Finite-rate sparse quantum codes aplenty

Maxime Tremblay, Guillaume Duclos-Cianci, and Stefanos Kourtis

Département de physique & Institut quantique, Université de Sherbrooke, Sherbrooke, Québec, Canada, J1K 2R1

We introduce a methodology for generating random multi-qubit stabilizer codes based on solving a constraint satisfaction problem (CSP) on random bipartite graphs. This framework allows us to enforce stabilizer commutation, X/Z balancing, finite rate, sparsity, and maximum-degree constraints simultaneously in a CSP that we can then solve numerically. Using a state-of-the-art CSP solver, we obtain convincing evidence for the existence of a satisfiability threshold. Furthermore, the extent of the satisfiable phase increases with the number of qubits. In that phase, finding sparse codes becomes an easy problem. Moreover, we observe that the sparse codes found in the satisfiable phase practically achieve the channel capacity for erasure noise. Our results show that intermediate-size finite-rate sparse quantum codes are easy to find, while also demonstrating a flexible methodology for generating good codes with custom properties. We therefore establish a complete and customizable pipeline for random quantum code discovery that can be geared towards near- to mid-term quantum processor layouts.

1 Introduction

Quantum error-correcting codes are an essential prerequisite of reliable quantum computing. While codes like surface codes [10, 18, 30] and color codes [25, 26] are sufficient to encode and protect a small constant number of qubits, their performance degrades rapidly when performing computations with more qubits [12, 15]. While quantum error-correcting codes with asymptotically good performance have been known to exist for more than 20 years [7, 14], these codes require measurement of operators whose weight increases with block size. This is a major limitation to efficient implementation of measurement circuits for these codes.

More recently, Gottesman showed that finite-rate quantum low-density parity-check (LDPC) codes can be used to build constant-overhead fault-tolerant quantum computers [21]. Quantum LDPC codes are stabilizer codes with bounded-weight stabilizers. That is, they involve only bounded-weight measurements.

This seminal result together with the plethora of asymptotically optimal classical LDPC code constructions [19, 28] led to a surge of research toward quantum LDPC codes inspired by their classical counterparts. This resulted in many quantum LDPC code constructions such as hypergraph product codes [31], homological product codes [11] and fiber bundle codes [24]. Finally, more than 20 years after the introduction of the first quantum error-correcting codes, Panteleev and Kalachev [27] introduced the first family of quantum LDPC codes with optimal asymptotic performance.

These constructions are based on homological products of classical or quantum error-correcting codes. While there is some flexibility in the choice of the input codes, these methods are hard to adapt to precise connectivity limitations of near to mid-term quantum devices. Furthermore, they are generally more relevant for a larger number of qubits. For example, recent numerical studies of hypergraph product codes [22, 32] involve a few thousands to hundreds of thousands of qubits. Other constructions such as building codes from low-depth random circuits [13, 23], have been proposed but, to our knowledge, no procedure generates the stabilizer operators directly while considering arbitrary physical limitations.

Stefanos Kourtis: stefanos.kourtis@usherbrooke.ca

In this work, we introduce a methodology for the generation of an arbitrary number of stabilizer operators directly for any given number of qubits. We reformulate code generation as a constraint satisfaction problem (CSP) on random bipartite graphs of varying edge inclusion probability, whose vertices correspond to qubits and stabilizers. The constraints imposed correspond to desired code properties, including, but not limited to, stabilizer commutation, X/Z balancing, finite rate, sparsity, and maximum-degree bounds. The resulting CSP is akin to paradigmatic NP-complete problems like random k-SAT and random 2-coloring of k-uniform hypergraphs. Although these CSPs are hard to solve in the worst case, they often display a transition between a hard phase, where problem instances are hard to solve on average, and an easy phase, within which typical instances can be solved in polynomial time, as a function of some parameter [5, 6]. This motivates us to ask whether our code generation CSP on random bipartite graphs also exhibits a transition to an easy phase as a function of some parameter, which in the present case we choose to be the edge inclusion probability of the sampled graphs.

