Topological Subsystem Codes From Graphs and Hypergraphs

Pradeep Sarvepalli* and Kenneth R. Brown
Schools of Chemistry and Biochemistry; Computational Science and Engineering; and Physics, Georgia Institute of Technology, GA 30332
(Dated: June 30, 2012)

Topological subsystem codes were proposed by Bombin based on 3-face-colorable cubic graphs. Suchara, Bravyi and Terhal generalized this construction and proposed a method to construct topological subsystem codes using 3-valent hypergraphs that satisfy certain constraints. Finding such hypergraphs and computing their parameters however is a nontrivial task. We propose families of topological subsystem codes that were previously not known. In particular, our constructions give codes which cannot be derived from Bombin's construction. We also study the error recovery schemes for the proposed subsystem codes and give detailed schedules for the syndrome measurement that take advantage of the 2-locality of the gauge group. The study also leads to a new and general construction for color codes.

Keywords: quantum codes, topological quantum codes, subsystem codes, hypergraphs, decoding, color codes

I. INTRODUCTION

A major challenge in the theory of quantum error correcting codes is to design codes that are well suited for fault tolerant quantum computing. Such codes have many stringent requirements imposed on them, constraints that are usually not considered in the design of classical codes. An important metric that captures the suitability of a family of codes for fault tolerant quantum computing is the threshold of that family of codes. Informally, the threshold of a sequence of codes of increasing length is the maximum error rate that can be tolerated by the family as we increase the length of the codes in the sequence. The threshold is affected by numerous factors and there is no single parameter that we can optimize to design codes with high threshold. Furthermore, in the literature thresholds are reported under various assumptions. As the authors of [15] noted, there are three thresholds that are of interest: i) the code threshold which assumes there are no measurement errors, ii) the phenomenological threshold which incorporates to some extent the errors due to measurement errors, and iii) the circuit threshold which incorporates all errors due to gates and measurements. For a given family of codes, invariably the code threshold is the highest and the circuit threshold the lowest.

One of the nonidealities that affects the lowering of thresholds is the introduction of measurement errors. So codes which have same code thresholds, such as the toric codes and color codes, can end up with different circuit thresholds [15, 23]. At this point one can attempt to improve the circuit threshold by designing codes that have efficient recovery schemes and are more robust to measurement errors among other things. An important development in this direction has come in the form of subsystem codes, also called as operator error correcting codes [3, 11–14, 19]. By providing additional degrees of freedom subsystem codes allow us to design recovery schemes which are more robust to circuit nonidealities. That they can improve the threshold has already been reported in the literature [1].

A class of codes that have been found to be suitable for fault tolerant computing are the topological codes. These codes have local stabilizer generators, enabling the design of a local architecture for computing with them and also have the highest thresholds reported so far [21]. It is tempting to combine the benefits of these codes with the ideas of subsystem codes. This was first achieved in the work of Bombin [6], followed by Suchara, Bravyi and Terhal [22].

However, the code thresholds reported in [22] were lower than the thresholds of the toric codes and color codes. Nonetheless, this should not lead us to a hasty conclusion that the topological subsystem codes are not as good as the toric codes. There are at least two reasons why topological subsystem codes warrant further investigation. Firstly, the threshold reported in [22] is about 2% while [2] showed that the topological subsystem codes can have a threshold as high as 5.5%. This motivates the further study on decoding topological subsystem codes that are closer to their theoretical limits as well as the study of subsystem codes that have higher code thresholds.

The second point that must be borne in mind is the rather surprising lower circuit threshold of color code on the square octagon lattice as compared to the toric codes. Both of these codes have a code threshold of about 11%. But the circuit threshold of the color codes is about an order of magnitude lower than that of the toric codes. Both codes enable local architectures for fault tolerant quantum computing, both architectures realize gates by code deformation techniques, and both achieve universality in quantum computation through magic state distillation. Moreover, the color codes considered in [15] unlike the surface code can even realize the entire Clifford group transversally. Despite this apparent advantage over the toric codes, the color codes lose out to the surface codes in one crucial aspect—the weight of the check operators. Some of the check operators for the square octagon color code have a weight that is twice the weight of the check operators in the toric codes. Even though these higher weight check operators are approximately a fifth of the operators, they appear to be the dominant reason for the lower circuit threshold of the color codes.

The preceding discussion indicates that measurement errors can severely undermine the performance of a code with many

* pradeep.sarvepalli@gatech.edu

good properties including a good code threshold. Thus any improvement in circuit techniques or error recovery schemes to make the circuits more robust to these errors are likely to yield significant improvements in the circuit thresholds. This is precisely where topological subsystem codes come into picture. Because they can be designed to function with just two-body measurements, these codes can greatly mitigate the detrimental effects of measurement errors. A strong case in favor of the suitability of the subsystem codes with current quantum information technologies has already been made in [22].

For all these reasons topological subsystem codes are worth further investigation. This work is aimed at realizing the potential of topological subsystem codes. Our main contribution in this paper is to give large classes of topological subsystem codes, which were not previously known in literature. Our results put at our disposal a huge arsenal of topological subsystem codes, which aids in the evaluation of their promise for fault tolerant quantum computing. In addition to building upon the work of [22] it also sheds light on color codes, an area of independent interest.

The paper is structured as follows. After reviewing the necessary background on subsystem codes in Section II, we give our main results in Section III. Then in Section IV we show how to measure the stabilizer for the proposed codes in a consistent fashion. We conclude with a brief discussion on the significance of these results in Section V.

II. BACKGROUND AND PREVIOUS WORK

A. Subsystem codes

In the standard model of quantum error-correction, information is protected by encoding it into a subspace of the system Hilbert space. In the subsystem model of error correction [3, 11–14, 19], the subspace is further decomposed as $L \otimes G$. The subsystem L encodes the logical information, while the subsystem G provides additional degrees of freedom; it is also called the gauge subsystem and said to encode the gauge qubits. The notation $[[n, k, r, d]]$ is used to denote a subsystem code on n qubits, with $\dim L = 2^k$ and $\dim G = 2^r$ and able to detect errors of weight up to $d-1$ on the subsystem L. In this model an $[[n, k, d]]$ quantum code is the same as an $[[n, k, 0, d]]$ subsystem code.

The introduction of the gauge subsystem allows us to simplify the error recovery schemes [1, 3] since errors that act only on the gauge subsystem need not be corrected. Although sometimes this comes at the expense of a reduced encoding rate, nonetheless as in the case of the Bacon-Shor code, this can substantially improve the performance with respect to the corresponding stabilizer code associated with it without affecting the rate [3].

We assume that the reader is familiar with the stabilizer formalism for quantum codes [8, 9]. We briefly review it for the subsystem codes [11, 19]. A subsystem code is defined by a (nonableian) subgroup of the Pauli group; it is called the gauge group $\mathcal{G}$ of the subsystem code. We denote by $S' = Z(\mathcal{G})$, the centre of $\mathcal{G}$. Let $\langle i\mathbf{I}, S\rangle = S'$. The subsystem code is simply the space stabilized by S. (Henceforth, we shall ignore phase factors and let S be equivalent to S'.) Henceforth, we shall ignore the phase factors and let S. The bare logical operators of the code are given by the elements in $C(\mathcal{G})$, the centralizer of $\mathcal{G}$. (We view the identity also as a logical operator.) These logical operators do not act on the gauge subsystem but only on the information subsystem. The operators in $C(S)$ are called dressed logical operators and in general they also act on the gauge subsystem as well. For an $[[n, k, r, d]]$ subsystem code, with the stabilizer dimension $\dim S = s$, we have the following relations:

$$n = k + r + s, \tag{1}$$

$$\dim \mathcal{G} = 2r + s, \tag{2}$$

$$\dim C(\mathcal{G}) = 2k + s, \tag{3}$$

$$d = \min\{\mathrm{wt}(e) \mid e \in C(Z(\mathcal{G})) \setminus \mathcal{G}\}. \tag{4}$$

The notation $\mathrm{wt}(e)$ is used to denote the number of qubits on which the error e acts nontrivially.

B. Color codes

In the discussion on topological codes, it is tacitly assumed that the code is associated to a graph which is embedded on some suitable surface. Color codes [4] are a class of topological codes derived from 3-valent graphs with the additional property that they are 3-face-colorable. Such graphs are called 2-colexes. The stabilizer of the color code associated to such a 2-colex is generated by operators defined as follows:

$$B_f^\sigma = \prod_{i \in f} \sigma_i, \sigma \in \{X, Z\}, \tag{5}$$

where f is a face of Γ_2. A method to construct 2-colexes from standard graphs was proposed in [5]. Because of its relevance for us we briefly review it here.

Construction A Topological color code construction

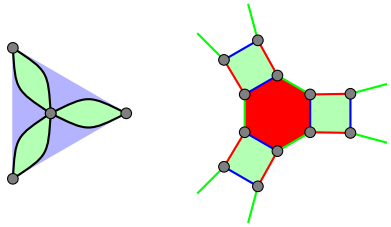
Input: An arbitrary graph Γ.

Output: A 2-colex Γ_2.

1: Color each face of the embedding by $x \in \{r, b, g\}$.

2: Split each edge into two edges and color the face by $y \in \{r, b, g\} \setminus x$ as shown below.

3: Transform each vertex of degree d into a face containing d edges and color it $z \in \{r, b, g\} \setminus \{x, y\}$. Denote this graph by Γ_2.



Notice that in the above construction, every vertex, face and edge in Γ lead to a face in Γ_2. Because of this correspondence,

we shall call a face in Γ_2 a v-face if its parent in Γ was a vertex, a f-face if its parent was a face and an e-face if its parent was an edge. Note that an e-face is always 4-sided.

C. Topological subsystem codes via color codes

At the outset it is fitting to distinguish topological subsystem codes from non-topological codes such as the Bacon-Shor codes that are nonetheless local. A more precise definition can be found in [6, 7], but for our purposes it suffices to state it in the following terms.

(i) The stabilizer S (and the gauge group) have local generators and $O(1)$ support.

(ii) Errors in $C(S)$ that have a trivial homology on the surface are in the stabilizer, while the undetectable errors have a nontrivial homology on the surface.

