POSTER TITLE

Rate Adjustable Bivariate Bicycle Codes for Quantum Error Correction

POSTER AUTHORS

Ming Wang: mwang42@ncsu.edu
Frank Mueller (main contact): fmuelle@ncsu.edu

POSTER ABSTRACT

This work (1) proposes a novel numerical algorithm to accelerate the search process for good Bivariate Bicycle (BB) codes and (2) defines a new variant of BB codes suitable for quantum error correction. In contrast to vanilla BB codes, where parameters remain unknown prior to code discovery, the rate of the proposed code can be determined before the search by specifying a factor polynomial. A number of new BB codes found by this algorithm are reported. In particular, by using the proposed construction of BB codes, we found a number of surprisingly short to medium-length codes that were previously unknown.

POSTER RELEVANCE

- Quantum computing
- Quantum error-correction and mitigation

Rate Adjustable Bivariate Bicycle Codes for Quantum Error Correction

Ming Wang, Frank Mueller
Department of Computer Science
North Carolina State University
Raleigh, NC 27695, USA
Email: {mwang42,fmuelle}@ncsu.edu

***Abstract*—This work (1) proposes a novel numerical algorithm to accelerate the search process for good Bivariate Bicycle (BB) codes and (2) defines a new variant of BB codes suitable for quantum error correction. In contrast to vanilla BB codes, where parameters remain unknown prior to code discovery, the rate of the proposed code can be determined before the search by specifying a factor polynomial. A number of new BB codes found by this algorithm are reported. In particular, by using the proposed construction of BB codes, we found a number of surprisingly short to medium-length codes that were previously unknown.**

A. Introduction: Quantum Error Correction (QEC) is the cornerstone of advancing from the current Noisy Intermediate-Scale Quantum (NISQ) era to the era of fault-tolerant quantum computing. Among the various QEC codes, quantum Low-Density Parity-Check (qLDPC) codes stand out due to their lower-weight stabilizers, which require fewer gate operations. Unlike surface codes, which also feature low-weight stabilizers, qLDPC codes support more logical qubits. In particular, one type of qLDPC code, BB codes [1], known for their high threshold and low overhead, have received much attention recently.

In this work, we proposed a numerical algorithm to search for good BB codes. In addition, a new construction of BB codes is proposed that allows us to customize the code rate *before* performing a search, much in contrast to prior search techniques that identified the rate only after returning a new code as a search result.

B. Preliminaries: Let S_m and I_m be the circulant permutation matrix and identity matrix, respectively, of size m. Furthermore, let $x = S_l \otimes I_m$, and $y = I_l \otimes S_m$. It is easy to verify that $xy = yx$. The BB codes can be defined by two polynomials, $a(x, y)$ and $b(x, y)$, where each monomial can be expressed as a matrix. Thus, the polynomials $a(x, y)$ and $b(x, y)$ have a natural matrix representation, A and B, respectively. In [1], the authors restricted the polynomials as follows:

$$\begin{aligned} a(x,y) &= x^a + y^b + y^c \\ b(x,y) &= y^d + x^e + x^f. \end{aligned} \tag{1}$$

Each polynomial has 3 terms and can be written as $A = A_1 + A_2 + A_3$ and $B = B_1 + B_2 + B_3$ in matrix form. Besides, $A^T = A_1^T + A_2^T + A_3^T = A_1^{-1} + A_2^{-1} + A_3^{-1}$ as A_i is the power of x or y, which are permutation matrices. It is easy to see that the weight of the stabilizers, i.e., the row weight of parity-check matrices, is 6. In the rest of the paper, we will focus on codes with row-weight 6 because they are easier to implement in hardware.