To define our problem instances, we sample bipartite graphs at random with varying edge inclusion probability, define appropriate constraints on qubit and stabilizer vertices that ensure our desired code properties, and then solve the resulting instances using a state-of-the-art CSP solver. We discover a satisfiability threshold at a critical edge inclusion probability that decreases with increasing number of qubits. We therefore show that the CSP has an easy phase, whose extent grows with increasing number of qubits and within which we readily find finite-rate stabilizer codes. Furthermore, we find that the resulting codes are much sparser than the initial graphs. Going a step further, we show that we can guarantee code sparsity via maximum-degree constraints, at the expense of a computational overhead. Finally, we demonstrate that the codes we obtain using our methodology achieve the erasure channel capacity. We therefore establish a complete and customizable pipeline for random quantum code discovery that can be geared towards near- to mid-term quantum processor layouts.

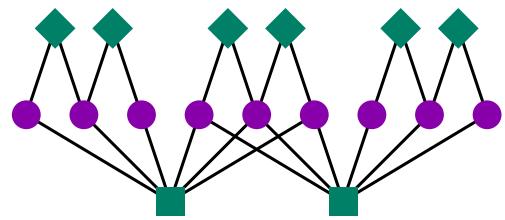

Figure 1: The Tanner graph of the 9-qubit Shor code. Purple circles represent qubit vertices and green squares and diamonds represent respectively X and Z stabilizer generators. The top right stabilizer vertex corresponds to the Z_7Z_8 operator and the bottom left vertex corresponds to the $X_0X_1X_2X_3X_4X_5$ operator.

The rest of this paper is organized as follows. In Section 2, we recall notions of quantum coding theory and stabilizer code construction. In Section 4, we introduce our methodology for the generation of stabilizer codes through solving a CSP on random bipartite graphs, detailing the implementation of constraints. Finally, we present numerical results on code discovery and near-optimal error correction performance for the erasure channel in Section 5.

2 Stabilizer codes

Given the n-qubit Pauli group $\mathcal{P}_n$, a stabilizer group is a commuting subgroup of $\mathcal{P}_n$ that does not include the $-I$ operator. A stabilizer group $\mathcal{S}$ defines the stabilizer code [20]

$$\mathcal{C}(\mathcal{S}) = \{|\psi\rangle \mid S|\psi\rangle = |\psi\rangle, \forall S \in \mathcal{S}\}. \tag{1}$$

That is, a stabilizer code is the common $+1$ eigenspace of the operators of a stabilizer group. We often define a stabilizer group using a set of generators $g(\mathcal{S}) = \{S_1, S_2, ..., S_m\}$ with $S_i \in \mathcal{S}$.

A family of codes is a finite-rate family if there exists a constant $c > 0$ such that $\frac{n-m}{n} > c$ for $n \to \infty$. Finite-rate code families are crucial for achieving large-scale fault-tolerant quantum computation since they allow encoding a constant ratio of logical qubits to physical qubits with vanishing error rate. This is in contrast with zero-rate code families such as surface codes, which can only encode a small constant number of log-

ical qubits without significantly degrading error-correcting performance [12].

A common class of stabilizer codes are Calderbank-Shor-Steane (CSS) codes [14, 29]. These are codes for which there exists a set of generators such that each of them is a product of only I and X or only I and Z. While the techniques introduced in this work are applicable to both CSS and non-CSS stabilizer codes alike, in what follows we focus mainly on CSS codes as this simplifies the presentation of our methodology. We discuss the case of non-CSS stabilizer codes in Sec. 6.

Below we make use of the graphical representation of stabilizer codes based on Tanner graphs, illustrated in Figure 1. A Tanner graph $T = (Q \cup g(\mathcal{S}), E)$ is a bipartite graph that contains edge $\{q, S\} \in E$ if and only if the stabilizer generator S acts as X, Y or Z on qubit q. The degree of a vertex is its number of neighbors. In error-correction terms, the degree of a stabilizer corresponds to its weight and the degree of a qubit corresponds to the number of stabilizers acting on it.

