Report of QEC course project

Problem: Use a [n,1,n] repetition code (call A the associated PCM) and a [n,n-1,2] SPC code (with B as its PCM). Note that A and B are orthogonal matrices (so they can form a CSS code). We then consider the asymmetric product code given by:

$$H_x = [A \bigotimes A]$$

$$H_z = [I \bigotimes B; B \bigotimes I]$$

This is a quantum code with n^2 physical qubits and 0 logical qubits. And with pure distances $\delta_X = 4$ and $\delta_Z = n$. In order to achieve a positive encoding rate, we can eliminate a few of the X stabilisers. Systematic ways of doing this shall be analysed in order to achieve a good minimum distance properties, and further generalised to d-fold product constructions.

---

1 Introduction

We first start saying what's obvious and comes to mind at the very first sight.

1.1 What's obvious

1.1.1 Appearance of the H_x's Taking a brief look at the structure of H_x's might not be the worst idea:

$$H_x^{(2)} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

,

$$H_x^{(3)} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

,

$$H_x^{(4)} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

,

$$H_x^{(5)} = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}$$

1.1.2 Augmenting PCM's According to [2], augmenting is "When augmenting a code C we add codewords to C". The point is when increasing the rate of the CSS code discussed in this project, our trade off is playing with the boundaries of minimum distance. d_x doesn't really have much space to do any trade-off ($d_x \geq 4 = \delta_x$), but $d_z \geq \delta_z = n$ has plenty. Therefore augmentation better happens for $H_x^{(n)}$. First, we have to translate what it means to add codewords in terms of PCM's. Say our generator matrix looks like this in the systematic form before adding new codewords:

$$G = \left[I_{k\times k}; P_{k\times(n-k)} \right]$$

After adding the codewords and making the f=generator systematic again, our generator looks like this:

$$G^* = \left[I_{(k+1)\times(k+1)}; P^*_{(k+1)\times(n-k-1)} \right]$$

For H, this process means:

$$H = \left[P^T_{(n-k)\times k}; I_{(n-k)\times(n-k)} \right]_{(n-k)\times n}$$

and:

$$H^* = \left[P^{(*T)}_{(n-k-1)\times(k+1)}; I_{(n-k-1)\times(n-k-1)} \right]_{(n-k-1)\times n}$$

or, to put it simply, deleting checknodes or rows from the parity check matrices.

1.1.3 Can augmenting preserve the CSS commutativity condition?

This is how our PCM's look like normally:

$$H_x = \begin{bmatrix} h_x^1 \\ h_x^2 \\ \dots \\ h_x^{(N-K)} \end{bmatrix}$$

$$(H_z)^T = \left[(h_z^1)^T; (h_z^2)^T; \dots; (h_z^{(N-K)})^T \right]$$

The condition for a CSS code:

$$h_x^i.(h_z^j)^T = 0, \quad \forall i, j$$

After augmentation, for some i's the h_i^x is a row of 0's. Therefore, the above condition is satisfied automatically. Thus, the required CSS commutativity condition is even better satisfied by deleting some of the H_x's rows.

1.1.4 Information before any analysis for $n < \infty$

.	$[[n, k, \delta_z]]$	$[w_x^{c,n}, w_x^{r,n}]$
$n = 2$	$[[4, 3, 2]]$	[{1},{4}]
$n = 3$	$[[9, 5, 3]]$	[{{1,2,1},{2,4,2},{1,2,1}},{4}]
$n = 4$	$[[16, 7, 4]]$	[{{1,2,2,1},{2,4,4,2},{2,4,4,2},{1,2,2,1}},{4}]
$n = 5$	$[[25, 9, 5]]$	[{{1,2,2,2,1},{2,4,4,4,2},{2,4,4,4,2},{2,4,4,4,2},{1,2,2,2,1}},{4}]

1.1.5 Obvious information from the looks of PCM's From the weight of the columns and rows of the PCM, and basic information from Error Correction theory, it's obvious that for $H_x^{(n)}$ deleting the first and last $n-1$ series of rows can't be that wise. We will be deleting information from qubits that aren't incorporated in many parity check equations. So we expect to severely reduce the true minimum distance (d_z) if we remove rows from the aforementioned series of rows. Same is true for rows number k such that $mod(k, n-1) = 1$. As it deletes too much information from two qubits. Due to the specific structure of the PCM, the way we look at the structure of the PCM is the following.