We denote the vertex set and edge set of a graph Γ by $V(\Gamma)$, $E(\Gamma)$ respectively. We denote the set of edges incident on a vertex v by $\delta(v)$ and the edges that constitute the boundary of a face by $\partial(f)$. We denote the Euler characteristic of a graph by χ, where $\chi = |V(\Gamma)| - |E(\Gamma)| + |F(\Gamma)|$. The dual of a graph is the graph obtained by replacing every face f with a vertex f^*, and for every edge in the boundary of two faces f_1 and f_2, creating a dual edge connecting f_1^* and f_2^*. The subsystem code construction due to [6] takes the dual of a 2-colex, and modifies it to obtain a subsystem code. The procedure is outlined below:

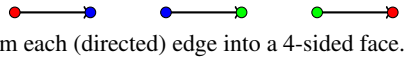
Construction B Topological subsystem code construction

Input: An arbitrary 2-colex Γ_2.

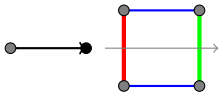
Output: Topological subsystem code.

1: Take the dual of Γ_2. It is a 3-vertex-colorable graph.

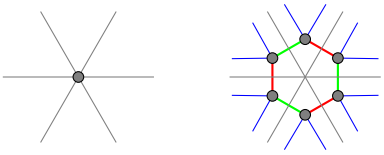
2: Orient each edge as a directed edge as per the following:



3: Transform each (directed) edge into a 4-sided face.



4: Transform each vertex into a face with as many sides as its degree. (The preceding splitting of edges implicitly accomplishes this. Each of these faces has a boundary of alternating blue and red edges.) Denote this expanded graph as $\overline{\Gamma}$.



5: With every edge $e = (u, v)$, associate a link operator $\overline{K}_e \in \{X_uX_v, Y_uY_v, Z_uZ_v\}$ depending on the color of the edge.

6: The gauge group is given by $\mathcal{G} = \langle \overline{K}_e \mid e \in E(\overline{\Gamma})\rangle$.

Our presentation slightly differs from that of [6] with respect to step 2.

Theorem A ([6]). *Let Γ_2 be a 2-colex embedded on a surface of genus g. The subsystem code derived from Γ_2 via Construction B has the following parameters:*

$$[[3|V(\Gamma_2)|, 2g, 2|V(\Gamma_2)| + 2g - 2, d \geq \ell^*]], \tag{6}$$

where ℓ^* is the length of smallest nontrivial cycle in Γ_2^*.

The cost of the two-body measurements is reflected to some extent in the increased overhead for the subsystem codes. Comparing with the parameters of the color codes, this construction uses three times as many qubits as the associated color code while at the same time encoding half the number of qubits. Our codes offer a different tradeoff between the overhead and distance.

D. Subsystem codes from 3-valent hypergraphs

In this section we review a general construction for (topological) subsystem codes based on hypergraphs proposed in [22]. A hypergraph Γ_h is an ordered pair (V, E), where $E \subseteq 2^V$ is a collection of subsets of V. The set V is called the vertex set while E is called the edge set. If all the elements of E are subsets of size 2, then Γ_h is a standard graph. Any element of E whose size is greater than 2 is called a hyperedge and its rank is its size. The rank of a hypergraph is the maximum rank of its edges. A hypergraph is said to be of degree k if at every site k edges are incident on it.

A hypercycle in a hypergraph is a set of edges such that on every vertex in the support of these edges an even number of edges are incident [16]. Note that this definition of hypercycle includes the standard cycles consisting of rank-2 edges. A hypercycle is said to have trivial homology if we can contract it to a point, by contracting its edges. Homological equivalence of cycles is somewhat more complicated than in standard graphs.

The following construction is due to [22]. Let Γ_h be a hypergraph satisfying the following conditions:

H1) Γ_h has only rank-2 and rank-3 edges.

H2) Every vertex is trivalent.

H3) Two edges intersect at most at one vertex[17].

H4) Two rank-3 edges are disjoint.

We assume that at every vertex there is a qubit. For each rank-2 edge $e = (u, v)$ define a link operator K_e where $K_e \in \{X_uX_v, Y_uY_v, Z_uZ_v\}$ and for each rank-3 edge (u, v, w) define

$$K_e = Z_uZ_vZ_w. \tag{7}$$

The assignment of these link operators is such that

$$K_eK_{e'} = (-1)^{|e\cap e'|}K_{e'}K_e. \tag{8}$$

We denote the cycles of Γ_h by Σ_{Γ_h}. Let σ be a hypercycle in Γ_h, then we associate a (cycle) operator $W(\sigma)$ to it as follows:

$$W(\sigma) = \prod_{e\in\sigma} K_e. \tag{9}$$

The group of these cycle operators is denoted $\mathcal{L}_{\Gamma_h}$ and defined as

$$\mathcal{L}_{\Gamma_h} = \langle W(\sigma) \mid \sigma \text{ is a hypercycle in } \Gamma_h\rangle \tag{10}$$

It is immediate that $\dim \mathcal{L}_{\Gamma_h} = \dim \Sigma_{\Gamma_h}$.

Construction C Topological subsystem code via hypergraphs

Input: A hypergraph Γ_h satisfying assumptions H1–4
Output: A subsystem code specified by its gauge group $\mathcal{G}$.

1: Color all the rank-3 edges, say with r. Then assign a 3-edge-coloring of Γ_h using $\{r, g, b\}$.
2: Define a graph $\overline{\Gamma}$ whose vertex set is same as Γ_h.
3: For each rank-2 edge (u, v) in Γ_h assign an edge (u, v) in $\overline{\Gamma}$ and a link operator $\overline{K}_{u,v} = K_{u,v}$ as

$$\overline{K}_{u,v} = \begin{cases} X_u X_v & (u,v) \text{ is } r \\ Y_u Y_v & (u,v) \text{ is } g \\ Z_u Z_v & (u,v) \text{ is } b \end{cases}$$

4: For each rank-3 edge (u, v, w) assign three edges in $\overline{\Gamma}$, namely, $(u, v), (v, w), (w, u)$ and three link operators $\overline{K}_{u,v} = Z_u Z_v$, $\overline{K}_{v,w} = Z_v Z_w$, and $\overline{K}_{w,u} = Z_w Z_u$.
5: Define the gauge group $\mathcal{G} = \langle \overline{K}_e \mid e \in \overline{\Gamma} \rangle$.

Theorem B ([22]). *A hypergraph Γ satisfying the conditions H1-4, leads to a subsystem code whose gauge group is the centralizer of Σ_{Γ_h}, i.e., $\mathcal{G} = C(\mathcal{L}_{\Gamma_h})$.*

Since $S = \mathcal{G} \cap C(\mathcal{G})$, a subgroup of cycles corresponds to the stabilizer. Let us denote this subgroup of cycles by Δ_{Γ_h}. Note that we have slightly simplified the construction proposed in [22], in that we let our our link operators to be only $\{X \otimes X, Y \otimes Y, Z \otimes Z\}$. But we expect that this results in no loss in performance, because the number of encoded qubits and the distance are topological invariants and are not affected by these choices.

Our notation is slightly different from that of [22]. We distinguish between the link operators associated with the hypergraph Γ_h and the derived graph $\overline{\Gamma}_h$; they coincide for the rank-2 edges. Because the hypergraph is 3-edge-colorable, we can partition the edge set of the hypergraph as $E(\Gamma_h) = E_r \cup E_g \cup E_b$ depending on the color. The derived graph $\overline{\Gamma}_h$ is not 3-edge-colorable, but we group its edges by the edges of the parent edges in Γ_h. Thus we can partition the edges of $\overline{\Gamma}_h$ also in terms of color as $E(\overline{\Gamma}_h) = \overline{E}_r \cup \overline{E}_g \cup \overline{E}_b$.

This following result is a consequence of the definitions of $\mathcal{G}$, Σ_{Γ_h} and Theorem B.

Corollary C. *If σ is a cycle in Γ_h and consists of only rank-2 edges, then $W(\sigma) \in S$.*

An obvious question posed by Theorem B is how does one construct hypergraphs that satisfy these constraints. This question will occupy us in the next section. A related question is the syndrome measurement schedule for the associated subsystem code. This will be addressed in Section IV.

III. PROPOSED TOPOLOGICAL CODES

A. Color codes

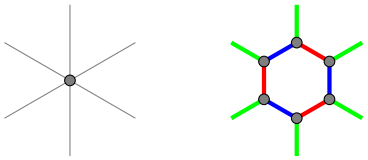
While our main goal is to construct subsystem codes, our techniques use color codes as intermediate objects. The previously known methods [5] for color codes do not exhaust all possible color codes. Therefore we make a brief digression to propose a new method to construct color codes. Then we will return to the question of building subsystem codes.

The constructions presented in this paper assume that the associated graphs and hypergraphs are connected, have no loops and all embeddings are such that the faces are homeomorphic to unit discs, in other words, all our embeddings are 2-cell embeddings.

Construction 1 Topological color code construction

Input: An arbitrary bipartite graph Γ.
Output: A 2-colex Γ_2.

1: Consider the embedding of the bipartite graph Γ on some surface. Take the dual of Γ, denote it Γ^*.
2: Since Γ is bicolorable, Γ^* is a 2-face-colorable graph.
3: Replace every vertex of Γ^* by a face with as many sides as its degree such that every new vertex has degree 3.



4: The resulting graph is a 2-colex.

Theorem 1 (Color codes from bipartite graphs). *Any 2-colex must be generated from Construction 1 via some bipartite graph.*

Proof. Assume that there is a 2-colex that cannot be generated by Construction 1. Assuming that the faces and the edges are 3-colored using $\{r, g, b\}$, pick any color $c \in \{r, g, b\}$. Then contract all the edges of the remaining colors, namely $\{r, g, b\} \setminus c$. This process shrinks the faces that are coloured c. The c-colored faces become the vertices of the resultant 2-face-colorable graph. The dual of this graph is bipartite as only bipartite graphs are 2-colorable. But this is precisely the reverse of the process described above. Therefore, the 2-colex must have risen from a bipartite graph. □

Note that there need not be a unique bipartite graph that generates a color code. In fact, three distinct bipartite graphs may generate the same color code, using the above construction.

