagger$$
$$\mathrm{Meas}(E\rho E^\dagger) = E\rho E^\dagger \text{ with a syndrome } z \in \{0,1\}^{N-K}$$

  where the measurement will not affect the state $E\rho E^\dagger$.
- The decoding is expected to output $\hat{E}$ with smallest-weight error first.
- For example, $[[9,1,3]]$:
  $\hat{E} = Z_1$ is expected when $z = (1,0,0,0,0,0,0,0)$;
  $\hat{E} = X_1$ is expected when $z = (0,1,0,0,0,0,0,0)$;
  $\hat{E} = Y_1$ is expected when $z = (1,1,0,0,0,0,0,0)$; $\ldots$.
- Since there are many low-weight stabilizers, γ grows as N increases.
  - For an optimum decoding, $\gamma \to 18.9\%$ = quantum hashing bound (= quantum Hamming bound for small $\frac{K}{N} \to 0$).
  - **Threshold**: the achievable γ of some decoding algorithm.

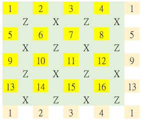
Often-mentioned decoding algorithms

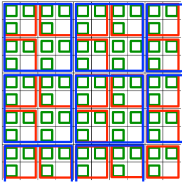
- Minimum-weight matching (MWM)



- X errors and Z errors separately decoded;
- based on the blossom algorithm (Edmonds, 1965), which has a complexity $O(N^3)$ and can be simplified to $O(N^2)$ for the case here.
- Achievable threshold $\approx 15.5\%$.

- Renormalization group (RG)





Toric codes:

- by concatenated decoding with a complexity $O(N \log N)$.
- Basic-RG threshold $\approx 7.8\%$.
- Improved-RG threshold $\approx 15.2\%$ (handling RG boundaries additionally).
- BP + Improved-RG threshold $\approx 16.4\%$.

[RG]: G. Duclos-Cianci and D. Poulin, Fast decoders for topological quantum codes, Phys. Rev. Lett. 104, p. 050504, 2010.

BP decoding of quantum codes

BP, as mentioned, has a complexity almost linear in N.

BP issues for decoding quantum codes:

- Complexity: handling I, X, Y, Z needs a **quaternary BP** (BP_4).
  - It is 16 times more complex than the classical **binary BP** (BP_2).
- Performance: **short cycles** in the check matrix's **Tanner graph** will affect the decoding convergence & degrade the decoding performance.

Our approach:

- Refine & Improve it as a BP_4 with additional memory effect (MBP_4):

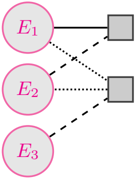
(Original) BP_4 $\xrightarrow{\text{refine}}$ Refined BP_4 $\xrightarrow{\text{improve}}$ MBP_4 (α_i)

(equivalent to BP_4 but with scalar messages)
lower complexity

(normalize the scalar messages by α_i with additional memory effect)
improved performance

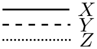
BP: Message Passing on Tanner Graph

- BP decoding is an iterative **message-passing** algorithm run on a bipartite graph (called **Tanner graph**) defined by S.
- For example, $S = \begin{bmatrix} X & Y & I \\ Z & Z & Y \end{bmatrix}$ has a Tanner graph:



$(S_1):\ \langle E_1, X\rangle + \langle E_2, Y\rangle = z_1$

$(S_2):\ \langle E_1, Z\rangle + \langle E_2, Z\rangle + \langle E_3, Y\rangle = z_2$



This graph contains a cycle of length 4.

- BP$_4$ starts with an initial belief (for the error in each qubit)

$$p_n = (p_n^I, p_n^X, p_n^Y, p_n^Z) \quad \left(\text{here} = (1-\epsilon_0, \frac{\epsilon_0}{3}, \frac{\epsilon_0}{3}, \frac{\epsilon_0}{3})\right) \tag{1}$$

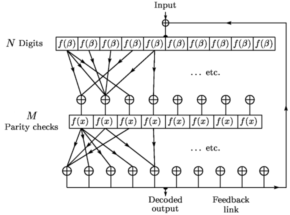
(given S and z) to compute an updated belief

$$q_n = (q_n^I, q_n^X, q_n^Y, q_n^Z) \tag{2}$$

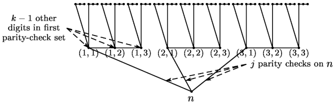
and infer $\hat{E}_n = \arg\max_{W \in \{I,X,Y,Z\}} q_n^W$.