1.1.6 Row Structure of the PCM We refer to each row with two indices (i, j). Where $0 \leq i, j \leq n-2$, and if k represents the row number for any $H_x^{(n)}$ (we know that $0 \leq k < (n-1)^2$), $k = i.(n-1) + j$.

1.2 What's not obvious

What's not obvious is how deleting the rows affects the minimum distances. Well, probably a man with more intuition would have done it differently. But I'm a man who thinks intuition can only be gained through careful analysis. So, basically what I did was to remove every combination of rows from the computationally possible PCM's, like for $n = 3, 4, 5, 6, 7$. And I looked to find any pattern that might help. I don't think going through 0's and 1's would have helped at all in the first place. So in the next section I will remove rows from the PCMs which are computationally comprehensible.

2 Almost all possible row pattern removals

Probably before removing any rows, one might be interested to find minimum distance for the case where no row is removed for $H_x^{(n)}$. $c \in \mathcal{F}_2^{n^2}$ is a minimum distance codeword iff $H_x^{(n)} c = 0$ and $c \notin span\{(H_z^{(n)})^T\}$. But with removing no rows, only all zero codeword is transmittable. So,

$$d_z^{(n)} = \infty, \quad \forall n$$

2.1 Tables for single row removals

In order to check for the distance being true and not pure, I wrote the following MATLAB code, and apparently $rank(.)$ function alone doesn't give correct outputs for binary matrices:

```
function flag = isNotInSpan(Hz,c,n)
c = c(:);
```

```
    H_test = gf([gf(Hz,2);c'],2);
    r = rank(H_test);
    if r > 2*n-1
        flag =  true;
    else
         flag = false;
    end
end
```

Listing 1: checking for c not in span space

Consider $\mathcal{R}_r(.) : \mathcal{F}_2^{(n-1)^2 \times n^2} \to \mathcal{F}_2^{[(n-1)^2 - r] \times n}$ to be the operator that takes a PCM and removes r rows and gives out another PCM. Take $d_z(H)$ to be the minimum distance of some PCM, H. Then obviously:

$$d_z^r := \max_{H' \in \mathcal{R}_r(H)} d_z(H')$$

is a decreasing function of r, the number of row removals. So d_z^1 is an upperbound for true distances of a specific family of PCM's, $H_x^{(n)}$. Now, the purpose of this section is to compute true distances of $\mathcal{R}_1(H_x^{(n)})$ family for each row removal as n changes from 3 to 6 or 7. Depending on how computationally easy it is for us to compute.

2.1.1 $n = 3$

removed row	d_z
1	1
2	1
3	1
4	1

Therefore,

$$d_z^1(n = 3) = 1$$

2.1.2 $n = 4$

removed row	d_z

1	1
2	2
3	1
4	2
5	4
6	2
7	1
8	2
9	1

Therefore,

$$d_z^1(n = 4) = 4$$

2.1.3 $n = 5$

removed row	d_z
1	1
2	2
3	2
4	1
5	2
6	4
7	4
8	2

9	2
10	4
11	4
12	2
13	1
14	2
15	2
16	1

Therefore,

$$d_z^1(n=5)=4$$

2.1.4 $n=6$

removed row	d_z
1	1
2	2
3	3
4	2
5	1
6	2
7	4
8	6
9	4

10	2
11	3
12	6
13	9 (Computationally time consuming)
14	6
15	3
16	2
17	4
18	6
19	4
20	2
21	1
22	2
23	3
24	2
25	1

Therefore,

$$d_z^1(n=6)=9$$

2.1.5 $n = 7$

removed row	d_z
1	1

2	2
3	3
4	3
5	2
6	1
7	2
8	4
9	6
10	6
11	4
12	2
13	3
14	6
15	9 (Need 25GB RAM, predicted)
16	9 (Need 25GB RAM, predicted)
17	6
18	3
19	3
20	6
21	9 (Need 25GB RAM, predicted)
22	9 (Need 25GB RAM, predicted)
23	6
24	3

25	2
26	4
27	6
28	6
29	4
30	2
31	1
32	2
33	3
34	3
35	2
36	1

Therefore,

$$d_z^1(n = 7) = 9$$

2.1.6 INDUCTIONS AND DEDUCTIONS FOR $r = 1$ As it appears from the tables above for $n \leq 7$,

$$d_z^1(n) = \lfloor \frac{n}{2} \rfloor^2$$

We take a look at the codeword for such minimum distance and see if it possesses some pattern to be able to extend the conclusion to any $n > 7$. However, there's still another problem. Even if we prove that there exists some c_n with weight above, we have only proved an upperbound for $d_z^1(n) \leq \lfloor \frac{n}{2} \rfloor^2$. To prove that it is equal to this upperbound, we must prove that no c_n exists with weight less than this upperbound that satisfies minimum distance requirements.