We also note that the 2-colexes obtained via construction A have the property that for one of the colours, all the faces are of size 4. The following result shows the relation between our result and Construction A. The proof is straightforward and omitted.

Corollary 2. *The color codes arising from Construction A can be obtained from Construction 1 using bipartite graphs which have the property that one bipartition of vertices contains only vertices of degree two.*

B. Subsystem codes via color codes

Here we outline a procedure to obtain a subsystem code from a color code. This uses the construction of [22]. We

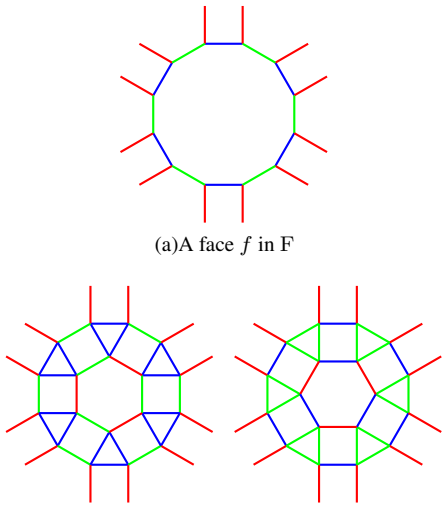
first construct a hypergraph that satisfies H1–4. We start with a 2-colex that has an additional restriction, namely it has a nonempty set of faces each of which has a doubly even number of vertices.

Construction 2 Topological subsystem code construction

Input: A 2-colex Γ_2, assumed to have a 2-cell embedding.

Output: A topological subsystem code specified by the hypergraph Γ_h.

1: We assume that the faces of Γ_2 are colored r, b, and g. Let $\mathrm{F_r}$ be the collection of r-colored faces of Γ_2, and $\mathrm{F} \subseteq \mathrm{F_r}$ such that $|f| \equiv 0 \bmod 4$ and $|f| > 4$ for all $f \in \mathrm{F}$.
2: **for** $f \in \mathrm{F}$ **do**
3: Add a face f' inside f such that $|f| = 2|f'|$.
4: Take a collection of alternating edges in the boundary of f. These are $|f|/2$ in number and are all colored either b or g.
5: Promote them to rank-3 edges by adding a vertex from f' so that the resulting hyperedges do not "cross" each other. In other words, the rank-3 edge is a triangle and the triangles are disjoint. Two possible methods of inserting the rank-3 edges are illustrated in Fig. 1. In the first method, the hyperedges can be inserted so that they are in the boundary of the g colored faces, see Fig. 1(b). Alternatively, the hyperedges can be inserted so that they are in the boundary of the b colored faces, see Fig. 1(c).
6: Color the rank-3 edge with the same color as the parent rank-2 edge.
7: Color the edges of f' using colors distinct from the color of the rank-3 edges incident on f'.
8: **end for**
9: Denote the resulting hypergraph Γ_h and use it to construct the subsystem code as in Construction C.



(a)A face f in F

(b)Inserting rank-3 edges in f by promoting the b-edges to rank-3 edges.

(c)Inserting rank-3 edges in f by promoting the g-edges to rank-3 edges.

FIG. 1. (Color online) Inserting rank-3 edges in the faces of Γ_2 to obtain the hypergraph Γ_h. The rank-3 edges correspond to triangles.

Theorem 3 (Subsystem codes from color codes)**.** *Construction 2 gives hypergraphs which satisfy the constraints H1-4 and therefore give rise to 2-local subsystem codes whose cycle group Σ_{Γ_h} is defined as in Eq.* (10) *and gauge group is $\mathcal{G} = C(\mathcal{L}_{\Gamma_h})$.*

Proof. Requirement H1 is satisfied because by construction, only rank-3 hyper edges are added to Γ_2, which only contains rank-2 edges. The hypergraph has two types of vertices those that come from Γ_2 and those that are added due to introduction of the hyperedges. Since all hyperedges come by promoting an edge to a hyperedge, it follows that the hypergraph is trivalent on the original vertices inherited from Γ_2. By construction, the vertices in $V(\Gamma_h) \setminus V(\Gamma_2)$ are trivalent and thus Γ_h satisfies H2. Note that $|f| \equiv 0 \bmod 4$ and $|f| > 4$, therefore f' can be assigned an edge coloring that ensures that Γ_h is 3-edge colorable. Since $|f| > 4$ we also ensure that no two edges intersect in more than one site, and H3 holds. By construction, all rank-3 edges are disjoint. This satisfies requirement H4. □

Let us illustrate this construction using a small example. It is based on the 2-colex shown in Fig. 2. The hypergraph derived from this 2-colex is shown in Fig. 3. Its rate is nonzero.

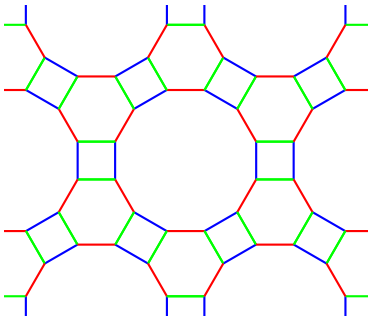


FIG. 2. (Color online) Color code on a torus from a 4-6-12 lattice. Opposite sides are identified.

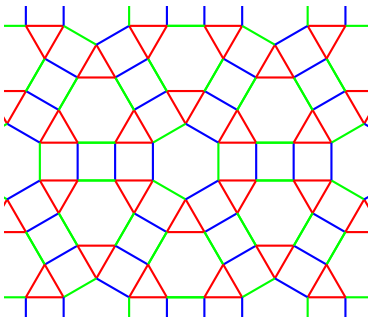


FIG. 3. (Color online) Illustrating Construction 2.

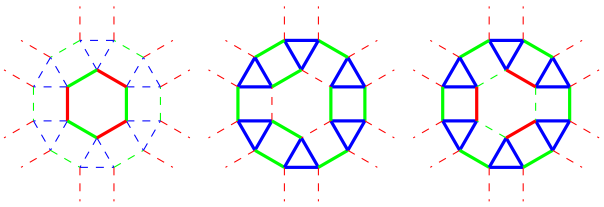
At this point, Theorem 3 is still quite general and we do not have expressions for the code parameters in closed form. Neither is the structure of the stabilizer and the logical operators very apparent. We impose some constraints on the set F so

that we can remedy this situation. These restrictions still lead to a large class of subsystem codes.

(i) $\mathrm{F} = \mathrm{F_c}$ is the set of all the faces of a given color; see Theorem 4.

(ii) F is an alternating set and $\mathrm{F_c}$ and $\mathrm{F} \setminus \mathrm{F_c}$ form a bipartite graph (in a sense which will be made precise shortly); see Theorem 5.

Before, we can evaluate the parameters of these codes, we need some additional results with respect to the structure of the stabilizer and the centralizer of the gauge group. The stabilizers vary depending on the set F, nevertheless we can make some general statements about a subset of these stabilizers.



(a)A hypercycle σ_1 in f (shown in bold edges) consisting of only rank-2 edges.

(b)A hypercycle σ_2 in f (shown in bold edges) with both rank-2 and rank-3 edges.

(c)A dependent hypercycle σ_3 which is a combination of σ_1 and σ_2 over $\mathbb{F}_2$.

FIG. 4. (Color online) Stabilizer generators from a face in F for the subsystem codes of Construction 1; one of them is dependent. We shall view σ_1 and σ_2 as the two independent hypercycles associated with f.

Lemma 1. *Suppose that f is a face in* F *in Construction 1. Then there are two independent hypercycles that we can associate with this face and consequently two independent stabilizer generators as shown in Fig. 4*

Proof. We use the same notation as in Construction 1. Then Construction 1 adds a new face f' to Γ_2 in the interior of f. Let σ_1 be the cycle formed by the rank-2 edges in the boundary of f', see Fig. 4(a). By Corollary C, $W(\sigma_1) \in S$.

Now let σ_2, see Fig. 4(b), be the hypercycle consisting of all the edges in the boundary of f and an alternating set of rank-2 edges in the boundary of f'. In other words, σ_2 consists of all the rank-3 edges inserted in f as well as the rank-2 edges in its boundary and an alternating pair of rank-2 edges in f'. Because $|f| \equiv 0 \bmod 4$, the boundary of f' is 2-edge colorable. To prove that $W(\sigma_2)$ can be generated by the elements of $\mathcal{G}$, observe that $W(\sigma_2)$ can be split as

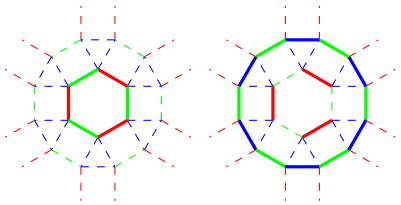
$$W(\sigma_2) = \prod_{e \in \partial(f)} K_e \prod_{e \in \partial(f') \cap E_g} K_e,$$

where E_g refers to the r-colored edges in Γ_h and the boundary is with respect to Γ_h. We can also rewrite this in terms of the link operators in $\overline{\Gamma}_h$.

$$W(\sigma_2) = \prod_{e \in \partial(f)} \overline{K}_e \prod_{e \in \partial(f') \cap \overline{E}_r} \overline{K}_e$$

where the boundary is with respect to $\overline{\Gamma}_h$ and $\overline{E}_r$ now refers to the r-colored edges in $\overline{\Gamma}_h$.

This is illustrated in Fig. 5(b). The third cycle σ_3, see Fig. 4(c), can be easily seen to be a combination of the cycles σ_1 and σ_2 over $\mathbb{F}_2$. □



(a)Decomposing the hypercycle σ_1.

(b)Decomposing the hypercycle σ_2.

FIG. 5. (Color online) Decomposing σ_i in Fig. 4 so that $W(\sigma_i)$ can be generated using the elements of $\mathcal{G}$. In each of the above $W(\sigma_i)$ can be generated as the product of link operators corresponding to the bold edges. Note that these decompositions are with respect to the link operators of the derived graph $\overline{\Gamma}_h$ while the cycles are defined with respect to the hypergraph Γ_h.