The Message Passing

- BP_2 (classical case):

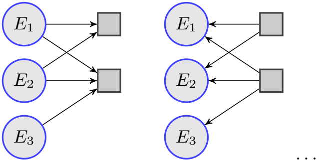


- The connected edges depends on a (parity-)check matrix $H \in \{0,1\}^{M\times N}$
- Updating the belief at n is like a tree.



- BP is close to maximum-likelihood decoding (MLD) if H is designed without short cycles.

- BP_4 (quantum case):
- The scenarios is similar, if we draw by S:



- But BP_4 passes vectors of length 4, unlike that BP_2 passes scalars.
- This increases the complexity 16 times.
- The Tanner graph of a large S inevitably has many short cycles due to commutation property:
- creating strong dependencies between the messages inputting to a node, possibly making BP far from MLD.

[Gal]: R. G. Gallager, Low-density parity-check codes (MIT Press, Cambridge, MA, 1963).

Original BP_4: (every message is a vector)

- To complete the 1st iteration:
  variable node n passes to check node m the message
  $\boldsymbol{q}_{n\to m} = (q^I_{mn}, q^X_{mn}, q^Y_{mn}, q^Z_{mn}) = \boldsymbol{p}_n$,
  and check node m passes to variable node n the message
  $\boldsymbol{r}_{m\to n} = (r^I_{mn}, r^X_{mn}, r^Y_{mn}, p^Z_{mn})$, with

$$r^W_{mn} = \sum_{\substack{E|_{\mathcal{N}(m)}:\ E_n=W, \\ \langle E|_{\mathcal{N}(m)}, S_m|_{\mathcal{N}(m)}\rangle = z_m}} \left(\prod_{n'\in\mathcal{N}(m)\setminus n} q^{E_{n'}}_{mn'} \right)$$

  for $W \in \{I, X, Y, Z\}$, where $\mathcal{N}(m) = \{n \mid S_{mn} \neq I\}$.

- For any next iteration, $\boldsymbol{q}_{n\to m} = (q^I_{mn}, q^X_{mn}, q^Y_{mn}, q^Z_{mn})$ with

$$q^W_{mn} \propto p^W_n \prod_{m'\in\mathcal{M}(n)\setminus m} r^W_{m'n}$$

  where $\mathcal{M}(n) = \{m \mid S_{mn} \neq I\}$.

- To infer $\hat{E}_n$ is by $\boldsymbol{q}_n = (q^I_n, q^X_n, q^Y_n, q^Z_n)$ with $q^W_n = p^W_n \prod_{m\in\mathcal{M}(n)} r^W_{mn}$.

Refine and Improve

To refine: (to have a lower complexity)

- An observation: $\langle E_1, S_{m1}\rangle = z_m + \sum_{n=2}^{N}\langle E_2, S_{mn}\rangle \mod 2$
  - In other words, the message from a neighboring check will tell us more likely whether the error E_1 commutes or anti-commutes with S_{m1}
- We derived a refined algorithm by, e.g., if $S_{mn} = X$, then passing $d_{n\to m} = (q_{mn}^I + q_{mn}^X) - (q_{mn}^Y + q_{mn}^Z)$ is sufficient for computation.
- Every message becomes a scalar, and the check-node efficiency is 16-fold improved.

Using scalar messages allows simple improvement:

- Ever message can be normalized by a parameter $\alpha_i > 0$.
- Wrong beliefs caused by **short cycles** are usually suppressed.
- We do a derivation by joining *gradient descend (GD) & BP rules* to have a BP_4 with additional memory effect (MBP_4).[1]
  (There is another parameter β after derivation; but α_i is more dominated so we focus on α_i)

[1] BP is like a recurrent neural network (RNN)—we found what we create is like **inhibition** between nodes, which enhances the perception capability in Hopfield nets.

