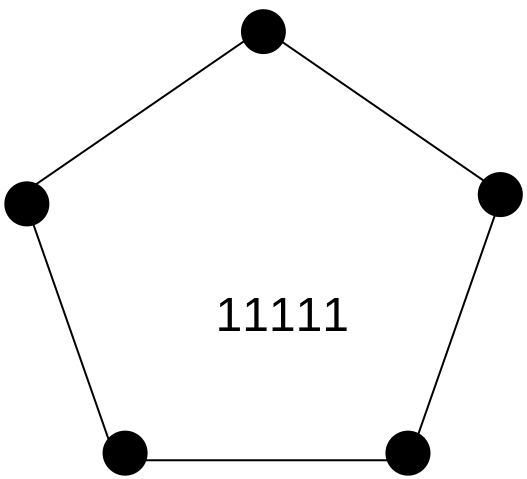
Codeword stabilized quantum codes

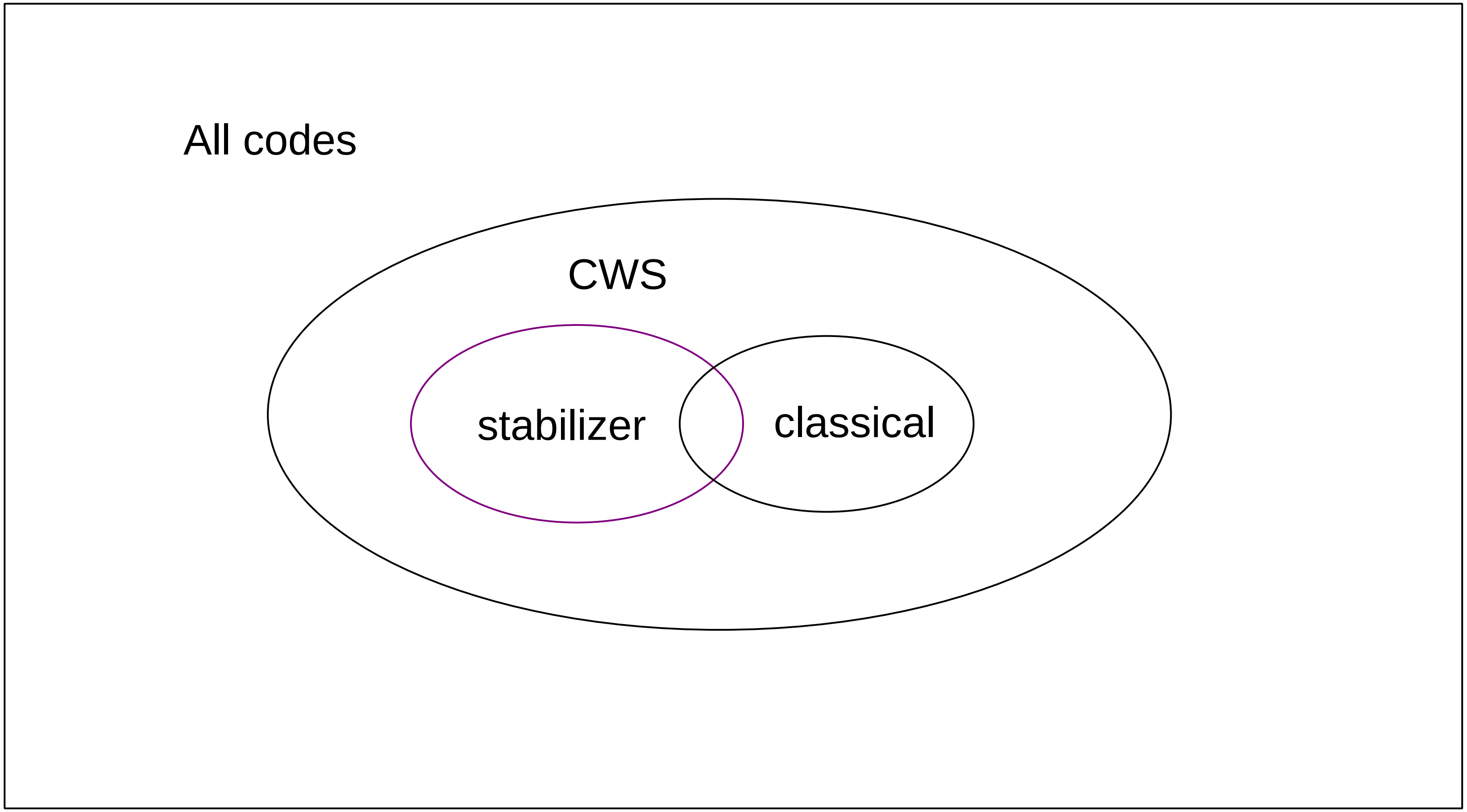
Andrew Cross, Graeme Smith, John A. Smolin, and Bei Zeng

CWS codes for short



quant-ph/0708.1021

Venn Diagram



Some notation

[[n,k,d]] additive quantum code: n raw qubits, k protected qubits, distance d

((n,K,d)) quantum code: n raw qubits, K protected states, distance d

For an additive code $K=2^k$

((9,12,3))

Our work was inspired by the ((9,12,3)) code of
Yu, Chen, Lai and Oh quant-ph/0704.2122.

This was the first nonadditive quantum code with distance > 2 that outperforms any known additive code (and even any *possible* additive code).

Stabilizers

We consider only the Pauli group

Pauli group: I,X,Z,Y and tensor products written like $XIZZIYZ$

Stabilizer for an [[n,k,d]] code is a set of commuting members of this group generated by $S_1 \ldots S_{n-k}$

There are also logical operators $X_1 \ldots X_k$ and $Z_1 \ldots Z_k$

These also commute with the stabilizers

A *stabilizer state* is an [[n,0,d]] code, *i.e.* it is the +1 eigenstate of a maximal abelian subgroup of the Paulis having n generators and no logical operators.