3 Stabilizing edge coloring

To introduce our algorithmic approach to principled search for finite-rate stabilizer codes, we reformulate the task as a graph coloring problem. Our strategy is to start with a bipartite graph $G = (Q \cup g(\mathcal{S}), E_0)$, called the support graph, then search for an edge coloring $l : E_0 \to \{I, X, Y, Z\}$ for which all stabilizers commute. We call such a coloring a stabilizing edge coloring. The result of this procedure is a Tanner graph $T = (Q \cup g(\mathcal{S}), E)$ where the edge $e \in E$ iff $l(e) \neq I$ and hence $E \subseteq E_0$. Deciding whether there exists a stabilizing edge coloring for a given support graph G is equivalent to determining whether there exists a valid stabilizer code whose Tanner graph is a subgraph of G as defined above.

As formulated above, stabilizing edge coloring admits trivial solutions. For example, assigning the same color to all edges is always a valid solution, but is of little interest for quantum error correction. To avoid such trivial solutions, or other undesirable solutions, we must introduce additional constraints. We represent a constraint as a pair (F, L) where $F \subseteq E_0$ and $L \subseteq \{I, X, Y, Z\}^{|F|}$. A coloring l satisfies a constraint (F, L) for $F = \left\{e_1, \ldots, e_{|F|}\right\}$ if $(l(e_1), ..., l(e_{|F|})) \in L$. We can now define the constrained stabilizing edge coloring problem we are interested in.

Definition 1 ((Constrained) stabilizing edge coloring)**.**
Given
a bipartite graph $G = (Q \cup g(\mathcal{S}), E_0)$ and a set of constraints $\mathcal{F} = \left\{(F_1, L_1), \ldots, (F_{|\mathcal{F}|}, L_{|\mathcal{F}|})\right\}$ with $F_i \subseteq E_0$ and $L_i \subseteq \{I, X, Y, Z\}^{|F_i|}$,
find
a stabilizing edge coloring $l : E_0 \to \{I, X, Y, Z\}$ **satisfying** $\mathcal{F}$.

The problem is constrained whenever $\mathcal{F}$ is not empty. Table 1 presents estimates of $|\mathcal{F}|$ for the instances we study numerically. This defines a rich class of constraint satisfaction problems. In the following section, we provide concrete realisations of unconstrained and constrained stabilizing edge coloring which we solve to generate non-trivial CSS codes.

4 CSS codes from constraints on boolean variables

For CSS codes, we simplify stabilizing edge coloring by assigning Pauli values $l_p : g(\mathcal{S}) \to \{X, Z\}$ to stabilizer vertices and boolean values $l_a : E_0 \to \{0, 1\}$ to edges. Then, an edge $e \in E_0$ has color $l(e) = I$ when $l_a(e) = 0$ and color $l_p(e)$ otherwise. We say that an edge e is active if $l_a(e) = 1$. In this representation, the active neighborhood of a stabilizer S is the set of qubits for which $\{q, S\} \in E_0$ and $l_a(\{q, S\}) = 1$. Then, two stabilizers with different Pauli values commute if the intersection of their active neighborhoods has even cardinality.

To solve the stabilizing edge coloring problem numerically, we represent the coloring functions l_p and l_a with boolean variables. To each edge $e \in E_0$, we assign a boolean variable $\boldsymbol{v_a}(e) \in \{0, 1\}$ such that $l_a(e) = \boldsymbol{v_a}(e)$. To each stabilizer $S \in g(\mathcal{S})$, we assign a boolean variable $\boldsymbol{v_p}(S)$ such that $l_p(S) = X$ if $\boldsymbol{v_p}(S) = 1$ and $l(S) = Z$ otherwise. We call $\boldsymbol{v_a}$ an activator variable and $\boldsymbol{v_p}$ a Pauli variable. We also