MBP_4 with a parameter α_i

- **The most high-complexity step** is refined.
- **The computation in (3) and (4)** improves the performance.

  (The nonlinear function can be efficiently implemented by Schraudolph's approximation)
- **Using (3) without (4) is the classical normalized BP** (w/o) additional memory.

Algorithm 1 : Quaternary BP (BP_4) with message normalization and inhibition between nodes controlled by α_i.

Input: $S \in \{I, X, Y, Z\}^{M\times N}$, $\{\boldsymbol{p}_n = (p_n^I, p_n^X, p_n^Y, p_n^Z)\}_{n=1}^N$, target $z \in \{0,1\}^M$, and a real parameter α_i.

Initialization. For $n = 1, 2, \dots, N$ and $m \in \mathcal{M}(n)$, let

$$d_{n\to m} = q_{n\to m}^{(0)} - q_{n\to m}^{(1)},$$

where $q_{n\to m}^{(0)} = p_n^I + p_n^{S_{mn}}$ and $q_{n\to m}^{(1)} = 1 - q_{n\to m}^{(0)}$.

Horizontal Step. For $m = 1, 2, \dots, M$ and $n \in \mathcal{N}(m)$, compute

$$\delta_{m\to n} = (-1)^{z_m} \prod_{n' \in \mathcal{N}(m)\setminus n} d_{n'\to m},$$

Vertical Step. For $n = 1, 2, \dots, N$ and $m \in \mathcal{M}(n)$, do:

- Compute

$$r_{m\to n}^{(0)} = \left(\tfrac{1+\delta_{m\to n}}{2}\right)^{1/\alpha_i},\ r_{m\to n}^{(1)} = \left(\tfrac{1-\delta_{m\to n}}{2}\right)^{1/\alpha_i}, \quad (3)$$

$$q_{n\to m}^I = p_n^I \prod_{m' \in \mathcal{M}(n)\setminus m} r_{m'\to n}^{(0)},$$

$$q_{n\to m}^W = p_n^W \prod_{m' \in \mathcal{M}(n)\setminus m} r_{m'\to n}^{\langle W, S_{m'n}\rangle}, \text{ for } W \in \{X, Y, Z\}.$$

- Let

$$q_{n\to m}^{(0)} = a_{mn}\left(q_{n\to m}^I + q_{n\to m}^{S_{mn}}\right) / \left(\tfrac{1+\delta_{m\to n}}{2}\right)^{1-1/\alpha_i},$$
$$q_{n\to m}^{(1)} = a_{mn}\left(\textstyle\sum_{W'} q_{n\to m}^{W'}\right) / \left(\tfrac{1-\delta_{m\to n}}{2}\right)^{1-1/\alpha_i}, \quad (4)$$

  where $W' \in \{X, Y, Z\} \setminus S_{mn}$ and a_{mn} is a chosen scalar such that $q_{n\to m}^{(0)} + q_{n\to m}^{(1)} = 1$.
- Update: $d_{n\to m} = q_{n\to m}^{(0)} - q_{n\to m}^{(1)}$.

Hard Decision. For $n = 1, 2, \dots, N$, compute

$$q_n^I = p_n^I \textstyle\prod_{m\in\mathcal{M}(n)} r_{mn}^{(0)}$$
$$q_n^W = p_n^W \textstyle\prod_{m\in\mathcal{M}(n)} r_{mn}^{\langle W, S_{mn}\rangle}, \text{ for } W \in \{X, Y, Z\}.$$

Let $\hat{E}_n = \arg\max_{W\in\{I,X,Y,Z\}} q_n^W$.

- Let $\hat{E} = \hat{E}_1 \hat{E}_2 \cdots \hat{E}_N$.
  - If $\langle \hat{E}, S_m \rangle = z_m$ for $m = 1, 2, \dots, M$, halt and return "SUCCESS";
  - otherwise, if a maximum number of iterations is reached, halt and return "FAIL";
  - otherwise, repeat from the horizontal step.