2.1.7 Patterns for c_n

Let n be an even number for the time being. We know that $c_n \in \mathcal{F}_2^{n^2}$. Then:

$$c_n(i) = \begin{cases} 0 & \text{if } i \le \frac{n}{2}, \\ 1 & \text{if } mod(i,n) > \frac{n}{2} \text{or } mod(i,n) = 0. \\ 0 & o.w. \end{cases}$$

For example for $n = 4, 6$, this looks like the following:

$$c_4 = [e_{11} + e_{12}] + [e_{15} + e_{16}]$$

$$c_6 = [e_{22} + e_{23} + e_{24}] + [e_{28} + e_{29} + e_{30}] + [e_{34} + e_{35} + e_{36}]$$

It turns out that this works for $n = 3, 4, ..., 100$. A similar pattern works for odd n too. By doing a simple induction-like reasoning one can assert that:

Lemma 1 *Let $d_z^1(n) = \max_{H' \in \mathcal{R}_1(H)} d_z(H_x^{(n)})$ be the minimum distance defined, then one can say that,*

$$d_z^1(n) \le \lfloor \frac{n}{2} \rfloor^2, \quad \forall n \ge 3$$

Also, as stated earlier,

Lemma 2 *Let $d_z^r(n) = \max_{H' \in \mathcal{R}_r(H)} d_z(H_x^{(n)})$ be the minimum distance defined, then,*

$$d_z^1(n) \ge d_z^r(n), \quad \forall r \ge 2 \implies$$

by previous lemma,

$$d_z^r(n) \le \lfloor \frac{n}{2} \rfloor^2, \quad \forall r \ge 1$$

To prove the equality for $r = 1$, we must make sure that no c_n with $weight(c_n) < \lfloor \frac{n}{2} \rfloor^2$ can be the minimum distance codeword for single removals. At the moment, I don't have in mind such proof. So, for the rest of this document we only assume such conjecture.

2.2 Tables for multiple row removals

Based on the data of previous tables, removing row (i, j), where $j = 1$, or $i = 1$, or $i = n - 1$ isn't in our area of interest. As the d_z is either 1 or 2. So, the maximal d_z for multiple row removals can't exceed these values. Another obvious lemma,

Lemma 3 *Let $d_{z,R}^r(n)$, where $R = \{$set of removed rows from $H_x^{(n)}$ PCM$\} \in \mathcal{P}(\{1, 2, ..., (n-1)^2\})$ ($\mathcal{P}(.)$ being the power set) and $d_{z,\{s\}}^1(n)$ be the minimum distance from a single removal of row s, and $|R| = r$, Then,*

$$d_{z,R}^r(n) \le \min_{s \in R} d_{z,\{s\}}^1(n), \quad \forall n \ge 3$$

So, we're mostly interested in **row pattern removings including rows that their single removal produces the highest minimum distance**. Again, we try to derive patterns by doing multiple removals for each n that is computationally possible. For $n = 3$, any removals result in $d_z^2 = 1$. So we start from $n = 4$.

2.2.1 $n = 4$

removed rows $(k, l, ...) \xrightarrow{mod(.,n-1)} (i, j, ...)$	d_z
$(2, 4) \rightarrow (2, 1)$	2
$(2, 5) \rightarrow (2, 2)$	2
$(2, 6) \rightarrow (2, 0)$	2
$(2, 8) \rightarrow (2, 2)$	2

2.2.2 $n = 5$ In this case, rows $(6, 7, 10, 11)$ have the highest minimum distance single row removal:

removed rows $(k, l, ...) \xrightarrow{mod(.,n-1)} (i, j, ...)$	d_z
$(6, 7) \rightarrow (2, 3)$	2
$(6, 11) \rightarrow (2, 3)$	4
$(7, 10) \rightarrow (3, 2)$	4
$(2, 5, 6, 7, 8, 10, 14) \rightarrow (2, 1, 2, 3, 0, 2, 2)$	2

2.2.3 $n = 6$ In this case, rows $(3, 7, 8, 9, 11, 12, 13, 14, 15, ...)$ have the highest minimum distance single row removal:

removed rows $(k, l, ...) \xrightarrow{mod(.,n-1)} (i, j, ...)$	d_z
$(12, 14) \rightarrow (2, 4)$	6