CWS codes are characterized by two objects:

$$\text{Stabilizer state } \langle S_1 \ldots S_n \rangle : \quad |S\rangle$$

and

$$\text{Set of codeword operators } w_1 \ldots w_K$$

$$\text{The codewords are } w_l|S\rangle$$

$$\text{These are all eigenstates of all } s \in S: \; sw_l|S\rangle = \pm w_l|S\rangle$$

Difference : Normal stabilizer code would have all $+1$ eigenvalues for all codewords

Instead each codeword has its own stabilizer : $w_l S w_l^\dagger$

Standard form CWS codes are defined by two objects

$$\text{Graph state } \langle S_1 \ldots S_n \rangle : \quad |G\rangle$$

$$\text{and}$$

$$\text{Classical code } C \text{ with codewords } c_1 \ldots c_K$$

$$\text{Word operators are } w_l = Z^{c_l}$$

$$\text{Codewords are } Z^{c_l}|G\rangle$$

$$C \text{ must correct } Cl_S(\mathcal{E}) \;\; \text{classical induced errors}$$

The stabilizer tells you which errors the classical code must correct

Error detection conditions (general)

For a general code with basis vectors $|\psi_i\rangle$
to detect errors from a set $\mathcal{E}$:

$$\langle\psi_i|E|\psi_j\rangle = c_E\delta_{ij}$$

for all $E \in \mathcal{E}$

for codewords of the form $|w_l\rangle = w_l|S\rangle$

$$\langle S|w_i^\dagger E w_j|S\rangle = c_E\delta_{ij}$$

Error detection conditions

$$w_i^\dagger E w_j \notin \pm S \;\; i \neq j$$

Codewords should not be confused

$$\text{and}$$

$$(\forall_i w_i^\dagger E w_i \notin \pm S) \text{ or}$$

Error should be detected

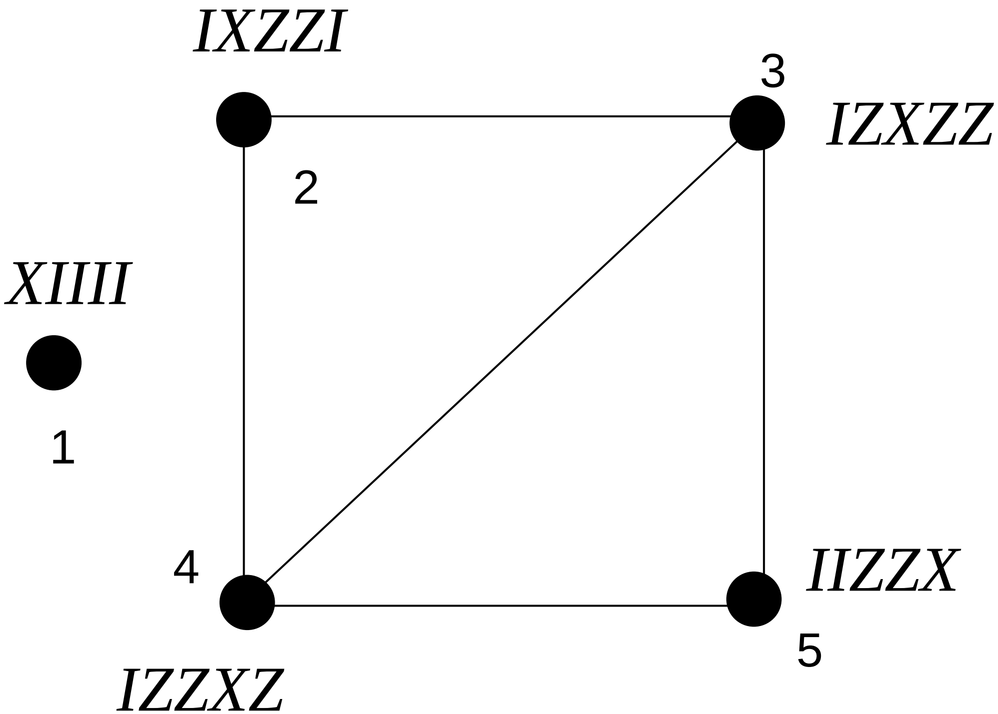
$$(\forall_i w_i^\dagger E w_i \in S) \text{ or}$$

$$(\forall_i w_i^\dagger E w_i \in -S)$$

Last two codewords are “immune”
to the error---*degeneracy* condition

Graph States

Graph states are stabilizer states which have stabilizer generators each with a single X and Z's on the nodes to which they're connected.



Standard Form

Theorem: Any codeword stabilized code is locally equivalent to one with a graph state stabilizer and word operators consisting only of Z's and including the identity. We call this *standard form*.

Proof ingredients:

1. any stabilizer state is locally clifford equivalent to a graph state. (proved elsewhere)
   basically row-reduction
2. This results in new codeword operators, still products of Paulis.
3. Any X's in the new codewords can be eliminated by multiplying by stabilizer elements from the graph state. Since these each have a single X, this is straightforward.

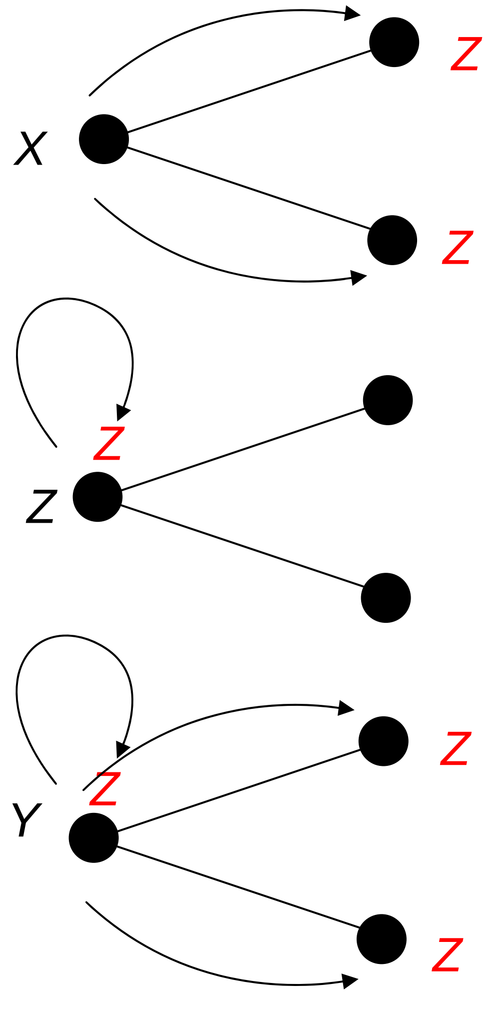
X-Z rule (lemma)

On a graph state X errors are equivalent to (possibly multiple) Z errors. We call these the *induced errors*.

$$X_i w_i |S\rangle = X_i w_i S_i |S\rangle$$
$$= \pm X_i S_i w_i |S\rangle$$

S_i has only one X on bit i so the X's cancel

X-Z rule



Y=XZ

Errors detection conditions (standard form)