The parameter α_i (assume $\alpha_i > 0$)

- Against short cycles: original BP decouples the $n \to m$ message and the $m \to n$ message **that are passed on the same edge**.
  - suitable for the case of less short cycles; need a different strategy here.
  - Introducing α_i (especially in (4)) breaks the decoupling rule and creates strong memory effect at check-node side (fed back from variable-node side).
- Properties of α_i, related to the degeneracy:
  - A code is said to have strong degeneracy if there are many measurements with $\mathrm{wt}(S_m) \ll d$.
  - A larger $\alpha_i > 1$ corresponds to a careful (smaller-step) search and the memory effect provides suppression, suitable for codes with less degeneracy.
  - A smaller $\alpha_i < 1$ corresponds to an aggregate (larger-step) search and the memory effect provides *momentum*, suitable for codes with strong degeneracy.

BP as GD, and different schedule

- Classically, BP is like doing a GD opt.: LHS (a) (c). (need $\alpha_i > 1$)
- If $d \gg \mathrm{wt}(S_m)$, it has strong degeneracy: LHS (b) (d). (need $\alpha_i < 1$)
- To take an aggregate update, using a serial schedule also helps.

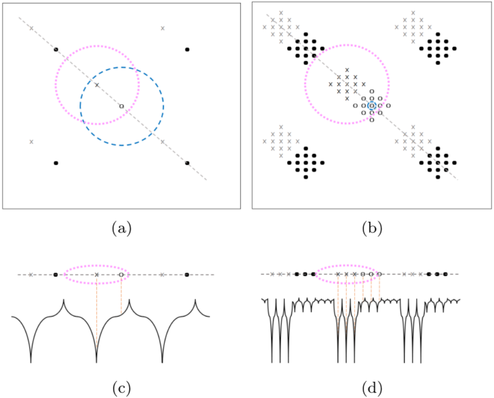
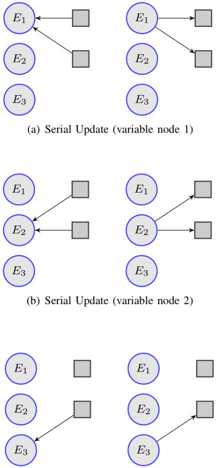


FIG. 2. Illustrations of the energy topology J_S of the decoding problem. (a) A classical code. (b) A degenerate quantum code. (c) and (d) are the profiles of the topologies along a dashed lines in (a) and (b), respectively.



(a) Serial Update (variable node 1)

(b) Serial Update (variable node 2)

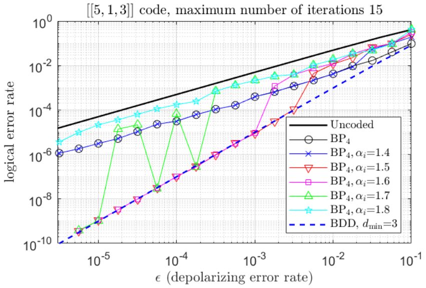
(c) Serial Update (variable node 3)

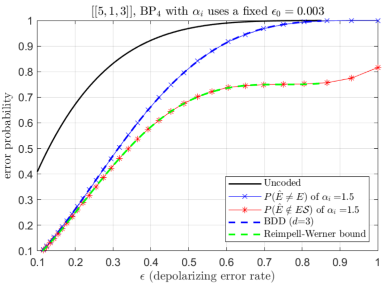
The five-qubit code $[[N=5,\ K=1,\ d=3]]$

- The code does not have strong degeneracy but has many short cycles:

$$S = \begin{bmatrix} X & Z & Z & X & I \\ I & X & Z & Z & X \\ X & I & X & Z & Z \\ Z & X & I & X & Z \end{bmatrix}.$$

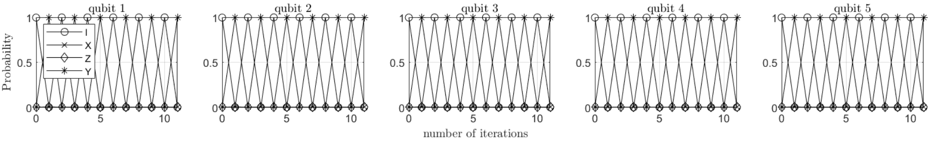
- This code should be able to correct any weight-one errors.
  - BP_4 without α_i (MBP_4 with $\alpha_i = 1$) cannot correct the error $IIIYI$.
  - BP_4 with $\alpha_i \approx 1.5$ successfully corrects all weight-one errors.