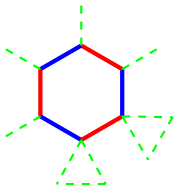
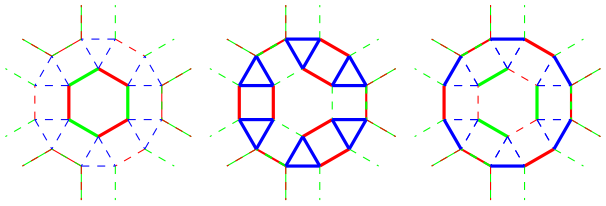


FIG. 6. (Color online) A cycle σ_1 of rank-2 edges in the boundary of f, shown in bold, when f has no rank-3 edges in its boundary. Some of the edges incident on f maybe rank-3 but none in the boundary are.



(a)A cycle σ_1 of rank-2 edges in the boundary of f, shown in bold, note that a rank-3 edge is incident on every vertex of f unlike Fig. 6.

(b)A cycle σ_2 of rank-2 and rank-3, shown in bold; σ_2 differs from the cycle in Fig. 4(b), in that the "outer" rank-2 edges maybe either r or g.

(c)Decomposing the hypercycle σ_2 so that $W(\sigma_2)$ can be generated using the elements of $\mathcal{G}$. Note the decomposition refers to $\overline{\Gamma}_h$.

FIG. 7. (Color online) Stabilizer generators for a face which has no rank-3 edges in its boundary when $\mathrm{F} = \mathrm{F_c}$ and $f \notin \mathrm{F}$.

Lemma 2. *Suppose that f has no rank-3 edges in its boundary $\partial(f)$ as in Fig. 6. Then $W(\partial(f))$ is in S. Further, if* $\mathrm{F} = \mathrm{F_r}$ *and $f \notin$* F*, then we can associate another hypercycle σ_2 to f, as in Fig. 7, such that $W(\sigma_2)$ is in S.*

Proof. If f has no rank-3 edges in its boundary, then $W(\partial(f))$ is in S by Corollary C. It is possible that some rank-3 edges are incident on f even though they are not in its boundary. This is illustrated in Fig. 6.

If $\mathrm{F} = \mathrm{F}_\mathrm{r}$, and $f \notin \mathrm{F}$, then a rank-3 edge is incident on every vertex of f and we can form another cycle by considering all the rank-3 edges, and rank-2 edges connecting all pairs of rank-3 edges, see Fig. 7(b). This includes an alternating set of edges in the boundary of f. This is different from the hypercycle in Fig. 4(b) in that the "outer" rank-2 edges connecting the rank-3 edges maybe of different color. Nonetheless by an augment similar to that in the proof of Lemma 1, and using the decomposition shown in Fig. 7(c) we can show that $W(\sigma_2)$ is in S. □

Remark 1. *(Canonical cycles.) For the faces in which have two stabilizer generators associated with them we make the following canonical choice for the stabilizer generators. The first basis cycle σ_1 always refers to the cycle consisting of the rank-2 edges forming the boundary of a face. The second basis cycle for f is chosen to be the cycle in which the rank-3 edges are paired with an adjacent rank-3 edge such that both the rank-2 edges pairing them are of the same color.*

The decomposition as illustrated in Fig. 7(c) works even when the stabilizer is for a face which is adjacent to itself.

Next we prove a bound on the distance of the codes obtained via Construction 2. This is defined by the cycles in space $\Sigma_{\Gamma_h} \setminus \Delta_{\Gamma_h}$. Recall that $W(\sigma) \in S$, if $\sigma \in \Delta_{\Gamma_h}$.

Lemma 3. *(Bound on distance) The distance of the subsystem code obtained from Construction 2 is upper bounded by the number of rank-3 edges in the hypercycle with minimum number of rank-3 edges in $\Sigma_{\Gamma_h} \setminus \Delta_{\Gamma_h}$.*

Proof. Every undetectable error of the subsystem code can be written as $gW(\sigma)$ for some $g \in \mathcal{G}$ and $\sigma \in \Sigma_{\Gamma_h} \setminus \Delta_{\Gamma_h}$. It suffices therefore, to check by how much the weight of $W(\sigma)$ can be reduced by acting with elements of $\mathcal{G}$. In particular, we can reduce $W(\sigma)$ such that only the rank-3 edges remain, and obtain an equivalent operator of lower weight. We can further act on this so that corresponding to every rank-3 edge in σ the modified error has support only on one of its vertices. This reduced error operator has weight equal to the number of rank-3 edges in σ. Thus the distance of the code is upper bounded by the number of rank-3 edges in the hypercycle with minimum number of hyperedges in $\Sigma_{\Gamma_h} \setminus \Delta_{\Gamma_h}$. □

It appears that this bound is tight, in that the distance is actually no less than the one specified above.

Theorem 4. *Suppose that Γ is a graph such that every vertex has even degree greater than 2. Then construct the 2-colex Γ_2 from Γ using Construction A. Then apply Consruction 2 with F being the set of v-faces of Γ_2 and with the rank-3 edges being in the boundaries of the e-faces of Γ_2. Let ℓ be the number of rank-3 edges in a cycle in $\Sigma_{\Gamma_h} \setminus \Delta_{\Gamma_h}$. Then we obtain a*

$$[[6e, 1 + \delta_{\Gamma^*,\textit{bipartite}} - \chi, 4e - \chi, d \leq \ell]] \tag{11}$$

subsystem code where $e = |E(\Gamma)|$ and $\delta_{\Gamma^*,\textit{bipartite}} = 1$ if Γ^* is bipartite and zero otherwise.

Proof. Assume that Γ has v vertices, f faces and e edges. Let us denote this by the tuple (v, f, e), then $\chi = v + f - e$. On applying Construction A, we obtain a 2-colex, Γ_2 with the parameters $(4e, v + f + e, 6e)$. When we apply Construction 1 to Γ_2, the resulting hypergraph Γ_h has $2e$ new vertices added to it. Further $2e$ edges are promoted to hyper edges, and as many new rank-2 edges are created. Thus we have a hypergraph with $6e$ vertices, $2e$ hyperedges, $6e$ rank-2 edges.

The important thing to note is that the dimension of the hypercycle space of Γ_h is related to I_{Γ_h}, the vertex-edge incidence matrix of Γ_h. Let $E(\Gamma_h)$ denote the edges of Γ_h including the hyperedges. Then

$$\dim \mathcal{L}_{\Gamma_h} = |E(\Gamma_h)| - \mathrm{rank}_2(I_{\Gamma_h}), \tag{12}$$

where rank_2 denotes the binary rank, [20].

By Lemma 5, $\mathrm{rank}_2(I_{\Gamma_h}) = |V(\Gamma_h)| - 1 - \delta_{\Gamma^*,\text{bipartite}}$. It now follows that

$$\begin{aligned}\dim \mathcal{L}_{\Gamma_h} &= |E(\Gamma_h)| - |V(\Gamma_h)| + 1 + \delta_{\Gamma^*,\text{bipartite}}. \\ &= 8e - 6e + 1 + \delta_{\Gamma^*,\text{bipartite}} \\ &= 2e + 1 + \delta_{\Gamma^*,\text{bipartite}}\end{aligned}$$

By Lemma 1 and 2 every v-face and f-face of Γ_2 lead to two hypercycles in Γ_h. These are $2v + 2f$ in number. But depending on whether Γ^* is bipartite of these only $s = 2v + 2f - 1 - \delta_{\Gamma^*,\text{bipartite}}$ are independent hypercycles. The dependencies are as given below:

$$\prod_{f \in v\text{-faces}} W(\sigma_1^f) = \prod_{f \in f\text{-faces}} W(\sigma_2^f). \tag{13}$$

If Γ^* is bipartite then we have the following additional dependency. Let Γ be face-colored black and white so that $F(\Gamma) = F_1 \cup F_2$, where F_1 and F_2 are the collection of black and white faces. Then

$$\prod_{f \in f\text{-faces}} W(\sigma_1^f) \prod_{f \in F_1} W(\sigma_2^f) = \prod_{f \in v\text{-faces}} W(\sigma_2^f) \tag{14}$$

$$\prod_{f \in f\text{-faces}} W(\sigma_1^f) \prod_{f \in F_2} W(\sigma_2^f) = \prod_{f \in v\text{-faces}} W(\sigma_1^f) W(\sigma_2^f) \tag{15}$$

(Note that among equations (13)–(15) only two are independent.) All these are of trivial homology. There are no other independent cycles of trivial homology. Furthermore, Lemma 6 and 7 show that hypercycles of nontrivial homology are not in the gauge group. Thus all the remaining (nontrivial) hypercycles are not in the stabilizer. We can now compute the number of encoded qubits as follows.

$$\begin{aligned}2k &= \dim C(\mathcal{G}) - s \\ &= 2e + 1 + \delta_{\Gamma^*,\text{bipartite}} - (2v + 2f - 1 - \delta_{\Gamma^*,\text{bipartite}}) \\ &= 2 + 2\delta_{\Gamma^*,\text{bipartite}} + 2(e - v - f),\end{aligned}$$

which gives $k = 1 + \delta_{\Gamma^*,\text{bipartite}} - \chi$ encoded qubits. The number of gauge qubits r can now be computed as follows:

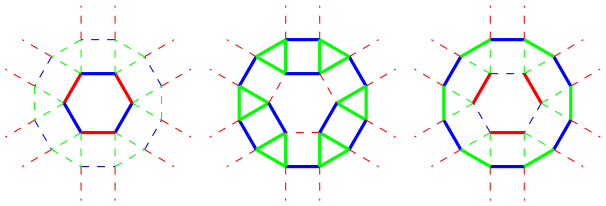
$$\begin{aligned} r &= n - k - s \\ &= 6e - (1 + \delta_{\Gamma^*,\text{bipartite}} - \chi) - (2v + 2f - 1 - \delta_{\Gamma^*,\text{bipartite}}) \\ &= 6e - 2v - 2f = 4e - \chi. \end{aligned}$$

The bound on distance follows from Lemma 3. □

Remark 2. *Note that there are no planar non-bipartite graphs Γ^* which satisfy the constraint in Theorem 4.*

Remark 3. *We might consider a variation is possible on the above, namely, adding the hyper edges in the f-faces as opposed to the v-faces. This however does not lead to any new codes that are not constructible using Theorem 4. Adding them in the f-faces is equivalent to applying Theorem 4 to the dual of Γ.*

In Theorem 4, when Γ^* is bipartite, the subsystem codes coincide with those obtained from [6]. However in this situation, a different choice of F in Construction 2 gives another family of codes that differ from [6] and Theorem 4. These codes are considered next. But first we need an intermediate result about the hypercycles in Δ_{Γ_h} those that define the stabilizer. Some of such as those in Fig. 8 are similar to those in Fig. 4 but some such as those in Fig. 9 are not.