Since all induced errors are *Z's,* things are essentially classical

$$\text{Given } S \text{ and } C$$

$$\text{Error detection conditions are :}$$

$$C \text{ must detect errors from } Cl_S(\mathcal{E}) \;\; \text{(the indueced errors)}$$

$$\text{and}$$

$$Cl_S(E) \neq 0$$

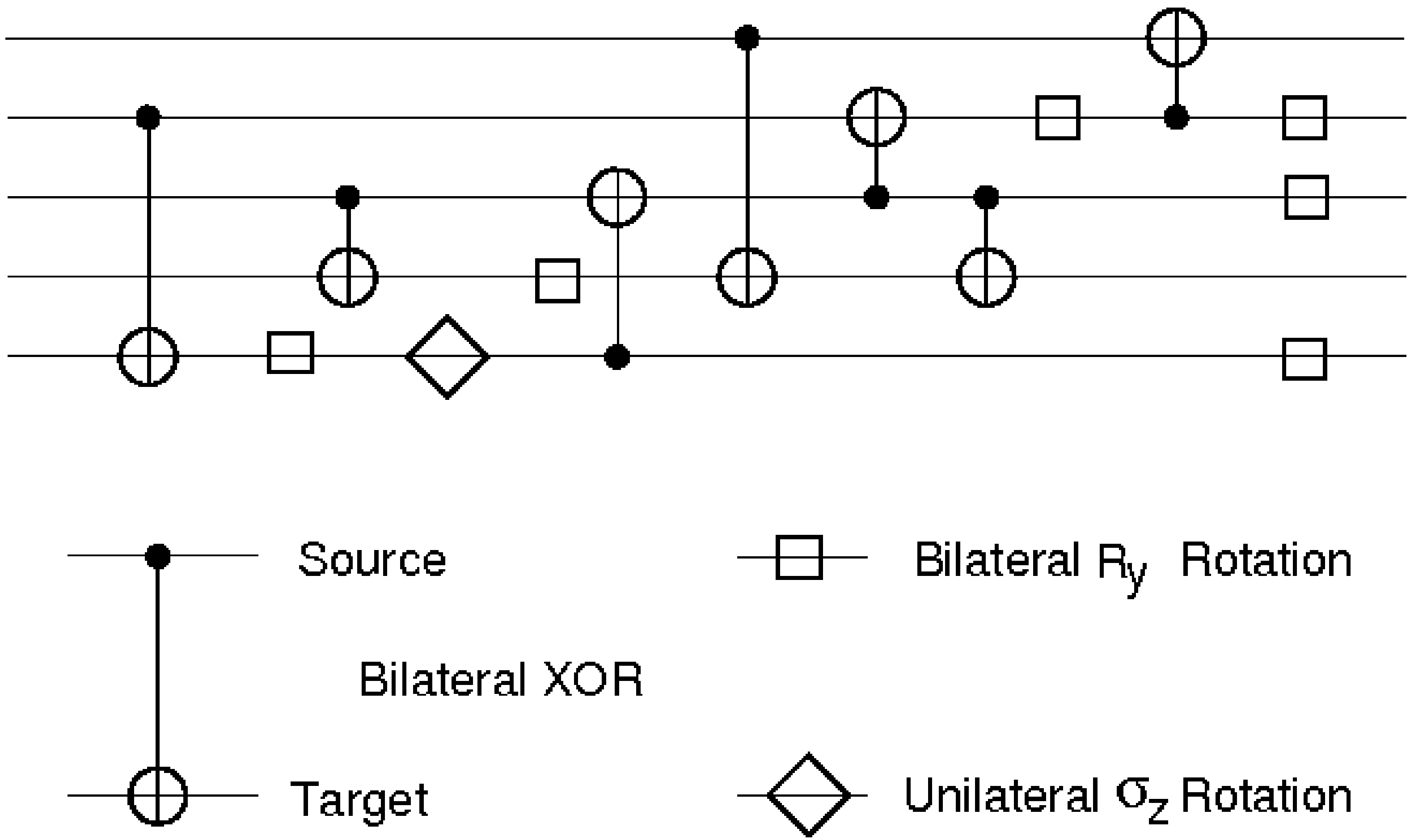
$$\text{or } \forall_i [w_i = Z^{c_i}, E] = 0$$ (degeneracy)

Relation to stabilizer codes

$$\text{Stabilizer } \langle S_1 \dots S_{n-k} \rangle$$
$$\text{with logical operators } \langle \bar{X}_1 \dots \bar{X}_k \rangle, \ \langle \bar{Z}_1 \dots \bar{Z}_k \rangle$$
$$\text{then}$$
$$S = \langle S_1 \dots S_{n-k}, \bar{X}_1 \dots \bar{X}_k \rangle \text{ and}$$
$$\text{word operators } \langle \bar{Z}_1 \dots \bar{Z}_k \rangle$$
$$\text{define a CWS code}$$

Furthermore, whenever the word operators form a group, a CWS code IS a stabilizer code

Antideluvian [[5,1,3]] code



BDSW96 'the big paper'

code also in LMPZ96

[[5,1,3]] codewords

$$\begin{aligned}|v_0\rangle = \; & -|00000\rangle - |11000\rangle - |01100\rangle - |00110\rangle - |00011\rangle \\ & -|10001\rangle + |10010\rangle + |10100\rangle + |01001\rangle + |01010\rangle \\ & +|00101\rangle + |11110\rangle + |11101\rangle + |11011\rangle + |10111\rangle + |01111\rangle\end{aligned}$$

$$\begin{aligned}|v_1\rangle = \; & -|11111\rangle - |00111\rangle - |10011\rangle - |11001\rangle - |11100\rangle \\ & -|01110\rangle + |01101\rangle + |01011\rangle + |10110\rangle + |10101\rangle \\ & +|11010\rangle + |00001\rangle + |00010\rangle + |00100\rangle + |01000\rangle + |10000\rangle\end{aligned}$$

all parity 0 string with some collection of signs, and the same with 0 and 1 interchanged