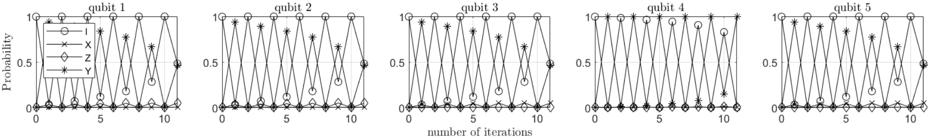
The convergence (for decoding the error $IIIYI$ by setting $\epsilon_0 = 0.003$)

- Without using α_i (equivalent to MBP$_4$ with $\alpha_i = 1$), each output belief q_n, $n = 1, 2, 3, 4, 5$, keeps oscillating:



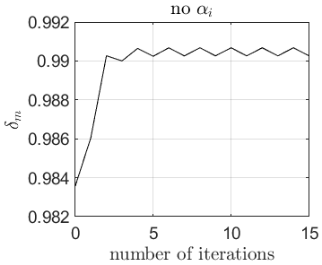
- With $\alpha_i = 1.5$, it converges correctly by **suppressing the wrong belief**:

I I I Y I

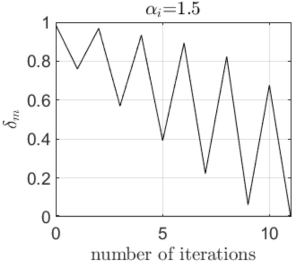


The convergence (cont.)

- Define $\delta_m \triangleq \prod_{n \in \mathcal{N}(m)} \left(\hat{q}_{nm}^{(0)} - \hat{q}_{nm}^{(1)} \right)$ and plot the change:
  ($IIIYI$ causes all $z_m = 1$ and the same $\delta_m\ \forall\ m$ during iteration; target $\delta_m < 0$)



(a) without α_i (note: the *swing is very tiny*).



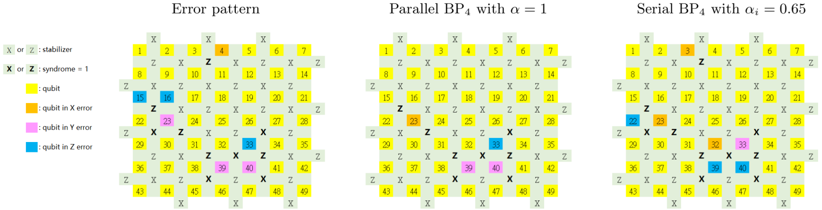
(b) with $\alpha_i = 1.5$.

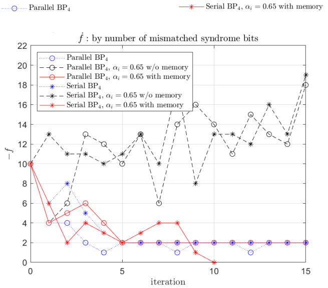
Recall that α_i **introduces memory-effect**:

- This is like a simplified **long short-term memory** (LSTM) method.

Surface code: strong degeneracy (an example by $L = 7$)

- Consider the error $E = X_4 Z_{15} Z_{16} Y_{23} Z_{33} Y_{39} Y_{40}$
- $\mathrm{wt}(E) = 7 > \mathrm{wt}(S_m)$: it needs $\alpha_i < 1$ and serial schedule to decode

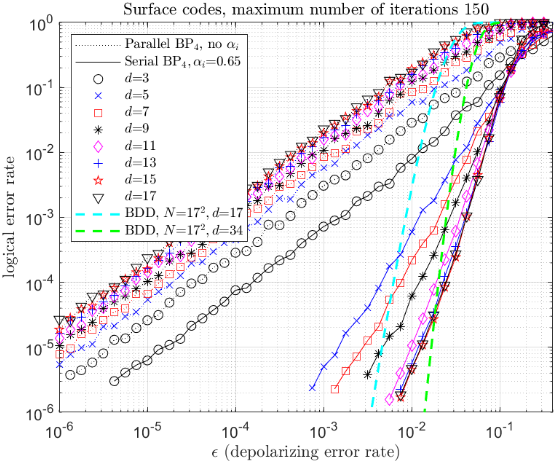




BP runs a GD optimization with an energy function positively correlated to the number of unmatched syndromed bits.