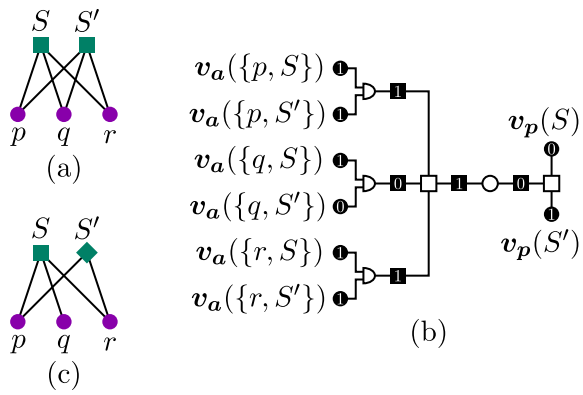

Figure 2: Graphical representation of the commutation constraints for a pair of stabilizers sharing three qubits. (a) The support graph G. The circles and squares respectively represent qubit and stabilizers. (b) The boolean variables and constraints assuring the commutation. Filled vertices correspond to variables. In particular circles are either activator or Pauli variables while the squares are auxiliary variables. The circle constraints correspond to Equation (2), the square constraints to Equations (3) and (4) and the half-circle constraints to Equation (5). The numbers within variable vertices illustrate a valid assignment. (c) The resulting Tanner graph according to the variable assignment.

introduce some auxiliary boolean variables. The auxiliary variable $\boldsymbol{v_s}(S,S')=1$ when the action of S and S' is the same. The variable $\boldsymbol{v_e}(S,S')=1$ when S and S' have an even overlap. Finally, the variable $\boldsymbol{v_b}(q,S,S')=1$ when both edges $\{q,S\}$ and $\{q,S'\}$ are active.

We are now ready to introduce the boolean constraints that define the stabilizing edge coloring problem for CSS codes. In the rest of this section, we represent a constraint as a boolean function $\boldsymbol{c}:\{0,1\}^*\to\{0,1\}$ acting on a subset of the boolean variables $\boldsymbol{v_a}$, $\boldsymbol{v_p}$, $\boldsymbol{v_s}$, $\boldsymbol{v_e}$, and $\boldsymbol{v_b}$. An assignment $\mathbf{x}\in\{0,1\}^*$ satisfies a constraint if $\boldsymbol{c}(\mathbf{x})=1$. Boolean constraints allow us to represent the stabilizing edge coloring constraints of Definition 1.

Stabilizers S and S' commute if they satisfy at least one of the following conditions: they have the same non-trivial action (X or Z) or, as discussed previously, they act non-trivially on an even number of common-neighbor qubits. We can thus write a commutation constraint $\boldsymbol{c_c}(S,S')$ as

$$\boldsymbol{c_c}(S,S')=\boldsymbol{v_s}(S,S')\vee\boldsymbol{v_e}(S,S'). \tag{2}$$

That the variable $\boldsymbol{v_s}(S,S')=1$ when the action of S and S' is the same is enforced by the constraint

$$\boldsymbol{c_s}(S,S')=\boldsymbol{v_s}(S,S')\oplus\boldsymbol{v_p}(S)\oplus\boldsymbol{v_p}(S'). \tag{3}$$

Similarly, to ensure that $\boldsymbol{v_e}(S,S')=1$ when S and S' have an even overlap, we add the constraint

$$\boldsymbol{c_e}(S,S')=\boldsymbol{v_e}(S,S')\oplus\left[\bigoplus_{q\in\eta(S)\cap\eta(S')}\boldsymbol{v_b}(q,S,S')\right], \tag{4}$$

where $\eta(v)$ is the set of neighbors of vertex v in G. Finally, to ensure that $\boldsymbol{v_b}(q,S,S')=1$ when both edges $\{q,S\}$ and $\{q,S'\}$ are active, we add the constraint

$$\boldsymbol{c_b}(q,S,S')=\neg\boldsymbol{v_b}(q,S,S')\oplus [\boldsymbol{v_a}(\{q,S\})\wedge\boldsymbol{v_a}(\{q,S'\})]. \tag{5}$$

Equations (2) to (5) define a set of constraints on activator, Pauli and auxiliary variables. We draw the variables and constraints for a small example in Figure 2. When simultaneously satisfied, they ensure that the coloring function l defines a stabilizing edge coloring. Thus, any variable assignment for which all constraints evaluate to 1 yields a valid CSS code. In Section 4.2, we present how we find such assignments.

4.1 Extra constraints for good codes

In this section, we introduce extra constraints to restrict the search to better codes. We represent these contraints using integer linear functions $\boldsymbol{c}:\mathbb{Z}^*\to\mathbb{Z}$ restricted to $D\subseteq\mathbb{Z}$. That is, an assignment $\mathbf{x}\in\mathbb{Z}^*$ satisfies constraint $\mathbf{c}$ if $\mathbf{c}(\mathbf{x})\in D$. We define these constraints by extending the boolean variables of the previous section to be integer variables.