C. Numerical Acceleration of Searches: In this section, we will introduce a technique to numerically perform an exhaustive search for BB codes under certain constraints to reduce the search space. First, we want to exclude equivalent codes. It is easy to prove that these four codes

$$\begin{aligned} C_1 &: H_X = [A|B], H_Z = [B^T|A^T] \\ C_2 &: H_X = [A^T|B^T], H_Z = [B|A] \\ C_3 &: H_X = [B|A], H_Z = [A^T|B^T] \\ C_4 &: H_X = [B^T|A^T], H_Z = [A|B] \end{aligned} \tag{2}$$

have the same parameters. Thus, we can reduce the search space to 1/4 by ignoring codes with polynomials like C_2, C_3, C_4. We further note that the two codes $C_1 : H_X = [A|B], H_Z = [B^T|A^T]$ and $C_5 : H_X = [A^T|B], H_Z = [B^T|A]$ do not always have the same parameters. For example, when $l = 6, m = 12$, the code constructed by $a(x, y) = x^4 + y^2 + y^6$ and $b(x, y) = y^5 + x^3 + x^4$ is a $[[144, 8, 10]]$ code, whereas the code constructed by $a(x, y) = x^2 + y^6 + y^{10}$ and $b(x, y) = y^5 + x^3 + x^4$ is a $[[144, 8, 8]]$ code.

In [1], the authors used BP-OSD [2] algorithms to estimate the distance of codes during searches. To accelerate this process, we use two thresholds, τ_k and τ_d, to discard bad codes. Any code with $k < \tau_k$ or the estimated distance $\hat{d} < \tau_d$ will be discarded immediately without further investigation. Besides finding known prior code, some of the prior unknown codes we found using this algorithm are listed in Table I.

D. Coprime-BB Codes: Based on the commutativity of matrices x and y, one can choose different polynomial forms and construct valid CSS codes. But we need to perform an extensive search to find codes with good k, d using polynomials of the shape given in Eq. (1). Here, we propose the coprime-BB codes that can provide codes for a pre-determined k.

Let l, m be two coprime numbers. As $x^l = y^m = I$, it is easy to verify that $\langle xy \rangle$ generates a cyclic group, and any monomial $\{x^i y^j | 0 \le i < l, 0 \le j < m\}$ can be expressed as a power of xy. Thus, let $\pi = xy$, any polynomial in $\mathbb{F}_2[x, y]/(x^l + 1, y^m + 1)$ can be expressed in $\mathbb{F}_2[\pi]/(\pi^{lm} + 1)$.

Algorithm 1: An algorithm to search BB codes

Data: l, m, τ_k, τ_d
Result: codes of parameters $[[2lm, k, \hat{d}]]$
Generate all polynomial pairs of the specified form
 $L \leftarrow [(a_1(x,y), b_1(x,y)), ...]$;
Remove codes with the same parameters:
 $L' \leftarrow$ `remove_equivalent`(L);
for $i \leftarrow 1$ **to** $|L'|$ **do**
  if `is_connected`$(a_i(x,y), b_i(x,y))$ **then**
    H_X, H_Z = `BB_matrices`$(a_i(x,y), b_i(x,y))$;
    $k \leftarrow 2lm - 2$`rank`(H_X);
    if $k < \tau_k$ **then**
      continue ;
    else
      $\hat{d} \leftarrow$ `distance_bound`(H_X, H_Z, τ_d);
    end
  else
    continue ;
  end
end

TABLE I
NOVEL CODES FOUND BY ALGORITHM 1

l	m	$a(x,y)$	$b(x,y)$	$[[n,k,d]]$
7	7	$x^3+y^5+y^6$	$y^2+x^3+x^5$	$[[98,6,12]]$
3	21	$1+y^2+y^{10}$	y^3+x+x^2	$[[126,8,10]]$
5	15	$1+y^6+y^8$	y^5+x+x^4	$[[150,16,8]]$
3	27	$1+y^{10}+y^{14}$	$y^{12}+x+x^2$	$[[162,8,14]]$
6	15	x^3+y+y^2	$y^6+x^4+x^5$	$[[180,8,16]]$

Let $g(\pi) = \mathrm{GCD}(a(\pi), b(\pi), \pi^{lm}+1)$, then the code defined by $a(\pi)$ and $b(\pi)$ has $k = 2\deg g(\pi)$.