Surface codes $[[N = L^2, K = 1, d = L]]$

- Serial MBP_4, with small $\alpha_i < 1$, provides good results.



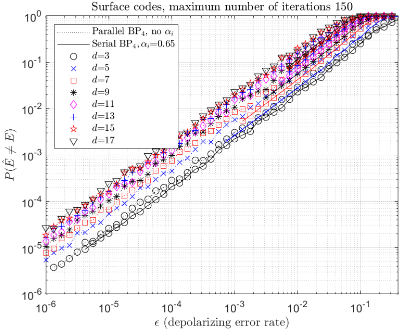
Improvement is from exploiting the degeneracy

- Write the logical error rate as:

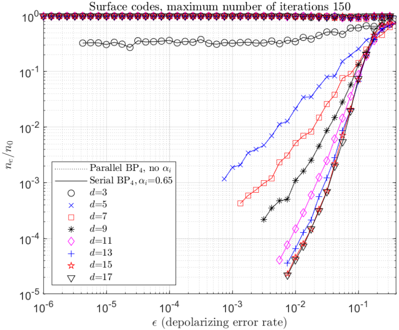
$$\begin{aligned} P(\hat{E} \notin E\mathcal{S}) &= P(\hat{E} \notin E\mathcal{S}, \hat{E} \neq E) \\ &= P(\hat{E} \neq E) \times P(\hat{E} \notin E\mathcal{S} \mid \hat{E} \neq E) = \frac{n_0}{n} \times \frac{n_e}{n_0}. \end{aligned}$$

- We plot $\frac{n_0}{n}$ and $\frac{n_e}{n_0}$ (both the lower the better, and the lower $\frac{n_e}{n_0}$ means the more the decoder exploits the degeneracy):

Two schemes have similar $P(\hat{E} \neq E) = \frac{n_0}{n}$.

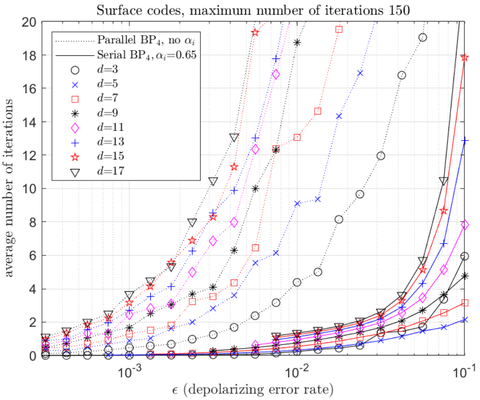


The proposed scheme has a much lower $\frac{n_e}{n_0}$.



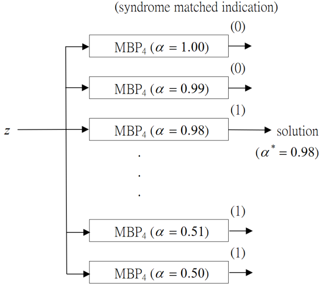
The convergence behavior

- By viewing the **average number of iterations**, it shows that the improvement is achieved by better algorithm convergence, rather than spending complexity on doing more iterations:



Further improvement

- BP can do about $2 \times$BDD [Gal].
- A good threshold needs about $(0.189\sqrt{N} \times 2) \times$BDD
- MBP$_4$ can be improved by a technique like Monte Carlo sampling (sometimes called parallel tempering): we use about 50 instances