[[5,1,3]] Stabilizer

XZZXI
IXZZX
XIXZZ
ZXIXZ

Generators of the stabilizer

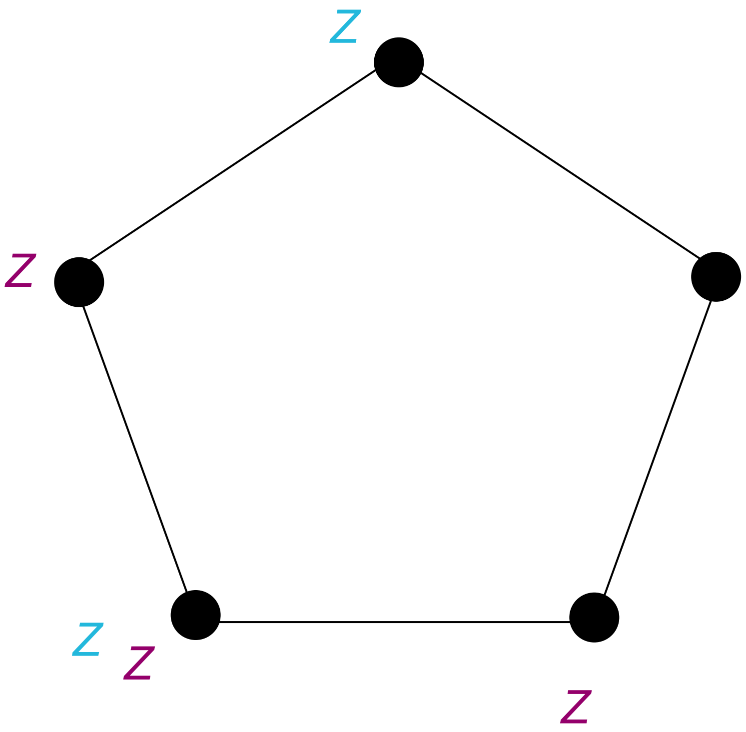
XXXXX logical X
ZZZZZ logical Z

Can be made into a CWS code by adding in XXXX to the stabilizer, and using 00000 and ZZZZZ as the codeword operators

Illuminating?

On a ring

To correct single errors, need to detect double errors



these cancel

If the codewords are 00000 and 11111 a nondetectable error would have to be weight 5

The X-Z rule tells us all single errors on the ring are weight 1, 2, or 3

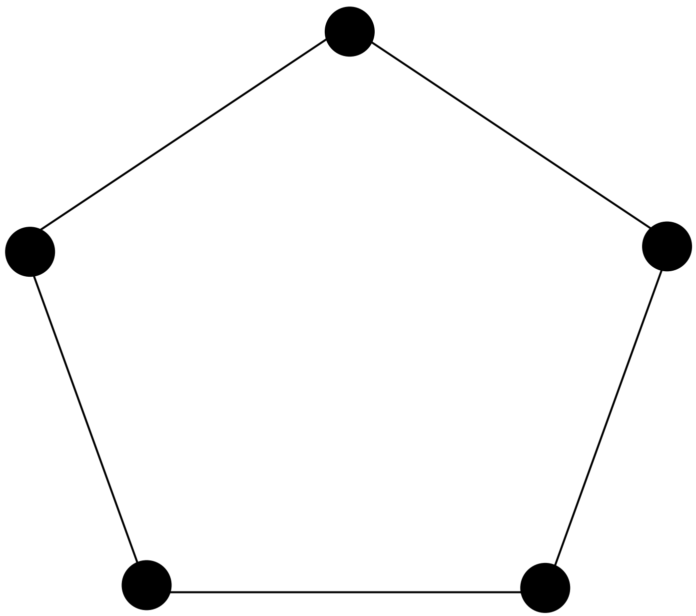
We need one weight 3 and one weight 2

But the only weight 2 errors are nonadjacent

((5,6,2)) Rains, Harden, Shor, Sloane code

“The symmetries discussed above generate a group of order 640. There is an additional symmetry which can be described as follows:
First, permute the columns as k->k^3, that is exchange qubits 2 and 3.
Next, for each qubit negate one of the Pauli matrices and exchange the other two, where the Pauli matrices negated are Z, Y, X, X, Y, respectively.
This increases the size of the symmetry group to 3840. This group acts as the permutation group S5 on the qubits. This is the full group of symmetries of the code. That is, the full subgroup of the semidirect product of S5 and $PSU2^5$ that preserves the code [10].”