(a)A hypercycle σ_1 for a v-face in F.

(b)A cycle σ_2 of rank-2 and rank-3 edges, shown in bold.

(c)Decomposing σ_2 so that $W(\sigma_2)$ can be generated using the elements of $\mathcal{G}$.

FIG. 8. (Color online) Stabilizer generators for a v-face in F, for the subsystem codes in Theorem 5. Also shown is the decomposition for $W(\sigma_2)$. The decomposition for $W(\sigma_1)$ is same as in Fig. 8(a).

Before, we give the next construction, we briefly recall the definition of a medial graph. The medial graph of a graph Γ is obtained by placing a vertex on every edge of Γ and an edge between two vertices if and only if they these associated edges in Γ are incident on the same vertex. We denote the medial graph of Γ by Γ_m.

Theorem 5. *Let Γ be a graph whose vertices have even degrees greater than 2 and Γ_m its medial graph. Construct the 2-colex Γ_2 from Γ_m^* using Construction A. Since Γ_m^* is bipartite, the set of v-faces of Γ_2, denoted $\mathrm{F_r}$, form a bipartition $\mathrm{F_v} \cup \mathrm{F_f}$, where $|\mathrm{F_v}| = |\mathrm{V}(\Gamma)|$. Apply Consruction 2 with the set $\mathrm{F_v} \subsetneq \mathrm{F}$ such that the rank-3 edges are not in the boundaries of the e-faces of Γ_2. Let ℓ be the number of rank-3 edges in a cycle in $\Sigma_{\Gamma_h} \setminus \Delta_{\Gamma_h}$. Then we obtain a*

$$[[10e, 1 - \chi + \delta_{\Gamma^*,\textit{bipartite}}, 6e - \chi, d \leq \ell]] \tag{16}$$

subsystem code, where $e = |E(\Gamma)|$.

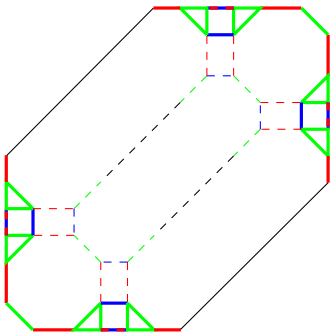


FIG. 9. (Color online) Stabilizer generators for a v-face in $\mathrm{F_r} \setminus \mathrm{F}$, for the subsystem codes in Theorem 5. i) $\sigma_1 = \partial(f)$ (not shown) and ii) σ_2 (in bold) consists of the rank-3 edges of all the adjacent f-faces in F adjacent through an e-face and the rank-2 edges connecting them. The decomposition for $W(\sigma_2)$ is shown in Fig. 10.

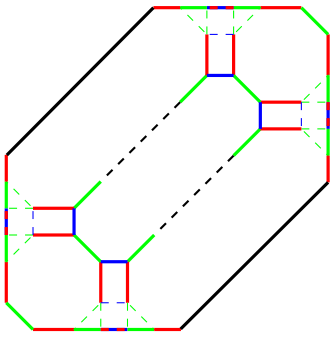


FIG. 10. (Color online) Decomposition for $W(\sigma_2)$. The product of the link operators shown in bold edges gives $W(\sigma_2)$.

Proof. The proof is somewhat similar to that of Theorem 4, but there are important differences. Suppose that Γ has v vertices, f faces and e edges. Let us denote this as the tuple (v, f, e). The medial graph Γ_m is 4-valent and has e vertices, $v + f$ faces and $2e$ edges. The dual graph Γ_m^* has the parameters $(v + f, e, 2e)$. Furthermore, Γ_m^* is bipartite. The 2-colex Γ_2 has the parameters, $(8e, v+f+3e, 12e)$. Of the $v+f+3e$ faces $v + f$ are v-type, e are f-type and $2e$ are e-type. The hypergraph has $10e$ vertices because a new vertex is added for every pair of rank-2 edge incident on the v-faces in $\mathrm{F_v}$. These incident edges are all of one color, which are a third of the total edges of Γ_m^* i.e., $(12e/3)$. Since a rank-3 edge is added only on one end of these edges for every pair, this implies that $2e$ edges are promoted to rank-3 edges, as many new vertices and new rank-2 edges are added to form the hypergraph Γ_h.

By Lemma 5, the rank of the vertex-edge incidence matrix of Γ_h is $|V(\Gamma_h)| - 1 - \delta_{\Gamma^*,\text{bipartite}} = 10e - 1 - \delta_{\Gamma^*,\text{bipartite}}$. The total number of edges of Γ_h is $14e$ including the rank-3 edges. Thus the rank of the cycle space of Γ_h is

$$\begin{aligned} \dim \mathcal{L}_{\Gamma_h} &= 14e - 10e + 1 + \delta_{\Gamma^*,\text{bipartite}} && (17) \\ &= 4e + 1 + \delta_{\Gamma^*,\text{bipartite}}. && (18) \end{aligned}$$

The stabilizer generators of this code are somewhat different than those in Theorem 4. Recall that the v-faces form a bipartition, $\mathrm{F_v} \cup \mathrm{F_f} = \mathrm{F} \cup (\mathrm{F_r} \setminus \mathrm{F})$, where $|\mathrm{F_v}| = \mathrm{v}$ and

$|\mathrm{F_f}| = \mathrm{f}$. We insert the rank-3 edges only in the faces in F, and by Lemma 1 each of these faces leads to two stabilizer generators. These are illustrated in Fig. 8. The remaining v-faces namely those in $\mathrm{F_r} \setminus \mathrm{F}$, have no rank-3 edges in their boundary. Therefore, by Lemma 2 there is a stabilizer generator associated with the boundary of the face. The other generator associated to a face in $\mathrm{F_r} \setminus \mathrm{F}$ is slightly more complicated. It is illustrated in Fig. 9. The idea behind the decomposition so that it is an element of the gauge group is illustrated Fig. 10.

Thus both the v-faces of Γ_2 give rise to two types of stabilizer generators. Since these are $v + f$ in number, we have $2(v + f)$ due to them. Each of the e-faces gives rise to one stabilizer generator giving $2e$ more generators. Thus there are totally $2(v + f) + 2e$. However there are some dependencies.

$$\prod_{f\in \mathrm{F_v}} W(\sigma_1^f) = \prod_{f\in e\text{-faces}} W(\sigma_1^f) \prod_{f\in \mathrm{F_f}} W(\sigma_1^f)W(\sigma_2^f) \quad (19)$$

When Γ^* is bipartite, then it induces a bipartition on the v-faces in $\mathrm{F_v} = \mathrm{F}_1 \cup \mathrm{F}_2$. as well as the e-faces, depending on whether the e-face is adjacent to a v-face in F_1 or F_2. Denote this bipartition of e-faces as $E_1 \cup E_2$. Then the following hold:

$$\prod_{f\in \mathrm{F_v}} W(\sigma_2^f) = \prod_{f\in E_1} W(\sigma_1^f) \prod_{f\in F_1} W(\sigma_2^f) \prod_{f\in F_2} W(\sigma_1^f)$$
$$\prod_{f\in \mathrm{F_v}} W(\sigma_1^f)W(\sigma_2^f) = \prod_{f\in E_2} W(\sigma_1^f) \prod_{f\in F_1} W(\sigma_1^f) \prod_{f\in F_2} W(\sigma_2^f)$$

Observe though there is only one new dependency when Γ^* is bipartite. The f-faces do not give rise to anymore independent generators. Thus there are $s = 2(v + f + e) - 1 - \delta_{\Gamma^*,\text{bipartite}}$ independent cycles of trivial homology. The remaining cycles are of nontrivial homology. By Lemma 6 and 7, these cycles are not in the gauge group. Therefore the number of encoded qubits is given by

$$\begin{aligned} 2k &= \dim \mathcal{L}_{\Gamma_h} - s \\ &= 4e + 1 + \delta_{\Gamma^*,\text{bipartite}} - 2(v + f + e) + 1 + \delta_{\Gamma^*,\text{bipartite}} \\ &= 2 + 2\delta_{\Gamma^*,\text{bipartite}} + 2(e - v - f) \\ &= 2 + 2\delta_{\Gamma^*,\text{bipartite}} - 2\chi \end{aligned}$$

Thus $k = 1 - \chi + \delta_{\Gamma^*,\text{bipartite}}$. It is now straightforward to compute the number of gauge qubits as $r = n-k-s = 10e-(1+\delta_{\Gamma^*,\text{bipartite}}-\chi)-2(v+f+e)+1+\delta_{\Gamma^*,\text{bipartite}} = 6e-\chi$. The bound on distance follows from Lemma 3. □

Theorem 5 can be strengthened without having to go through a medial graph but rather starting with an arbitrary graph Γ and then constructing a 2-colex via Construction A. We now demonstrate that Construction 1 gives rise to subsystem codes that are different from those obtained in [6].