The proof is similar to Proposition 1 in [3]. Given $\mathrm{colsp}(H_X) = \{H_X\boldsymbol{x} | \boldsymbol{x} \in \mathbb{F}_2^{2lm}\} = \{A\boldsymbol{u} + B\boldsymbol{v} | \boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}_2^{lm}\}$, the column space can be expressed as polynomials, $\mathrm{colsp}(H_X) = \{a(\pi)u(\pi) + b(\pi)v(\pi) | u(\pi), v(\pi) \in \mathbb{F}_2[\pi]/(\pi^{lm}+1)\}$. Since $\mathbb{F}_2[\pi]/(\pi^{lm}+1)$ is a univariate polynomial ring, $a(\pi)\mathbb{F}_2[\pi]/(\pi^{lm}+1)$ and $b(\pi)\mathbb{F}_2[\pi]/(\pi^{lm}+1)$ are principal ideals. Thus, $\mathrm{colsp}(H_X) = \{a(\pi)u(\pi) + b(\pi)v(\pi) | u(\pi), v(\pi) \in \mathbb{F}_2[\pi]/(\pi^{lm}+1)\}$ is a principal ideal generated by $g(\pi)$ and $\mathrm{rank}(H_X) = \dim \mathrm{colsp}(H_X) = lm - \deg g(\pi)$. Thus, the number of logical qubits $k = 2lm - \mathrm{rank2}(H_X) = 2lm - 2(lm - \deg g(\pi)) = 2\deg g(\pi)$.

Using the proposed algorithm 2, we find a number of interesting coprime-BB codes shown in Table II.

E. Conclusion/Future Work: We developed an algorithm for fast numerical searches for the discovery of BB codes. Furthermore, we proposed a novel construction of BB codes that enables us to set the rate before constructing them. Simulations should be done in order to compare the error rates of the newly found codes and to assess how well these codes map onto architectural constraints of existing quantum device technologies.

Algorithm 2: An algorithm to search coprime-BB codes.

Data: $l, m, \tau_d, p(\pi)$; `/* ` $p(\pi)$ ` is a factor of ` $\pi^{lm}+1$ ` */`
Result: codes of parameters $[[2lm, k, \hat{d}]]$
$C \leftarrow$ all polynomials $f(\pi)$ in $\mathbb{F}_2[\pi]/(\pi^{lm}+1)$ s.t.
 $\mathrm{wt}(f(\pi)) = 3$;
$C' \leftarrow$ all polynomials $c(\pi)$ in C s.t. $c(\pi)$
 $\mathrm{mod}\ p(\pi) = 0$;
$L \leftarrow$ all polynomial pairs $(a(\pi), b(\pi))$ in C' s.t.
 $\mathrm{GCD}(a(\pi), b(\pi)) = p(\pi)$;
$L' \leftarrow$ `remove_equivalent`(L);
for $i \leftarrow 1$ **to** $|L'|$ **do**
  if `is_connected`$(a_i(x,y), b_i(x,y))$ **then**
    H_X, H_Z = `BB_matrices`$(a_i(x,y), b_i(x,y))$;
    $k \leftarrow 2lm - 2$`rank`(H_X);
    $\hat{d} \leftarrow$ `distance_bound`(H_X, H_Z, τ_d);
  else
    continue ;
  end
end