We first add lower bounds on the number of stabilizers of each kind connected to a qubit. For each edge $\{q,S\}\in E_0$, we add a variable $\boldsymbol{v_X}(\{q,S\})$ with value 1 if the edge is active and the corresponding stabilizer is of the X type. This is enforced by the constraint

$$\boldsymbol{c_X}(q,S)=\neg\boldsymbol{v_X}(q,S)\oplus[\boldsymbol{v_a}(\{q,S\})\wedge\boldsymbol{v_p}(S)]. \tag{6}$$

Then, we impose that the number of variables $\boldsymbol{v_X}(\{q, S\})$ with value 1 for $S \in \eta(q)$ is at least δ_q. That is, we add the constraint

$$\boldsymbol{c_X}(q) = \sum_{S \in \eta(q)} \boldsymbol{v_X}(q, S) \geq \delta_q \tag{7}$$

for each qubit q. We use similar constraints to lower bound the number of Z stabilizer generators per qubit.

These constraints together with the commutation constraints are the first set of constraints we numerically study in the following section. Subsequently, we add more constraints to search for codes with improved decoding performances.

We impose to each stabilizer $S \in g(\mathcal{S})$ that the number of variables $\boldsymbol{v_a}(\{q, S\})$ with value 1 for $q \in \eta(S)$ is at least δ_s. Then, to keep the codes sparse, we impose that at most Δ_S edges per stabilizer are active. These are both enforced by constraints akin to Equation (7).

Finally, we impose that the number of stabilizers of each kind is balanced by enforcing that

$$\sum_{S \in g(\mathcal{S})} \boldsymbol{v_p}(S) = \left\lfloor \frac{|g(\mathcal{S})|}{2} \right\rfloor . \tag{8}$$

4.2 Constraint satisfaction problem solver

To obtain CSS codes from the aforementioned constraints, we use the *OR-Tools* library [3]. We decompose the constraints of Equations (2), (5), and (6) into a constant number of OR constraints and we leave those of Equations (3) and (4) as XOR constraints. Both OR and XOR constraints, as well as the linear constraints used to enforce the minimum and maximum degree and balancing, are native to the library. We provide our implementation in an online repository [4].

5 Results

5.1 Phase transition

To search for codes with n qubits and m stabilizer generators, we start by building random support graphs with the corresponding numbers of vertices using the Erdős–Rényi model [17]. That is, we sample the graphs with an edge inclusion probability of

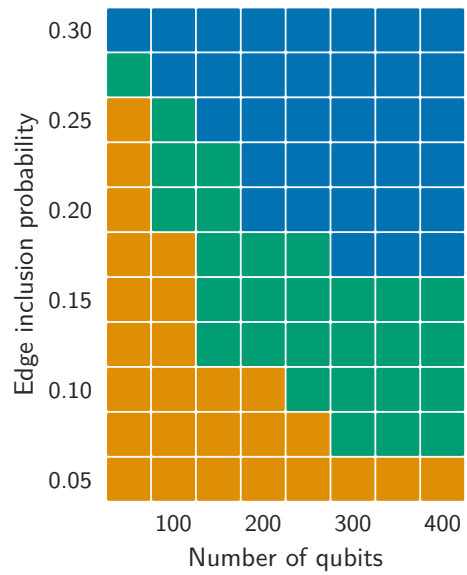

Figure 3: Satisfiability phase diagram for the commutation and minimum qubit degree constraints. For each pixel we generate 100 support graphs, then solve the corresponding stabilizing edge coloring instance for each graph. For each support graph, we run the CSP solver on four CPU cores running at 2.4 GHz for up to four hours. A green pixel indicates a combination of edge inclusion probability and number of qubits for which the solver is able to solve less than 10% of the instances within the allocated timeout. In the rest of the phase diagram, where more than 90% of the instances are solved for each combination of parameters, a blue pixel indicates more satisfiable than unsatisfiable instances, whereas an orange pixel indicates the opposite: more unsatisfiable than satisfiable instances.