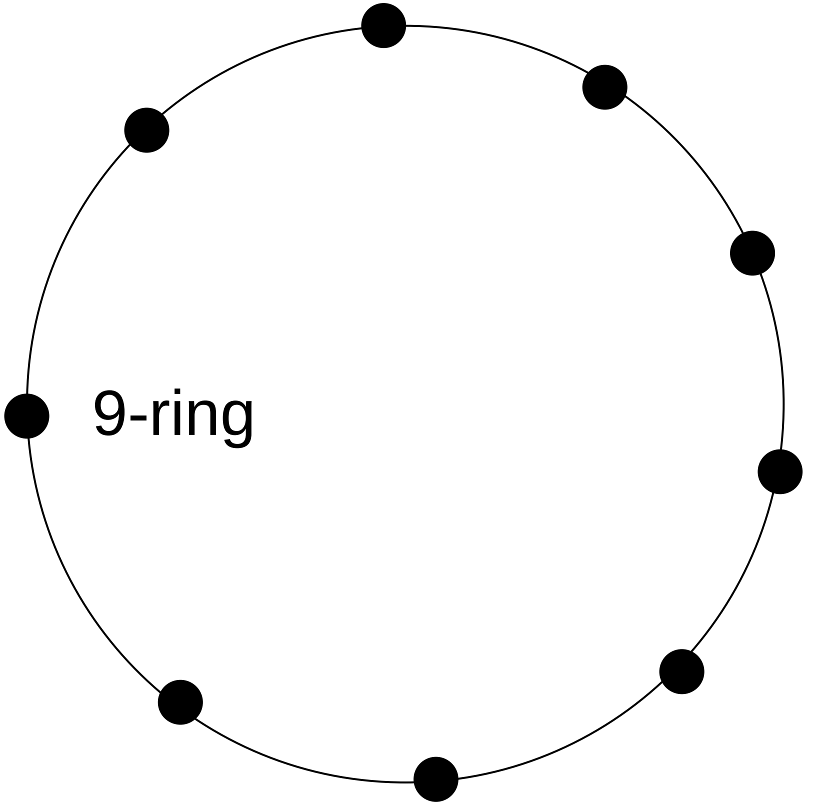
((5,6,2)) code



00000
11010
01101
10110
01011
10101

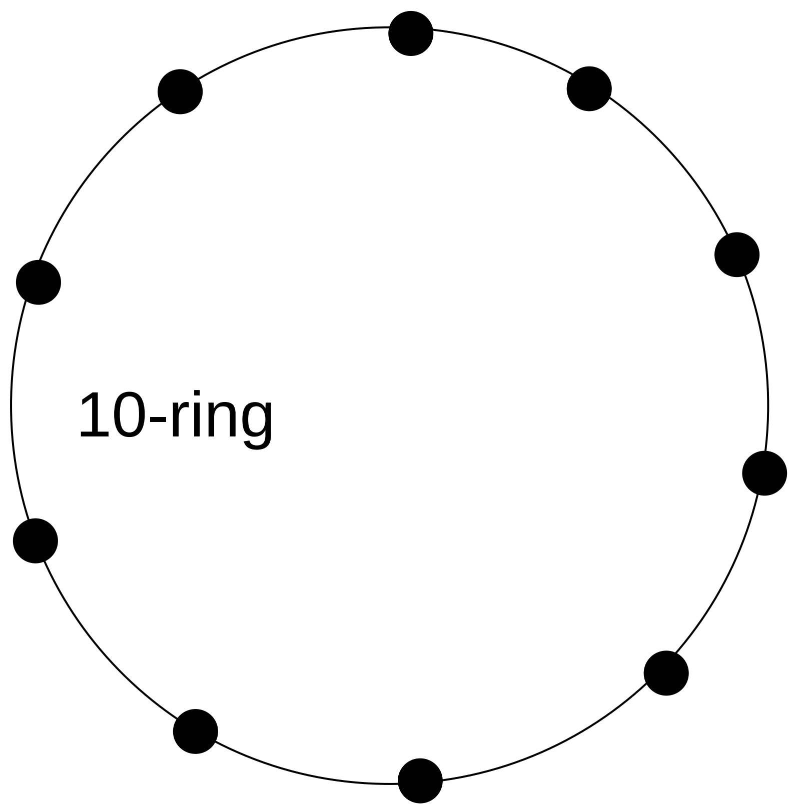
Since weight 3 induced errors are adjacent, the weight can't change by 3 so none of these can be transformed into 00000.
Since weight 2 errors are non-adjacent, they can't be transformed amongst each other.

((9,12,3))



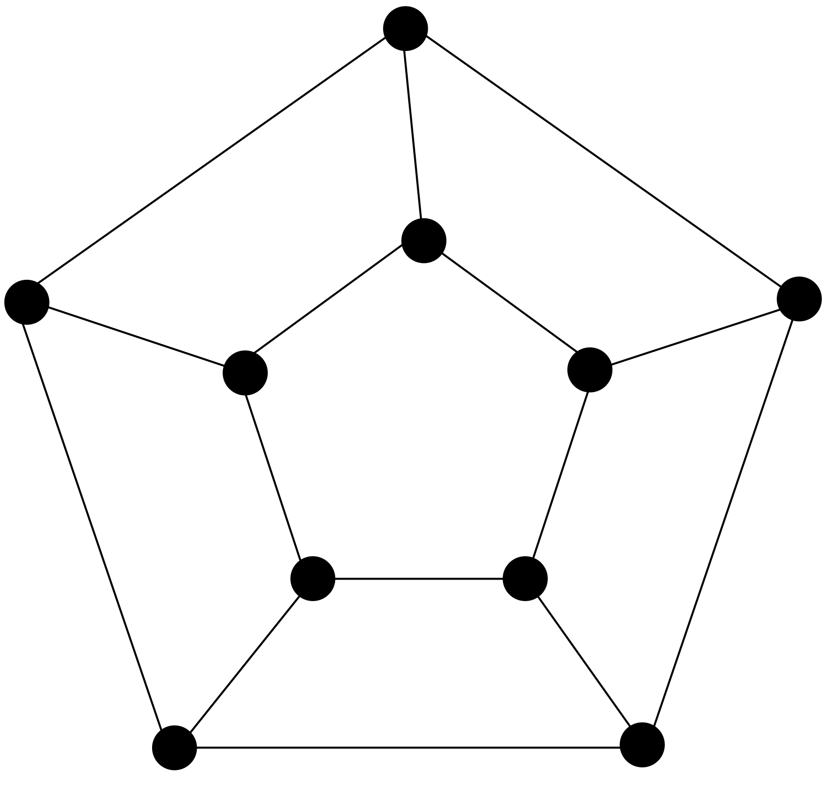
((10,18,3))

If rings are so great,
why not a bigger one?



Linear programming bound is ((10,24,3))

((10,20,3))



If one ring is good,
two must be better

Linear programming bound is ((10,24,3))

Big search problem

For all graphs of size n, search for best classical code of a given distance.