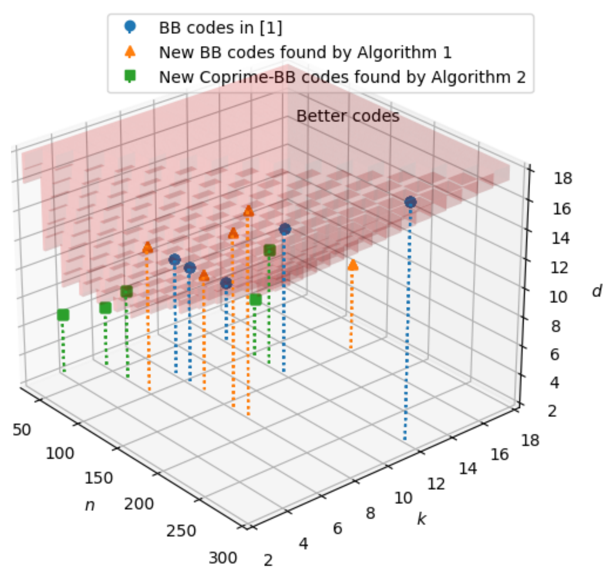


Fig. 1. The visualization of newly found codes and codes previously found in [1].

TABLE II
SOME CODES FOUND BY ALGORITHM 2

l	m	$a(\pi)$	$b(\pi)$	$[[n,k,d]]$
3	5	$1+\pi+\pi^2$	$\pi+\pi^3+\pi^8$	$[[30,4,6]]$
3	7	$1+\pi^2+\pi^3$	$\pi+\pi^3+\pi^{11}$	$[[42,6,6]]$
5	7	$1+\pi+\pi^5$	$1+\pi+\pi^{12}$	$[[70,6,8]]$
2	27	$\pi^2+\pi^5+\pi^{44}$	$\pi^8+\pi^{14}+\pi^{47}$	$[[108,12,6]]$
7	9	$1+\pi+\pi^{58}$	$\pi^3+\pi^{16}+\pi^{44}$	$[[126,12,10]]$

REFERENCES

[1] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, “High-threshold and low-overhead fault-tolerant quantum memory,” *Nature*, vol. 627, no. 8005, p. 778–782, Mar. 2024.
[2] 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, p. 043423, Dec 2020.
[3] P. Panteleev and G. Kalachev, “Degenerate quantum LDPC codes with good finite length performance,” *Quantum*, vol. 5, p. 585, 2021.

NC STATE UNIVERSITY

Rate Adjustable Bivariate Bicycle Codes for Quantum Error Correction

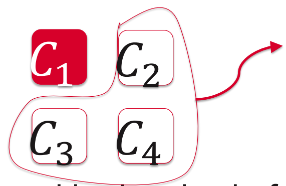
Ming Wang, Frank Mueller
Department of Computer Science, North Carolina State University,
Raleigh, NC, 27606

Motivation

- Find BB codes efficiently
- Design BB codes with specific parameters

Approach

- **Fast Search**
  - ✓ Reduce search space



  - ✓ Discard bad codes before knowing their tight distance bound

Algorithm 1: An algorithm to search BB codes

Data: l, m, τ_k, τ_d
Result: codes of parameters $[[2lm, k, \hat{d}]]$
Generate all polynomial pairs of the specified form
$L \leftarrow [(a_1(x,y), b_1(x,y)), ...]$;
Remove codes with the same parameters:
$L' \leftarrow \texttt{remove\_equivalent}(L)$;
for $i \leftarrow 1$ **to** $|L'|$ **do**
  if $\texttt{is\_connected}(a_i(x,y), b_i(x,y))$ **then**
    $H_X, H_Z = \texttt{BB\_matrices}(a_i(x,y), b_i(x,y))$;
    $k \leftarrow 2lm - 2\text{rank}(H_X)$;
    if $k < \tau_k$ **then**
      continue ;
    else
      $\hat{d} \leftarrow \texttt{distance\_bound}(H_X, H_Z, \tau_d)$;
    end
  else
    continue ;
  end
end

- **Coprime-BB codes:**
  - ✓ Require coprime l, m
  - ✓ Selecting polynomial $p(\pi)$ to get BB codes with determined k

Algorithm 2: An algorithm to search BB codes with the new form of polynomials.