γ. We use $G_{n,m,\gamma}$ to denote the corresponding random bipartite graph generator. Our goal is to find sparse codes for given n, m, and γ, or obtain a proof that such codes are statistically unlikely.

One can see that if $E \subseteq E'$ and $G = (V, E) \in \mathcal{P}$, where $\mathcal{P}$ denotes the existence of at least one stabilizing edge coloring, then for any graph $G' = (V, E')$, we have $G' \in \mathcal{P}$. Thus,

$$\Pr[G_{n,m,\gamma} \in \mathcal{P}] \leq \Pr[G_{n,m,\gamma'} \in \mathcal{P}] \tag{9}$$

when $\gamma \leq \gamma'$ and there exists a threshold function

	Type	Weight	Occurrence
Stab. edge color.	OR	3	$nm^2\gamma^2$
	OR	2	m^2
	XOR	$n\gamma^2$	m^2
	XOR	3	m^2
X/Z qubit deg.	Linear	$m\gamma$	n
	OR	3	$nm\gamma$
Stab. deg.	Linear	$n\gamma$	m
Balancing	Linear	m	1

Table 1: Expected constraint occurrences and weights. The data is obtained considering there are $\mathcal{O}(m^2)$ pairs of stabilizers each expected to share $\mathcal{O}(n\gamma^2)$ common qubits.

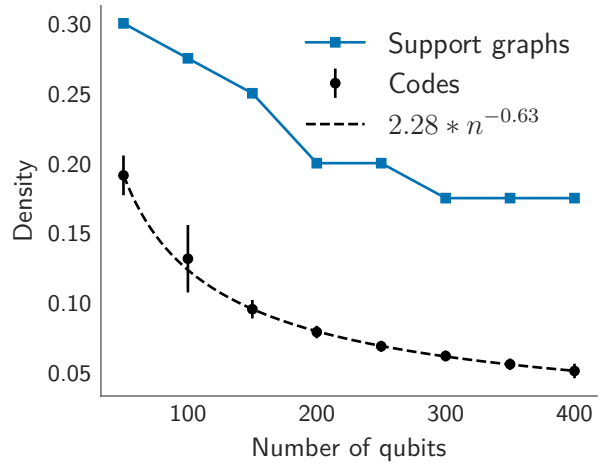

Figure 4: Densities of the resulting codes for commutation and minimum qubit degree constraints. For the support graphs, we plot the minimum edge inclusion probability in the satisfiable region. For the codes, we plot the average density over all solutions with the same number of qubits found in the satisfiable phase of Figure 3.

$\gamma^*(n)$ such that [9]

$$\lim_{n\to\infty} \Pr[G_{n,m,\gamma} \in \mathcal{P}] = \begin{cases} 0 & \gamma(n)/\gamma^*(n) \to 0, \\ 1 & \gamma(n)/\gamma^*(n) \to \infty. \end{cases} \tag{10}$$

Our numerical results, discussed below, suggest this threshold is a non-increasing function of the number of qubits.

We start by imposing only the commutation constraints and a minimum qubit degree. That is, we impose $\delta_q = 3$, $\delta_s = 0$ and $\Delta_s = \infty$ for graphs with fixed ratio $m/n = 9/10$, yielding codes with rate 1/10. If no solution was found and no proof of unsatisfiability was produced within a timeout, we label the instance as unknown. We note that most instances are solved in a fraction of the timeout, except close to the threshold. In Table 1 we give estimates for the number and weight (in number of variables) of each type of constraint in the CSS stabilizing edge coloring for the Erdős–Rényi model.

Figure 3 illustrates that the threshold γ^* is decreasing with the number of qubits. While we cannot definitively claim that the threshold does not start increasing for larger block sizes, we expect that finite-size effects are insignificant for larger numbers of qubits. This implies that γ^* either decreases monotonically or it plateaus to a value of at most 15%. The unknown region demarcates the parameter regime where typical instances of the CSP become hard to solve and our calculations time out. Crucially, Figure 4 shows that the densities $\frac{E}{mn}$ of the Tanner graphs of the codes go to zero as the number of qubits increase and are much lower than the edge inclusion probabilites of the support graphs. We observe similar results for different values of m/n.