$(8, 12, 14) \rightarrow (3, 2, 4)$	4
$(8, 18) \rightarrow (3, 3)$	6
$(8, 13, 18) \rightarrow (3, 3, 3)$	3
$(7, 9, 13, 17, 19) \rightarrow (2, 4, 3, 2, 4)$	4
$(7, 8) \rightarrow (2, 3)$	2
$(12, 13) \rightarrow (2, 3)$	3
$(2, 3) \rightarrow (2, 3)$	1

2.2.4 $n = 7$

removed rows $(k, l, \ldots) \xrightarrow{mod(.,n-1)} (i, j, \ldots)$	d_z
$(9, 10) \rightarrow (3, 4)$	2
$(3, 4) \rightarrow (3, 4)$	1
$(15, 16) \rightarrow (3, 4)$	3
$(3, 10) \rightarrow (3, 4)$	3
$(4, 8) \rightarrow (4, 2)$	3
$(8, 15) \rightarrow (2, 3)$	4
$(11, 14) \rightarrow (5, 2)$	4
$(3, 17) \rightarrow (3, 5)$	3
$(8, 11) \rightarrow (2, 5)$	4
$(9, 17, 20, 28) \rightarrow (3, 5, 2, 4)$	6
$(8, 10, 15, 17, 20, 22, 27, 29) \rightarrow (2, 4, 3, 5, 2, 4, 3, 5)$	4

2.3 SYMMETRIES

There are certain symmetries due to the symmetric properties of $H_x^{(n)}$.

Let (i, j) be a certain row that is a candidate of removal. Where if k is the overall row number $j = mod(k, n-1)$ and i being the index showing the location of the row among the $n-1$ series of each $n-1$ rows. Consider these $n-1$ series of rows. Due to the symmetry of the Parity Check Matrix, and the table of single removals which obviously asserts the following: The minimum distance remains unchanged if we remove a certain row or the mirror of this row with respect to the center row. Also blocks of rows that are mirrored with respect to each other have the same effect when removed.Due to this double symmetry, information of only $\frac{1}{4}$th of the single row removal has all of the information of the table within.

Also, for any $H_x^{(n)}$, the table of row removals for a certain row has a pattern. If $i = 1$, the minimum distance is as $\{1, 2, ..., \lfloor \frac{n-1}{n} \rfloor, \lfloor \frac{n-1}{n} \rfloor - 1, ..., 2, 1\}$. If $i = 2$, the pattern is a symmetric multiples of 2, and in general, the patern is multiples of i, for ith block of rows until the middle block. Then it goes back to 1.

2.4 RULES OF REMOVALS

Rules of thumb up to now that seem to be true:

Removing two adjacent rows in a single block results into the least minimum distance for single removals in that block.

Let $r = (i, j)$ and $r' = (i', j')$ be two rows. If $mod(j - j', n-1) \neq 0$, and $i \neq i'$, then $d^2_{z,\{r,r'\}} = \min\{d^1_{z,r}, d^1_{z,r}\}$.

2.5 EXTENSION TO $n = 8$

For single removals in the case of $n = 8$, we have big time problems of computations. The order of computations is of the order $10^2 3$. The method of computation here is **Simulated Annealing**. The algorithm was derived from [1]. It's described in 1.

Algorithm 1 Simulated Annealing for minimum Distance

$X = X_0$ inital state
$T = T_0$ inital temperature
REPEAT
REPEAT
$X_{NEW} = PERTURB(X)$
$\Delta f = f(X_{NEW}) - f(X)$
if ($\Delta f \leq 0$) or (random(0,1) < $\exp(-\frac{\Delta f}{T})$)) **then**
  $X = X_{NEW}$
end if
$T = UPDATE(T)$
Until $T \leq T_f$

For single row removal, it's expected to have $d_z^1(8) = 16$. The algorithm converges when I look for codewords with 16 ones, but it doesn't converge for 13, 14 or 15 ones. Obviously, this alone can't prove that the minimum distance is actually 16 for single row removal. But it's good to have some level of confidence in the propositions. The computations for $n = 8$ are cumbersome. Based on the rules I assume that are true, I will predict a table for $n = 8$. Later on, using the SA algorithm I will try to justify it. My predictions are as what follows.