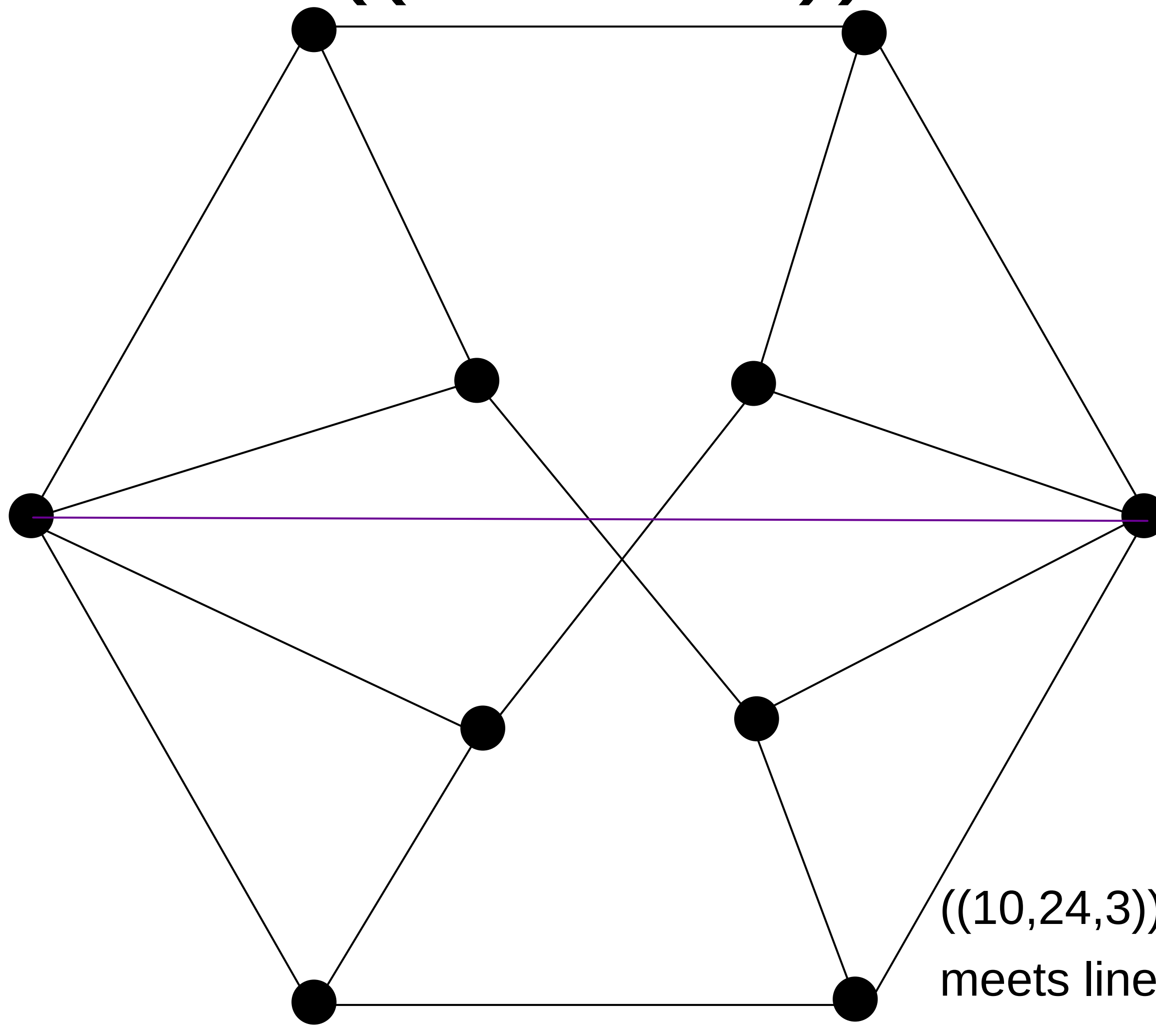
Super-exponential

Turns out to be quite doable for n up to 10 or maybe 11

((10,24,3)) code meeting linar programming bound:

Yu, Chen and Oh, 0709.1780

((10,24,3))



((10,24,3)) code is unique and meets linear programming bound

Almost the simplest nontrivial nonembeddable in 3D graph where edges represent orthogonality conditions

Encoding circuits (Bei's slide)

$$U_{(G,C)} = GC$$

Classicl Encoder C: $C\left|i\right\rangle = \left|\mathbf{c}_i\right\rangle = X^{\mathbf{c}_i}\left|00...0\right\rangle$

Gragh Encoder G: $G\left|00..0\right\rangle = \left|G\right\rangle$ $\quad G = \prod_{(j,k\in E)} P_{(j,k)} H^{\otimes n}$

$$GC\left|i\right\rangle = G\left|\mathbf{c}_i\right\rangle = \prod_{(j,k\in E)} P_{(j,k)} H^{\otimes n} X^{\mathbf{c}_i}\left|00...0\right\rangle$$

$$= \prod_{(j,k\in E)} P_{(j,k)} Z^{\mathbf{c}_i} H^{\otimes n}\left|00...0\right\rangle = Z^{\mathbf{c}_i} \prod_{(j,k\in E)} P_{(j,k)} H^{\otimes n}\left|00...0\right\rangle$$

$$= Z^{\mathbf{c}_i}\left|G\right\rangle$$

Future Work

Find more codes!

Particularly of higher distance

Generalize to higher dimensions

Looi, Yu, Gheorghiu and Griffiths 0712.1979

Understand strange classical error models

Codeword Stabilized Quantum Codes

Isaac Chuang[1], Andrew Cross[1,2], Graeme Smith[2], John Smolin[2], and Bei Zeng[1]

Massachusetts Institute of Technology

[1]Massachusetts Institute of Technology, Cambridge, MA 02139, USA; [2]IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

IBM

Abstract -- *We present a unifying approach to quantum error correcting code design that encompasses additive (stabilizer) codes, as well as all known examples of nonaddtive codes with good parameters. We consider an algorithm which maps the search for quantum codes in our framework to a problem of identifying maximum cliques in a graph. We use this framework to generate new codes with superior parameters to any previous known. In particular, we find ((10,18,3)) and ((10,20,3)) codes.*

Motivation & Idea

Quantum error correction codes play a central role in quantum computation and quantum information. While considerable understanding has now been obtained for a broad class of quantum codes, almost all of this has focused on stabilizer codes, the quantum analogues of classical additive codes. However, such codes are strictly suboptimal in some *settings---there exist nonadditive codes which encode a larger logical space than possible with a stabilizer code of the same length and capable of tolerating the same number of errors*. There are only a handful of such examples, and their constructions have proceeded in an ad hoc fashion, each code working for seemingly different reasons.

We present a unifying approach to quantum error correcting code design, namely, the codeword stabilized (CWS) quantum codes, that encompasses additive (stabilizer) codes, as well as all known examples of nonadditive codes with good parameters.

The IDEA: Building a quantum code from classical codes, analogy to CSS codes and Stabilizer (additive, ADD) codes.

Quantum Code	Classical Codes
CSS	Two binary linear codes C_1 and C_2, $C_2 \subset C_1$
ADD	An additive code over GF(4)
CWS	1. A self-dual additive code over GF(4) (a stabilizer state) 2. A binary code

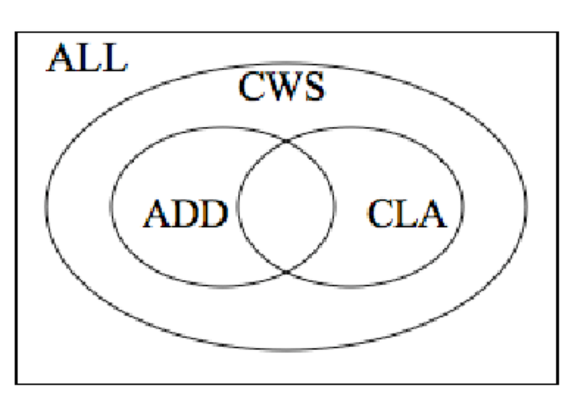
CWS Q = Graph G + Binary Code C

$$Q = (G, C)$$

1. The self-dual code over GF(4) can be taken to describe a graph state G, which transforms the quantum errors to be corrected into effectively classical errors.
2. A classical binary (linear or nonlinear) code C capable of correcting the induced classical error model.

With a fixed graph G, finding a quantum code is reduced to finding a classical code that corrects the (perhaps rather exotic) induced error model.

In the special case where the graph is unconnected, the binary classical code coincides with the usual binary classical code where independent error happens with equal probability on each bit.



The relationship of CWS codes with additive quantum codes and classical codes: ALL: all quantum codes; CWS: CWS codes; ADD: additive codes; CLA: classical codes.