Lemma 4. *Suppose that we have a topological subsystem code obtained by Construction B from a 2-colex* Γ. *Then in the associated hypergraph shrinking the hyperedges to a vertex gives a 6-valent graph and further replacing any multiple edges by a single edge gives us a 2-colex.*

Proof. Construction B adds a rank-3 edge in every face of Γ^*. On contracting these rank-3 edges we end up with a graph whose vertices coincide with the faces of Γ^*. Each of these vertices is now 6-valent and between any two adjacent vertices there are two edges. On replacing these multiple edges by a single edge, we end up with a cubic graph. Observe that the vertices of this graph are in one to one correspondence with the faces of Γ^* while the edges are also in one to one correspondence with the edges of Γ^*. Further an edge is present only if two faces are adjacent. This is precisely the definition of the dual graph. Therefore, the resulting graph is the same as Γ and a 2-colex. □

Theorem 6. *Construction 2 results in codes which cannot be constructed using Construction B. In particular, all the codes of Theorem 5 are distinct from Construction B and the codes of Theorem 4 are distinct when* Γ *therein is non-bipartite.*

Proof. Let us assume that the Construction 2 does not give us *any* new codes. Then every code constructed using this method gives a code that is already constructed using Construction B. Lemma 4 informs us that contracting the rank-3 edges results in a 6-valent graph, which on replacing the multiple edges by single edge gives us a 2-colex.

But note that if we applied the same procedure to a graph that is obtained from the proposed construction, then we do not always satisfy this criterion. In particular, this is the case for the subsystem codes of Theorem 5. These codes do not give rise to a 6-valent lattice on shrinking the rank-3 edges to a single vertex.

When we consider the codes of Theorem 4, on contracting that rank-3 edges, we end with up a 6-valent graph with double edges and replacing them leads to a cubic graph. In order that these codes do not arise from Construction B, it is necessary that this cubic graph is not a 2-colex. And if it were a 2-colex then further reducing the v-faces of this graph should give us a a 2-face-colorable graph. But this reduction results in the graph we started out with namely, Γ^*. Thus when Γ^* in non-bipartite, our codes are distinct from those in [6]. □

Lemma 5. *The vertex-edge incidence matrices of the hypergraphs in Theorems 4 and 5 have rank* $|V(\Gamma_h)| - 1 - \delta_{\Gamma^*,\textit{bipartite}}$.

Proof. We use the same notation as that of Theorems 4 and 5. Denote the vertex edge incidence matrix of Γ_2 as I_{Γ_2}. Depending on whether an edge in Γ_2 is promoted to a hyperedge in Γ_h we can distinguish two types of edges in Γ_2. Suppose that the edges in $\{e_1, \ldots, e_l\}$ are not promoted while the edges in $\{e_{l+1}, \ldots, e_m\}$ are promoted.

$$I_{\Gamma_2} = \left[\begin{array}{ccc|ccc} \overset{e_1}{i_{11}} & \overset{\cdots}{\cdots} & \overset{e_l}{i_{1l}} & \overset{e_{l+1}}{\cdot} & \overset{\cdots}{\cdots} & \overset{e_m}{i_{1m}} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ i_{n1} & \cdots & i_{nl} & \cdot & \cdots & i_{nm} \end{array}\right] \quad (20)$$

The vertex-edge incidence matrix of Γ_h is related to that of

I_{Γ_2} as follows:

$$I_{\Gamma_h} = \left[\begin{array}{ccc|ccc|ccc} & e_1 & \cdots & e_l & e_{l+1} & \cdots & e_m & e_{m+1} & \cdots & e_q \\ & i_{11} & \cdots & i_{1l} & \cdot & \cdots & i_{1m} & & & \\ & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & & \mathbf{0} & \\ & i_{n1} & \cdots & i_{nl} & \cdot & \cdots & i_{nm} & & & \\ & \mathbf{0} & & & \mathbf{I} & & & & I_{\Gamma_h\setminus\Gamma_2} & \end{array}\right]$$
$$= \left[\begin{array}{ccc} I_{\Gamma_2} & & 0 \\ 0 & I & I_{\Gamma_h\setminus\Gamma_2} \end{array}\right], \tag{21}$$

where $I_{\Gamma_h\setminus\Gamma_2}$ is the incidence matrix of the subgraph obtained by restricting to the vertices $V(\Gamma_h)\setminus V(\Gamma_2)$. We already know that $\mathrm{rank}_2(I_{\Gamma_2})$ is $|V(\Gamma_2)|-1$. Suppose there is an additional linear dependence among the rows of I_{Γ_h}. More precisely, let

$$b = \sum_{v\in V(\Gamma_2)} a_v\delta(v) = \sum_{v\in V(\Gamma_h)\setminus V(\Gamma_2)} a_v\delta(v), \tag{22}$$

where δ_v is the vertex-edge incidence vector of v. Then b must have no support on the edges in $\{e_1,\ldots,e_l\}\cup\{e_{m+1},\ldots,e_q\}$. It must have support only on the rank-3 edges of Γ_h.

Every rank-3 edge has the property that it is incident on exactly one vertex $u\in V(\Gamma_h)\setminus(\Gamma_2)$ and exactly two vertices in $v,w\in V(\Gamma_2)$. Thus if a rank-3 edge has nonzero support in b, then $a_u\neq 0$ and either $a_v\neq 0$ or $a_w\neq 0$ but not both.

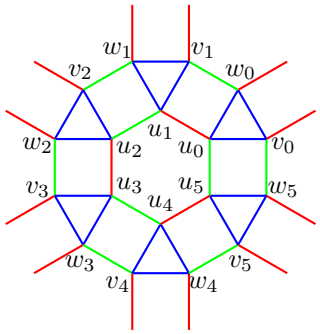


FIG. 11. (Color online) If b defined in Eq. (22) has support on one rank-3 edge of a v-face, then it has support on all the rank-3 edges of the v-face. Further, $\{a_{w_0},a_{w_2},a_{w_4},\ldots\}\cup\{a_{v_1},a_{v_3},\ldots\}$ are all nonzero or $\{a_{w_1},a_{w_3},\ldots\}\cup\{a_{v_0},a_{v_2},\ldots\}$ are all nonzero.

Suppose that a vertex $u_0\in V(\Gamma_h)\setminus V(\Gamma_2)$ is such that $a_{u_0}\neq 0$. Then because b has no support on the edges in $\{e_{m+1},\ldots,e_q\}$, all the rank-2 neighbors of u_0, that is those which are connected by rank-2 edges are also such that $a_{u_i}\neq 0$. This implies that in a given v-face, for all the vertices of $u_i\in(V(\Gamma_h)\setminus V(\Gamma_2))\cap f'$, we have $a_{u_i}\neq 0$. Further, only one of the rank-3 neighbors of u_i, namely v_i,w_i, can have $a_{v_i}\neq 0$ or $a_{w_i}\neq 0$, but not both. Additionally, pairs of these vertices must be adjacent as b has no support on the rank-2 edges. Thus either $\{a_{w_0},a_{w_2},a_{w_4},\ldots\}\cup\{a_{v_1},a_{v_3},\ldots\}$ are all nonzero or $\{a_{w_1},a_{w_3},\ldots\}\cup\{a_{v_0},a_{v_2},\ldots\}$ are all nonzero. Alternatively, we can say only the vertices in the support of an alternating set of rank-2 edges in the boundary of the face can have nonzero a_v in b. Consequently these vertices belong to an alternating set of f-faces in the boundary of f.

Consider now the construction in Theorem 4, in this rank-3 edges are in the boundary of every v-face and e-face of Γ_2. Further, they are all connected. Consider two adjacent v-faces as shown in Fig. 12.

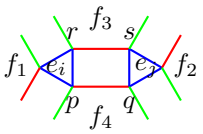


FIG. 12. (Color online) For the hypergraph in Theorem 4, if b has support on one rank-3 edge, then it has support on all rank-3 edges in Γ_h.

If $a_p\neq 0$, it implies that $a_r=0=a_s$ and $a_q\neq 0$. If the rank-3 edge e_j has support in b, then all the rank-3 edges incident on f_2 must also be present. Since all the v-faces are connected, b has support on all the rank-3 edges. Also note that the f-face f_3 has vertices in its boundary which are in the support of b. In order that no edge from its boundary is in the support of b, all the vertices in its boundary must be such that $a_v\neq 0$. The opposite holds for the vertices in f_4. None of these vertices must have $a_v\neq 0$. Thus the f-faces are portioned into two types and a consistent assignment of a_v is possible if and only if the f-faces form a bipartition. In other words, Γ^* is bipartite. Thus the additional linear dependency exists only when Γ^* is bipartite.

Let us now consider the graph in Theorem 5. In this case F and $\mathrm{F_c}\setminus\mathrm{F}$ form a bipartition. And only the the set v-faces in F have the rank-3 edges in their boundary. Consider two adjacent v-faces of Γ_2, $f_1\in\mathrm{F_c}\setminus\mathrm{F}$, $f_2\in\mathrm{F}$, as shown in Fig. 13.

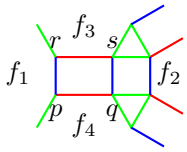


FIG. 13. (Color online) For the hypergraph in Theorem 5, if b has support on one rank-3 edge, then it has support on all rank-3 edges in Γ_h.

In this case $a_p=a_q=a_r=a_s$. So either all the vertices of f_1 are present or none at all. This creates a bipartition of the v-faces which are not having rank-3 edges in their boundary. Thus a consistent assignment of a_v is possible if and only if the rest of the v-faces in $F_c\setminus F$ form a bipartition. Since these are arising form the faces of Γ, this means that an additional linear dependency exists if and only if Γ^* is bipartite. □

Lemma 6. *Suppose that σ is a homologically nontrivial hyper cycle of Γ_h in Theorem 4 or 5. Then σ must contain some rank-3 edge(s).*

Proof. We use the notation as in Construction 2. We can assume that such a cycle does not contain a vertex from

$V(\Gamma_h) \setminus V(\Gamma_2)$. If such a vertex is part of the hyper cycle then all the vertices that belong to that v-face are also part of it and there exists another cycle σ' which consists of rank-2 edges and is not incident on the vertices in $V(\Gamma_h) \setminus V(\Gamma_2)$.