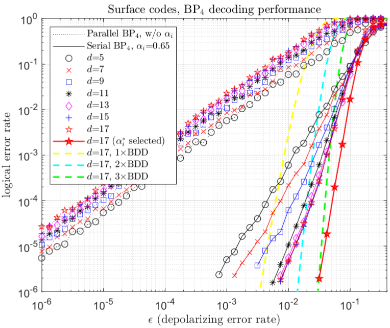


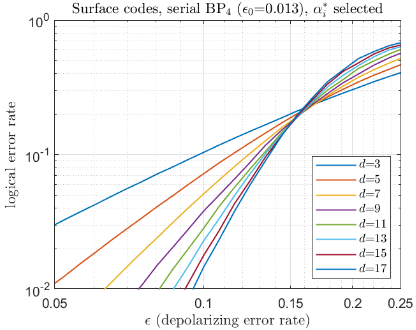
  - No need to combine the solutions from different MBP$_4$ instances. And if it runs in a sequential order, after the first syndrome matched indication $= 1$, the remaining instances can be skipped.
  - Precisely estimating an α^* can again use only one instance.[2]

[2] Interestingly, this possible simplification is similar to an observation when using Hopfield nets to do simulated-annealing: [Hop] J. Hopfield and D. Tank, "neural" computation of decisions in optimization problems, Biological cybernetics 52 (1985).

Further improvement (surface codes)

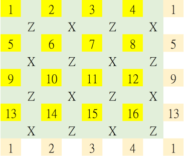
- Achieving a threshold $\sim 15.5\%$ to 16% for decoding surface codes.



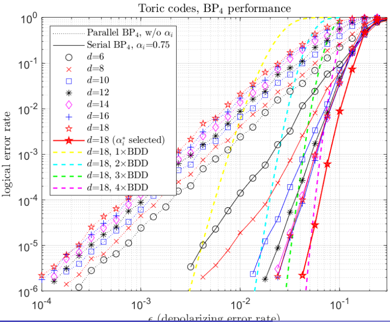


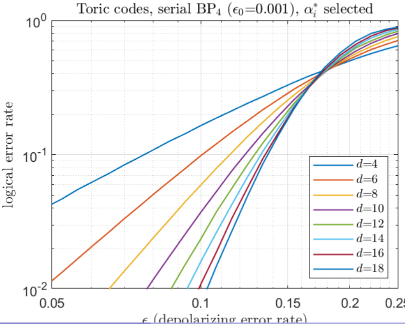
Further improvement (toric codes)

- If change to toric codes (without boundary conditions), $[[N = L^2, K = 2, d = L]]$ with even $L \geq 2$, e.g., $L = 4$:



- Achieving a threshold $\sim 17\%$ to 17.5% for decoding toric codes.





Conclusion and Ongoing Work

- We refine the BP_4 decoding of quantum codes to have a lower (check-node) complexity (16-fold improved).
- We improve the refined BP_4 as an MBP_4 with additional memory effect, keeping the same asymptotic complexity.
- We simulate the MBP_4 decoding performance for the $[[5,1,3]]$ code, surface codes, and toric codes.
- MBP_4 significantly improves the performance by exploiting the degeneracy, and it is achieved with a better convergence.
- The performance can be further improved by choosing an optimum α^* per the syndrome, achieving a threshold $\sim 15.5\%$ to 16% for decoding surface codes and $\sim 17\%$ to 17.5% for decoding toric codes.

Ongoing work:

- Partial parallelism; fault tolerance; estimating α_i^*; proof of convergence.

References

R. G. Gallager, *Low-Density Parity-Check Codes*, Cambridge, MA: MIT Press, 1963.

D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, pp. 399–431, 1999.

R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, pp. 533–547, 1981.

J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, pp. 498–519, 2001.

A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Ann. Phys.*, vol. 303, pp. 2–30, 2003.

D. Poulin and Y. Chung, "On the iterative decoding of sparse quantum codes," *Quantum Inf. Comput.*, vol. 8, pp. 987–1000, 2008.

K.-Y. Kuo and C.-Y. Lai, "Refined belief propagation decoding of sparse-graph quantum codes," accepted by *IEEE J. Sel. Areas Inf. Theory*, 2020. (DOI:10.1109/JSAIT.2020.3011758)

J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci.*, vol. 81, pp. 3088–3092, 1984.

S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1139–1147.

N. N. Schraudolph, "A fast, compact approximation of the exponential function," *Neural Comput.*, vol. 11, no. 4, pp. 853–862, 1999.