ADD Q ~~ Linear C

1. If C is linear, then Q=(G,C) is additive
2. If Q is additive, then there exist a linear C and a graph G such that Q=(G,C)
3. If C is nonlinear, we cannot tell whether Q=(G,C) is additive or nonadditive

General Construction

Given G and C, our CWS code is spanned by basis vectors

$$|w_l\rangle = w_l|G\rangle$$

where the graph state |G> is stabilized by the stabilizer S

$$S_l = X_l Z^{\mathbf{r}_l}$$

where $\mathbf{r}_l$ is the lth row of the graph's adjacency matrix, and

$$w_l = Z^{\mathbf{c}_l}$$

are Pauli operators defined by the classical codeword

$$\mathbf{c}_l \in C$$

The quantum error correction criterion

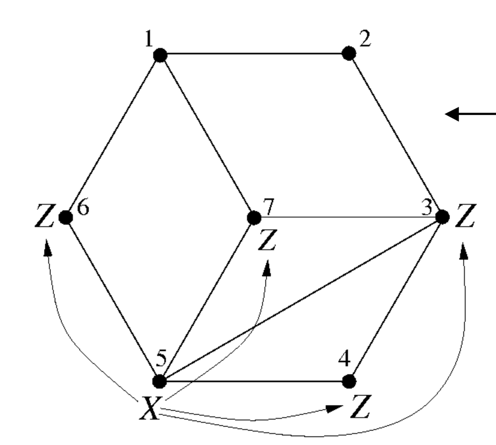
$$\langle G|w_i^+ E w_j|G\rangle = c_E \delta_{ij}$$

This criterion leads to a necessary and sufficient condition for detecting errors in $\mathcal{E}$ that for all $E \in \mathcal{E}$

$\forall i \neq j \; w_i^+ E w_j \notin \pm S$, and $\forall i \; w_i^+ E w_i \notin \pm S$ or $\forall i \; w_i^+ E w_i \in S$ or $\forall i \; w_i^+ E w_i \in -S$

Due to this condition, all errors become Zs, we can think the error model as classical, albeit consisting of strange multibit errors. We define this translation to classical errors by the function

$$Cl_G(E = \pm Z^{\mathbf{v}} X^{\mathbf{u}}) = \mathbf{v} \bigoplus_{l=1}^{n} (\mathbf{u})_l \mathbf{r}_l$$



Example of the induced error on a graph state: The state has stabilizer generators XZIIIZZ, ZXZIIII, ZXZIIZ, IIZXZII, IIZZXZZ, ZIIIZXI, and ZIZIZIX. An X error applied to node 5 in the lower-left is translated by multiplying with the stabilizer element IIZZXZZ and turns into Z errors on the nodes indicated.

Main Theorem: A CWS code with stabilizer S and codeword operators C detects errors from $\mathcal{E}$ if and only if C detects errors from $Cl_G(\varepsilon)$ and in addition we have for each $E \in \mathcal{E}$

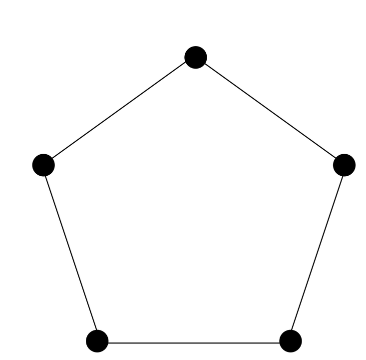
$$Cl_G(E) \neq 0 \text{ or } \forall i \; Z^{\mathbf{c}_i} E = E Z^{\mathbf{c}_i}$$

Thus, any CWS code is completely specified by a graph state stabilizer S and a classical code C.

Examples

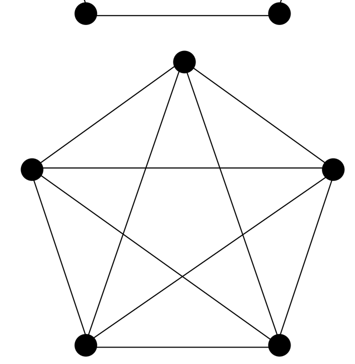
Known nonadditive codes with good parameters

((5,6,2)) in [3]



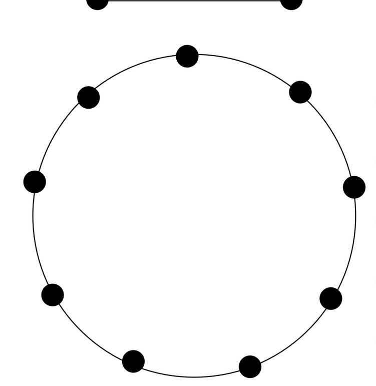
00000,11010,
01101,10110,
01011,10101.

((5,5,2)) in [4]



00000,11000,
10100,10010,
10001.

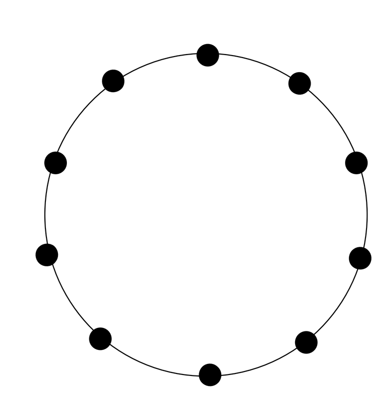
((9,12,3)) in [5]



000000000,100100100,
010001100,110101000,
000110001,100010101,
011001010,111101110,
001010011,101110111,
011111111,111011011.

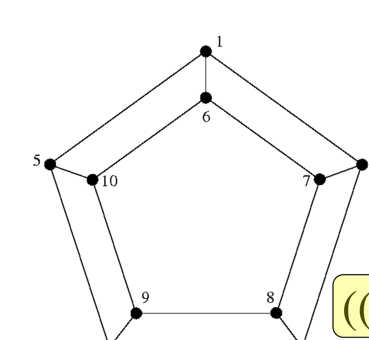
New Codes

((10,18,3)) ring code



0000000000,1101001100,
0011001010,0000011111,
0010001001,1111100000,
1000111110,1100100101,
0101101101,0001000110,
1010010010,0100110100,
1001010111,1011010001,
0110111000,0101110010,
1110100011,0111111011.

((10,20,3)) double - ring code



0000000000,1100101101,1100000100,
0010010010,1001100100,0111011011,
1101111110,0010111011,1001101111,
0111010000,1111000101,1011010100,
0101100000,1011011111,0101101011,
0011000001,0000101001,1110010110,
0001111010,1110111111.