Suppose on the contrary that σ contains only rank-2 edges of Γ_h. In the hypergraphs of Theorem 4, every vertex in Γ_h has one rank-3 edge incident on it, further each vertex of Γ_h is trivalent and 3-edge colourable with the rank-3 edges all colored the same. Therefore, σ consists of rank-2 edges which are alternating in color. Every vertex is in the boundary of some f-face of Γ_2, say Δ. Note that an f-face does not have any rank-3 edge in its boundary although such an edge is incident on its vertices. This implies that σ is the boundary of Δ, therefore, homologically trivial cycle in contradiction our assumption. Therefore, σ must contain some rank-3 edges. This proves the statement for the graphs in Theorem 4.

Suppose now that σ is a cycle in the hypergraphs from Theorem 5. Now assume that there is a vertex in σ that is in the v-face which has rank-3 edges in its boundary. This edge is incident on two vertices which are such that the rank-3 edges are in the boundary while the rank-2 edges are out going and form the boundary of the 4-sided e-face incident on u, v. Therefore, the hyper cycle σ can be modified so that it is not incident on any v-face which has a rank-3 edge in its boundary. This implies from the e-faces only those edges are present in σ that are in the boundary of e-face and an v-face that has no rank-3 edges in its boundary. This edge is also coloured same the color of the f-faces in Γ_2. Further σ cannot have any edges that are of the same color as the v-faces. Thus σ must have the edges that are colored b and g the colors of the f-faces and e-faces respectively. But this implies that σ is the union of the boundaries of v-faces, because only if there are edges of r-type can it leave the boundary of a v-face. This contradicts that σ is non trivial homologically. □

Lemma 7. *Suppose that σ is a homologically nontrivial hyper cycle of Γ_h in Theorem 4 or 5. Then $W(\sigma)$ is not in the gauge group.*

Proof. Without loss of generality we can assume that σ has a minimal number of rank-3 edges in it. If not, we can compose it with another cycle in Δ_{Γ_h} to obtain one with fewer rank-3 edges. Note that $W(\sigma) \in \mathcal{G}$ if and only if $W(\sigma') \in \mathcal{G}$.

Assume now that $W(\sigma)$ is in the gauge group. Let E_2 be the set of rank-2 edges and E_3 be the set of rank-3 edges in Γ_h.

$$W(\sigma) = \prod_{e \in E_2 \cap \sigma} K_e \prod_{e \in E_3 \cap \sigma} K_e$$

The edges in $E_2 \cap \sigma$ are also edges in $\overline{\Gamma}_h$ and the associated link operators are the same. Therefore, it implies that Z-only operator $O_\sigma = \prod_{e \in E_3 \cap \sigma} K_e$ is generated by the gauge group consisting of operators of the form $\{X \otimes X, Y \otimes Y, Z \otimes Z\}$.

The operator O_σ consists of (disjoint) rank-3 edges alone and therefore, for any edge (u, v, w) in the support of O_σ, for each of the qubits u, v, w, one of the following must be true:

(i) Exactly one operator $Z_u Z_v$, $Z_v Z_w$, $Z_w Z_u$ is required to generate the $Z_i Z_j$ on a pair of the qubits, where $i, j \in \{u, v, w\}$. The Z operator on the remaining qubit is generated by gauge generators of the form $X_i X_j$ and $Y_i Y_k$, where i is one of $\{u, v, w\}$

(ii) The support on all the qubits is generated by $X_i X_j$ and $Y_i Y_k$, where i is one of $\{u, v, w\}$.

For a qubit not in the support of O_σ, either no generator acts on it or all the three gauge operators $X_u X_i$, $Y_u Y_j$, and $Z_u Z_v$ act on it. If it is the latter case, then it follows that u, v must be in the support of same rank-3 edge and that v is also not in the support of O_σ.

Suppose that we can generate O_σ as follows:

$$O_\sigma = K^{(x,y)} K^{(z)},$$

where $K^{(x,y)}$ consists of only operators of the form $X \otimes X, Y \otimes Y$ and $K^{(z)}$ only of operators of the form $Z \otimes Z$. From the Lemma 6, we see that the $O_\sigma K^{(z)}$ must be trivial homologically. The rank-3 edges incident on the support of $O_\sigma K^{(z)}$ are either in the support of σ or not.

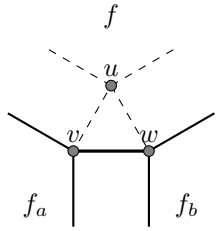


FIG. 14. A rank-3 edge which is not in the support of O_σ. The solid edges indicate the link operators which are in the support of $K^{(x,y)} K^{(z)}$, while the dashed edges do not. The edge must occur in two cycles, one which encloses f_a, and another which encloses f_b. If the same cycle encloses both f_a and f_b, then the edge occurs twice in that cycle. If we consider the stabilizer associated with these cycles then it has no support on this edge.

A rank-3 edge e which is not in σ must be such that exactly two vertices from e occur in the support of $O_\sigma K^{(z)}$. There are two faces f_a and f_b associated [18] with these two vertices, see Fig 14. There is a hypercycle that encloses f_a whose support contains e and another that encloses f_b and whose support contains e. The product of these two stabilizer elements has no support on e but has support on the edges in O_σ. We can therefore, find an appropriate combination which are associated with the trivial cycle in the support of $K^{(x,y)}$ such that σ has fewer rank-3 edges. But this contradicts the minimality of rank-3 edges in σ. Therefore, it is not possible to generate $W(\sigma)$ within the gauge group if σ is homologically nontrivial. □

IV. SYNDROME MEASUREMENT IN TOPOLOGICAL SUBSYSTEM CODES

One of the motivations for subsystem codes is the possibility of simpler recovery schemes. In this section, we show how the many-body stabilizer generators can be measured using only two-body measurements. This could help in lowering

the errors due to measurement and relax the error tolerance on measurement.

The proposed topological subsystem codes are not CSS-type unlike the Bacon-Shor codes. In CSS-type subsystem codes, the measurement of check operators is somewhat simple compared to the present case. The check operators are either X-type or Z-type. Suppose that we measure the X-type check operators first. We can simply measure all the X-type gauge generators and then combine the outputs classically to obtain the necessary stabilizer outcome. When we try to Z-type operator subsequently, we measure the Z-type gauge operators and once again combine them classically. This time around, the output of the Z-type gauge operators because they anti-commute with some X-type gauge operator we have uncertainty in the individual Z-type observables. Nonetheless because the Z-type check operator because it commutes with the X-type gauge generators, it can still be measured without inconsistency.

When we deal with the non-CSS type subsystem codes, the situation is not so simple. We need to find an appropriate decomposition of the stabilizers in terms of the gauge generators so that the individual gauge outcomes can be consistently. So it must be demonstrated that the syndrome measurement can be performed by measuring the gauge generators and a schedule exists for all the stabilizer generators. A condition that ensures that a certain decomposition of the stabilizer in terms of the gauge generators is consistent was shown in [22].

Theorem D (Syndrome measurement [22])**.** *Suppose we have a decomposition of a check operator S as an ordered product of link operators K_i such that*

$$S = K_m \cdots K_2 K_1 \text{ where } K_j \text{ is the link operator } K_{e_j} \quad (23)$$

$$[K_j, K_{j-1} \cdots K_1] = 0 \text{ for all } j = 2, \cdots, m \quad (24)$$

Let $s \in \mathbb{F}_2$ be the outcome of measuring S. Then to measure S, measure the link operators K_i for $i = 1$ to m and compute $s = \oplus_{i=1}^{m} g_i$, where $g_i \in \mathbb{F}_2$ is the outcome of measuring K_i.

Theorem 9. *The syndrome measurement of the subsystem codes in Theorems 4 and 5 can be performed in three rounds using the following procedure, using the decompositions given in Fig. 5, 7, for Theorem 4 and Fig. 8, 10 for Theorem 5.*

(i) Let a stabilizer generator $W(\sigma) = \prod_i K_i \in S$ be decomposed as follows

$$W(\sigma) = \prod_{i \in E_r} K_i \prod_{j \in E_g} K_j \prod_{k \in E_b} K_k = S_b S_g S_r \quad (25)$$

where K_i is a link operator and E_r, E_g, E_b are the link operators corresponding to edges coloured r, g, b respectively.

(ii) Measure the gauge operators corresponding to the edges of different color at each level.

(iii) Combine the outcomes as per the decomposition of $W(\sigma)$.

Proof. In the subsystem codes of Theorem 4, there are two stabilizer generators associated with the v-face and f-face. Those associated with the v-face are shown in Fig. 5. Consider the first type of stabilizer generator $W(\sigma_1)$. Clearly, $W(\sigma_1)$ consists of two kinds of link operators, r type and g type. The link operators corresponding to the r-type edges are all disjoint and can therefore be measured in one round. In the second round, we can measure the link operators corresponding to g-type edges. Since this is an even cycle we clearly have $[S_g, S_r] = 0$. Note that $E_b = \emptyset$ because there are no b-edges in σ_1. A similar reasoning holds for the generator $W(\sigma_1)$ shown in Fig. 7 corresponding to an f-face.

For the second type of the stabilizer generators $W(\sigma_2)$, observe that as illustrated in Fig. 5, the r-edges are disjoint with the "outer" b and g-edges and can be measured in the first round. The "outer" g-edges being disjoint with the r-edges we satisfy the condition $[S_g, S_r] = 0$. In the last round when we measure the b-edges, since the b-edges and g-edges overlap an even number of times and being disjoint with the r-edges we have $[S_b, S_g S_r] = 0$. Thus by Theorem D, this generator can be measured by measuring the gauge operators.

The same reasoning can be used to measure $W(\sigma_2)$ corresponding to the f-faces, but with one difference. The outer edges are not all of the same color, however this does not pose a problem because in this case as well we can easily verify that $[S_g, S_r] = 0$, because they are disjoint. Although the b-edges overlap with both the r and g-edges note that each of them individually commutes with $S_g S_r$ because they overlap exactly twice. Thus $[S_b, S_g S_r] = 0$ as well and we can measure $W(\sigma_2)$ through the gauge operators.

Syndrome measurement of two disjoint stabilizers do not obviously interfere with each other. However, when two generators have overlapping support, they will not interfere as demonstrated below.