Data: $l, m, \tau_d, p(\pi)$; /* $p(\pi)$ is a factor of $\pi^{lm}+1$ */
Result: codes of parameters $[[2lm, k, \hat{d}]]$
$C \leftarrow$ all polynomials $f(\pi)$ in $\mathbb{F}_2[\pi]/(\pi^{lm}+1)$ s.t. $\text{wt}(f(\pi)) = 3$;
$C' \leftarrow$ all polynomials $c(\pi)$ in C s.t. $c(\pi) \bmod p(\pi) = 0$;
$L \leftarrow$ all polynomial pairs $(a(\pi), b(\pi))$ in C' s.t. $\text{GCD}(a(\pi), b(\pi)) = p(\pi)$;
$L' \leftarrow \texttt{remove\_equivalent}(L)$;

* The rest are the same as **Algorithm 1**

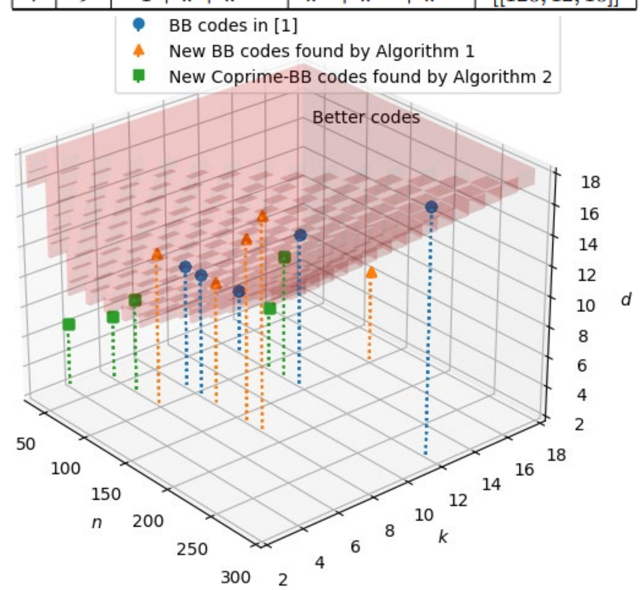
Preliminary Results

- BB Codes found by fast search algorithm:

l	m	$a(x,y)$	$b(x,y)$	$[[n,k,d]]$
7	7	$x^3+y^5+y^6$	$y^2+x^3+x^5$	$[[98,6,12]]$
3	21	$1+y^2+y^{10}$	y^3+x+x^2	$[[126,8,10]]$
5	15	$1+y^6+y^8$	y^5+x+x^4	$[[150,16,8]]$
3	27	$1+y^{10}+y^{14}$	$y^{12}+x+x^2$	$[[162,8,14]]$
6	15	x^3+y+y^2	$y^6+x^4+x^5$	$[[180,8,16]]$

- Coprime-BB Codes :

l	m	$a(\pi)$	$b(\pi)$	$[[n,k,d]]$
3	5	$1+\pi+\pi^2$	$\pi+\pi^3+\pi^8$	$[[30,4,6]]$
3	7	$1+\pi^2+\pi^3$	$\pi+\pi^3+\pi^{11}$	$[[42,6,6]]$
5	7	$1+\pi+\pi^5$	$1+\pi+\pi^{12}$	$[[70,6,8]]$
2	27	$\pi^2+\pi^5+\pi^{44}$	$\pi^8+\pi^{14}+\pi^{47}$	$[[108,12,6]]$
7	9	$1+\pi+\pi^{58}$	$\pi^3+\pi^{16}+\pi^{44}$	$[[126,12,10]]$



Conclusion

- More good BB codes are found.
- New construction provides more flexibility on code rates

Future Work

- Compare the error rates of newfound codes
- Explore how to implement these codes on actual hardware

References

[1] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, “High-threshold and low-overhead fault-tolerant quantum memory,” Nature, vol. 627, no. 8005, p. 778–782, Mar. 2024.

[2] 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, p. 043423, Dec 2020.

[3] P. Panteleev and G. Kalachev, “Degenerate quantum LDPC codes with good finite length performance,” Quantum, vol. 5, p. 585, 2021