The CWS-MAXCLIQUE Algorithm

Once the graph has been chosen, the problem of finding a quantum code is reduced to a search for classical codes. There are no handy systematical constructions in classical coding theory to deal with codes of such perhaps exotic error patterns induced by our approach. But we do know the algorithm of finding the best K for a given n,d has a natural encoding into the problem of MAXCLIQUE.

We present a MAXCLIQUE algorithm for finding our CWS quantum code for the largest possible dimension K for a given n,d, and G. This CWS-MAXCLIQUE algorithm (Algorithm 3) proceeds in three simple steps:

The first step finds the elements of $Cl_G(\varepsilon)$ and $D_G(\varepsilon)$. $D_G(\varepsilon)$ is a set of classical bit strings defined by

$$\{c \in \{0,1\}^n \| [Z^{\mathbf{c}}, E] \neq 0 \text{ for some } E \in \varepsilon \cap S\}$$

which is nonempty only if the CWS code (G,C) is degenerate.

Algorithm 1 Setup$(\mathcal{E}, \Lambda)$: Compute $Cl_{\mathcal{G}}(\mathcal{E})$ and $D_{\mathcal{G}}(\mathcal{E})$, where $\mathcal{E}$ is a set of Pauli errors and Λ is the adjacency matrix associated with graph $\mathcal{G}$.

Require: $\Lambda^T = \Lambda$, $\Lambda_{ij} = \{0,1\}$ and $\Lambda_{ii} = 0$
Ensure: $\mathrm{CL}[i] = \delta(\mathrm{String}(i) \in Cl_{\mathcal{G}}(\mathcal{E}))$ and $\mathrm{D}[i] = \delta(\mathrm{String}(i) \in D_{\mathcal{G}}(\mathcal{E}))$

The CWS-MAXCLIQUE algorithm for constructing the ((5,6,2)) code in [3]. Starting from the five qubit ring graph with adjacency matrix

$$\Lambda = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

The code is nondegenerate. The set CL[v]=0 is given by binary strings

00011,00110,01011,01100,
01101,01111,10001,10011,
10101,10110,10111,11000,
11010,11011,11110,11111

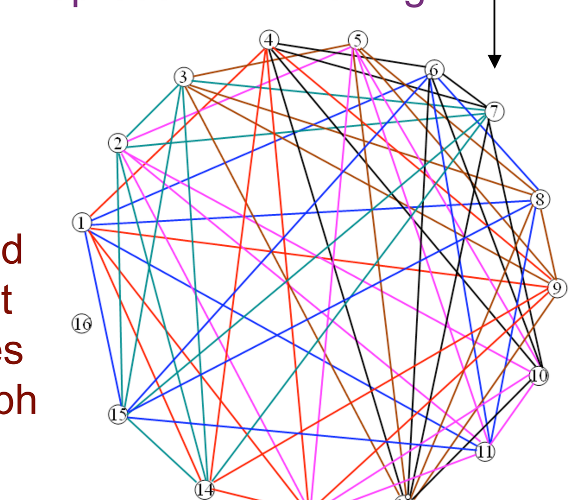
00000 is not included for it is in the code C by default.

The second step constructs the CWS clique graph whose vertices are classical codewords and whose edges indicate codewords that can be in the same classical code together. Vertices of the CWS clique graph are jointed by an edge if there is no error induced by the graph state that maps one codeword to the other.

Algorithm 2 MakeCWSCliqueGraph$(\mathrm{CL}, \mathrm{D})$: Construct a graph whose vertices V are classical codewords and whose edges E connect codewords that can belong to the same classical code, according to the error model indicated by $Cl_{\mathcal{G}}(\mathcal{E})$ and $D_{\mathcal{G}}(\mathcal{E})$.

Require: CL and D are binary arrays of length 2^n
Ensure: $0^n \in V$, $0^n \neq v \in V \Rightarrow \mathrm{D}[v] = 0$ and $\mathrm{CL}[v] = 0$, $(v,w) \in E \Rightarrow \mathrm{CL}[v \oplus w] = 0$

Make the CWS clique graph according to the set CL[v]=0 with 16 vertices, each vertex represents one string.



Lines of different colors correspond to different maxcliques of the graph of size 5.

The brown lines (3,5,8,9,13) → Codewords C: 00000,11010, 01101,10110, 01011,10101.

Finally, we call an external subroutine **findMaxClique**(V,E) that uses well-known techniques to find the maximum clique in the CWS clique graph. The clique-finding subroutine is not specified here because there are many exact and heuristic techniques for solving this classical NP -complete problem.

Algorithm 3 CWSMaxClique$(\mathcal{E}, \Lambda)$: Find a quantum code $\mathcal{Q}$ detecting errors in $\mathcal{E}$, and providing the largest possible dimension K for the given input. The input Λ specifies the adjacency matrix of the graph $\mathcal{G}$. The output $\mathcal{C}$ is a classical code such that $\mathcal{Q} = (\mathcal{G}, \mathcal{C})$ is a CWS code detecting errors in $\mathcal{E}$.

Require: $\Lambda^T = \Lambda$, $\Lambda_{ij} = \{0,1\}$ and $\Lambda_{ii} = 0 \; \forall i$
Ensure: $K = |\mathcal{C}|$ is as large as possible for the given input, $0^n \in \mathcal{C}$, and $\mathcal{C}$ satisfies the conditions in Theorem 3 of [1]

1: $(\mathrm{CL}, \mathrm{D}) \leftarrow$ **Setup**$(\mathcal{E}, \Lambda)$
2: $(V, E) \leftarrow$ **MakeCWSCliqueGraph**$(\mathrm{CL}, \mathrm{D})$
3: $\mathcal{C} \leftarrow$ **findMaxClique**(V, E)
4: **return** $\mathcal{C}$

References

[1] A. Cross, G. Smith, J. Smolin, and B. Zeng, *"Codeword stabilized quantum codes,"* arXiv:quant-ph/0708.1021, 2007.
[2] I. Chuang, A. Cross, G. Smith, J. Smolin, and B. Zeng, *"Codeword stabilized quantum codes: algorithm & structure,"* in preparation.
[3] E. M. Rains, R. H. Hardin, P. W. Shor, and N. J. A. Sloane, *"A nonadditive quantum code,"* Phys. Rev. Lett., vol. 79, p. 953, 1997.
[4] J. A. Smolin, G. Smith, and S. Wehner, *"A simple family of nonadditive quantum codes,"* Phys. Rev. Lett., vol. 99, p. 130505, 2007.
[5] S. Yu, Q. Chen, C. H. Lai, and C. H. Oh, *"Nonadditive quantum error-correcting code,"* arXiv:quant-ph/0704.2122, 2007.