Note that every every vertex of Γ_h in Theorem 4 has a rank-3 edge incident on it. As illustrated in Fig. 15, edges which are not shared are essentially the rank-3 edges and each one of them figures in only one of the stabilizer generators, but because they all commute they can be measured in the same round. The r and g edges are shared and appear in the support stabilizer generators of two adjacent faces. Nonetheless because edges of each color are disjoint they can be measured simultaneously. As has already been demonstrated the edges of each color are such that for each stabilizer generator $[S_g, S_r] = 0$ and $[S_b, S_g S_r] = 0$.

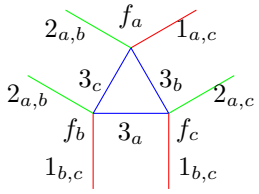
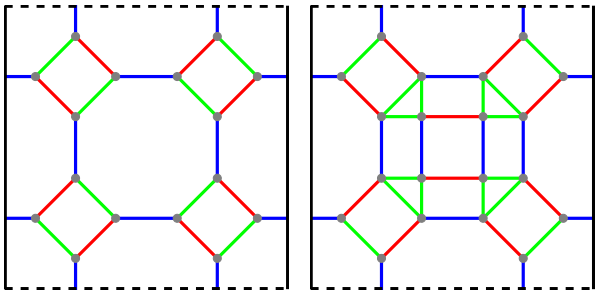


FIG. 15. Noninterference of syndrome measurement. The faces f_a, f_b, f_c have stabilizer generators that have overlapping support. The edges are labelled with the round in which they are measured, the subscripts indicate the faces with which the edge is associated. Thus 3_a indicates that this edge should be measured in the third round and it is used in the stabilizer generator $W(\sigma_2)$ of the face f_a.

A similar argument can be made for the codes in Theorem 5, the proof is omitted. □

The argument above shows that the subsystem codes of

Theorems 4 and 5 can be measured in three rounds using the same procedure outlined in Theorem 9 if we assume that a single qubit can be involved in two distinct measurements simultaneously. If this is not possible, then we need four time steps to measure all the checks. The additional time step is due to the fact that a rank-3 edge results in three link operators. However only two of these are independent and they overlap on a single qubit. To measure both operators, we need two time steps. Thus the overall time complexity is no more than four time steps. This is in contrast to the schedule in [6], which takes up to six time steps.



(a)2-colex. (b)Subsystem code.

FIG. 16. (Color online) A subsystem code in which some of the stabilizer generators are nonlocal. This is derived from the color code on a torus from a square octagon lattice. Opposite sides are identified.

V. CONCLUSION AND DISCUSSION

A. Significance of proposed constructions

To appreciate the usefulness of our results, it is helpful to understand Theorem B in more detail. First of all consider the complexity of finding hypergraphs which satisfy the requirements therein. Finding if a cubic graph is 3-edge-colorable is known to be NP-complete [10]. Thus determining if a 3-valent hypergraph is 3-edge-colorable is at least as hard. In view of the hardness of this problem, the usefulness of our results becomes apparent. One such family of codes is due to [6]. In this paper we provide new families of subsystem codes. Although they are also derived from color codes, they lead to subsystem codes with different parameters. With respect to the results of [22], our results bear a relation similar to a specific construction of quantum codes, say that of the Kitaev's topological codes, to the CSS construction.

Secondly, the parameters of the subsystem code constructed using Theorem B depend on the graph and the embedding of the graph. They are not immediately available in closed form for all hypergraphs. We give two specific families of hypergraphs where the parameters can be given in closed form. In addition our class of hypergraphs naturally includes the hypergraphs arising in Bombin's construction.

Thirdly, Theorem B does not distinguish between the case when the stabilizer is local and when the stabilizer is nonlocal. Let us elaborate on this point. The subsystem code on the honeycomb lattice, for instance, can be viewed as a hypergraph albeit with no edges of rank-3. In the associated subsystem code the stabilizer can have support over a cycle which is nontrivial homologically. In fact, we can even provide examples of subsystem codes derived from true hypergraphs in that there exist edges or rank greater than two, and whose stabilizer can have elements associated to nontrivial cycles of the surface. Consider for instance, the 2-colex shown in Fig. 16(a). The hypergraph derived from this 2-colex is shown in Fig. 16(b). This particular code has a nonzero-rate even though its stabilizer includes cycles that are not nontrivial homologically.

In contrast the subsystem codes proposed by Bombin, all have local stabilizers. It can be conceded that the locality of the stabilizer simplifies the decoding for stabilizer codes. But this is not necessarily a restriction for the subsystem codes. A case in point is the family of Bacon-Shor codes which have non-local stabilizer generators. It would be important to know what effect the non-locality of the stabilizer generators have on the threshold. Although we do not provide a criterion as to when the subsystem codes are topological in the sense of having a local stabilizer, our constructions provide a partial answer in this direction. It would be certainly more useful to have this criterion for all the codes of Theorem B.

Not every cubic graph can allow us to define a subsystem code. Only if it satisfies the commutation relations, namely Eq. (8) is it possible. As pointed out in [22], the bipartiteness of the graph plays a role. The Petersen graph satisfies H1–4 being a cubic graph but with no hyperedges. But it does not admit a subsystem code to be defined because there is no consistent assignment of colours that enables the definition of the gauge group. In other words, we cannot assign the link operators such that Eq. (8) are satisfied. We therefore, add the 3-edge- colorability requirement to the hypergraph construction of the Suchara et al. [22].

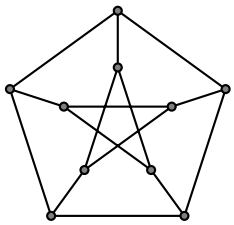


FIG. 17. The Petersen graph although cubic and satisfying H1–4, does not lead to a subsystem code via Construction C; it is not 3-edge colorable.

Fig.18 illustrates our contributions in relation to previous work.

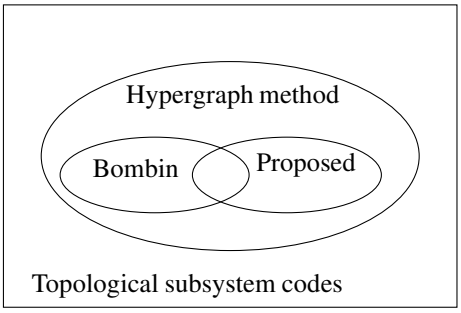


FIG. 18. Proposed constructions in context. Note that some of the hypergraph based subsystem codes may have homologically nontrivial stabilizer generators.

ACKNOWLEDGMENTS

This work was supported by the Office of the Director of National Intelligence - Intelligence Advanced Research Projects Activity through Department of Interior contract D11PC20167. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

[1] P. Aliferis and A. W. Cross. Sub-system fault tolerance with the bacon-shor code. *Phys. Rev. Lett.*, 98(220502), 2007.

[2] R. S. Andrist, H. Bombin, H. G. Katzgraber, and M. A. Martin-Delgado. Optimal error correction in topological subsystem codes. *Phys. Rev. A*, 85(050302), 2012.

[3] D. Bacon. Operator quantum error correcting subsystems for self-correcting quantum memories. *Phys. Rev. A*, 73(012340), 2006.

[4] B. Bombin and M. A. Martin-Delgado. Topological quantum distillation. *Phys. Rev. Lett.*, 97(180501), 2006.

[5] B. Bombin and M. A. Martin-Delgado. Exact topological quantum order in d = 3 and beyond: Branyons and brane-net condensates. *Phys. Rev. B*, 75(075103), 2007.

[6] H. Bombin. Topological subsystem codes. *Phys. Rev. Lett.*, 81(032301), 2010.

[7] S. Bravyi and B. Terhal. A no-go theorem for a two-dimensional self-correcting quantum memory based on stabilizer codes. *New J. Phys.*, 11(043029), 2009.

[8] A.R. Calderbank, E.M. Rains, P.W. Shor, and N.J.A. Sloane. Quantum error correction via codes over GF(4). *IEEE Trans. Inform. Theory*, 44:1369–1387, 1998.

[9] D. Gottesman. Stabilizer codes and quantum error correction. Caltech Ph. D. Thesis, eprint: quant-ph/9705052, 1997.

[10] I. Holyer. The NP-completeness of edge coloring. *SIAM J. Comput.*, 10(4):718 –720, 1981.

[11] A. Klappenecker and P. K. Sarvepalli. Clifford code constructions of operator quantum error-correcting codes. *IEEE Transactions on Information Theory*, 54(12):5760–5765, 2008.

[12] D. W. Kribs. A brief introduction to operator quantum error correction. Eprint: math/0506491, 2005.

[13] D. W. Kribs, R. Laflamme, and D. Poulin. Unified and generalized approach to quantum error correction. *Phys. Rev. Lett.*, 94(180501), 2005.

[14] D. W. Kribs, R. Laflamme, D. Poulin, and M. Lesosky. Operator quantum error correction. Eprint: quant-ph/0504189, 2005.

[15] A. J. Landahl, J. T. Anderson, and P. R. Rice. Fault-tolerant quantum computing with color codes. eprint:arXiv:1108.5738, 2011.

[16] There are various other definitions of hypercycles, see for instance [20] for an overview.

[17] Condition H3 implies that the hypergraphs that we are interested are also reduced hypergraphs. A reduced hypergraph is one in which no edge is a subset of another edge.

[18] The two faces f_a and f_b could be the same face. In this case the associated cycle contains both the vertices.

[19] D. Poulin. Stabilizer formalism for operator quantum error correction. *Phys. Rev. Lett.*, 95(230504), 2005.

[20] Duke R. Types of cycles in hypergraphs. *North-Holland Mathematics Studies*, 115:399–417, 1985.

[21] R. Raussendorf and H. J. Harrington. Fault-tolerant quantum computation with high threshold in two dimensions. *Phys. Rev. Lett.*, 98(150504), 2001.

[22] M. Suchara, S. Bravyi, and B. Terhal. Constructions and noise threshold of topological subsystem codes. *Journal of Physics A*, 3, 2011.

[23] D. S. Wang, A. G. Fowler, C. H. Hill, and L. C. L. Hollenberg. Graphical algorithms and threshold error rates for the 2d color code. *Quantum Information and Computation*, 10(9):780–802, 2010.