2.5.1 SINGLE REMOVALS OF $n = 8$

removed row	d_z
1	1
2	2
3	3
4	4
5	3
6	2
7	1

8	2
9	4
10	6
11	8
12	6
13	4
14	2
15	3
16	6
17	9
18	12
19	9
20	6
21	3
22	4
23	8
24	12
25	16
26	12
27	8
28	4
29	3
30	6

31	9
32	12
33	9
34	6
35	3
36	2
37	4
38	6
39	8
40	6
41	4
42	2
43	1
44	2
45	3
46	4
47	3
48	2
49	1

2.5.2 Predicted multi row removal, $n = 8$ ...TO BE COMPLETED... AWAITING NUMERICAL CONFIRMATIONS

removed rows $(k, l, \dots) \xrightarrow{mod(.,n-1)} (i, j, \dots)$	d_z
$(12, 14) \to (2, 4)$	6
$(8, 12, 14) \to (3, 2, 4)$	4
$(8, 18) \to (3, 3)$	6
$(8, 13, 18) \to (3, 3, 3)$	3
$(7, 9, 13, 17, 19) \to (2, 4, 3, 2, 4)$	6
$(7, 8) \to (2, 3)$	2
$(12, 13) \to (2, 3)$	3
$(2, 3) \to (2, 3)$	1

3 Notable Conclusion

The following lemma is only a proposition without proof based on observations made so far. Let n be large enough and even. The $\frac{n}{2}$th block of rows single-removal table looks like this:

removed row	d_z
$(\frac{n}{2} - 1)(n - 1) + 1$	$\frac{n}{2}$
$(\frac{n}{2} - 1)(n - 1) + 2$	n
$(\frac{n}{2} - 1)(n - 1) + 3$	$\frac{3n}{2}$
...	...
...	...
$\lfloor \frac{(n-1)^2}{2} \rfloor + 1$	$\frac{n^2}{4}$
...	...
...	...

Conjecture 1 *Let $d^r_{z,R}(n)$, where R = {set of removed rows from $H_x^{(n)}$ PCM} $\subset \mathcal{P}(\{1,2,...,(n-1)^2\})$ and $d^1_{z,\{s\}}(n)$ be the minimum distance from a single removal of row s, and $|R| = r$. If we define $d^r_z(n) = \max_{R\subset \mathcal{N}} d^r_{z,R}(n)$, then if,*

$$d^r_z(n) = O(n^2) = (\lfloor \frac{n}{2} \rfloor - 1)^2$$

then there exists a pattern of removal R,

$$r = O(1)$$

, and if,*

$$d^r_z(n) = O(n\log(n))$$

there exists a pattern or removal, R where,

$$O(n) \leq r \leq O(n\log(n))$$

It's important to note that the example of the rules introduced in section 2.4 is as following, for $n = 8$. If we remove rows $\{17, 19, 25, 31, 33\}$, (almost 5 rows for any n) the minimum distance will be 9, $(\frac{n}{2} - 1)^2$, order $O(n^2)$. In $mod(., 8 - 1)$, the rows are $\{3, 5, 4, 3, 5\}$. The pattern obviously states that it's not recommended to remove rows with the same mod consecutively. The error correction reason is that consecutive same *mod* removing is like removing the decoding information of a qubit, because the structure of the PCM preserves the distance of ones for consecutive rows with same *mod*. Therefore, the suggestion is to avoid removing the information from shared qubits. Same is true for the other rules in that section. Removing the decoding information extensively reduces minimum distance.

Conjecture 2 *So for any general n even, removing the rows , in $mod(., n-1)$, $\{\frac{n}{2}-1, \frac{n}{2}+1, \frac{n}{2}, \frac{n}{2}-1, \frac{n}{2}+1\}$ results into minimum distance $(\frac{n}{2} - 1)^2$.*

The order of removals is obviously $O(1)$. This is a more specific version of part 1 of conjecture 1. I think I may be able to prove this mathematically, as it's not very difficult, but the second part of conjecture 1 requires rigour and more investigation.

References

[1] ZHANG MUXIANG and MA FULONG. "Simulated annealing approach to the minimum distance of error-correcting codes". In: *International Journal of Electronics* 76.3 (1994), pp. 377–384. DOI: 10.1080/00207219408925934. eprint: https://doi.org/10.1080/00207219408925934. URL: https://doi.org/10.1080/00207219408925934.

[2] S.A. Ladhake Ravindra M. Deshmukh. "Analysis of Various Puncturing Patterns and Code Rates: Turbo Code". In: *International Journal of Electronic Engineering Research* 1.2 (2009), pp. 79–88.