We now restrict the problem further to the regime of quantum LDPC codes. That is, we impose $\delta_q = 3$, $\delta_s = 6$ and $\Delta_S = 20$ together with the stabilizer balancing constraint. Figure 5 indicates that the threshold function is non-increasing when increasing the number of qubits. We thus once again find a satisfiable phase that expands with increasing number of qubits and within which it is statistically likely that quantum LDPC codes exist and are easy to find.

5.2 Decoding experiment

We showed numerical evidence that even if it is generally hard to find commuting sets of random stabilizer operators, it is relatively easy to generate such a set from a random bipartite graph as long as the edge inclusion probability of the graph is above some threshold function. However, this result implies nothing about the decoding performance of

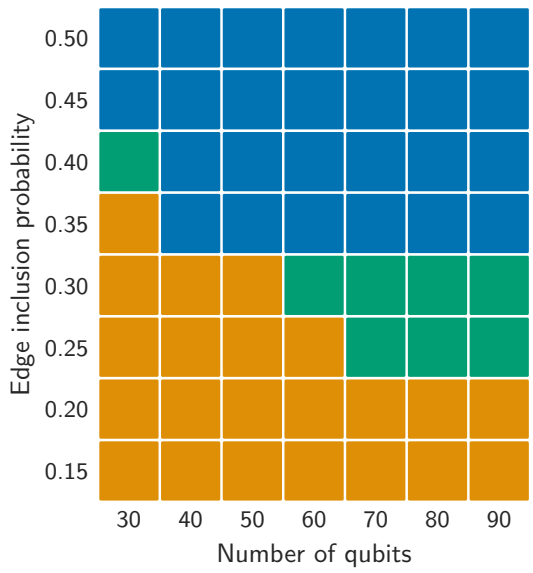

Figure 5: Satisfiability phase diagram with qubit degree, stabilizer degree, and balancing constraints. All parameters and details are the same as in Figure 3.

the code generated this way.

In this section, we probe the performance of the codes we find, using the erasure channel as suggested in [16, 23]. The erasure channel is particularly useful since there exists a maximum-likelihood decoder that runs in polynomial time. In contrast, we do not yet have a good universal decoder for the depolarizing channel.

The single-qubit erasure channel,

$$\mathcal{E}_p(\rho) = (1-p)\rho \otimes |0\rangle\langle 0| + p\frac{I}{2} \otimes |1\rangle\langle 1|\,, \tag{11}$$

erases a qubit with probability p by replacing it with a maximally mixed single-qubit state. The second register indicates whether a qubit has been erased. Since

$$\frac{I}{2} = \frac{1}{4}(I\rho I + X\rho X + Y\rho Y + Z\rho Z), \tag{12}$$

the multi-qubit version of the channel can be written as

$$\mathcal{E}_p^n(\rho) = \sum_{\mathbf{e}\in\{0,1\}^n} \Pr[\mathbf{e}] \left(\frac{1}{4^{|\mathbf{e}|}} \sum_{E\in P_n(\mathbf{e})} E\rho E \right) \otimes |\mathbf{e}\rangle\langle \mathbf{e}|\,, \tag{13}$$

where $P_n(\mathbf{e})$ is the set of n-qubit Pauli operators with a trivial action on every qubit q_i for which $e_i = 0$. Then, after a measurement of the second register identifying erasure positions, the channel reduces to a Pauli channel on the erased qubits where each error is equally likely. Therefore, from a measurement of the syndrome, a maximum likelihood decoder searches for any Pauli operator restricted to the erased region with the appropriate syndrome.

The success probability of this decoder is the inverse of the number of logical operators that cannot be moved out of the region of erased qubits by multiplying with a stabilizer. This probability can be computed by Gaussian reduction as described in [23].

The capacity, i.e. the maximum rate of information, of the erasure channel [8] is computed from the probability of erasure. That is, the capacity of the channel with erasure probability p is $R_{\max} = 1-2p$. Inverting this relation, we observe that a code family with rate R can be used to protect information as long as the erasure probability is at most $\frac{1-R}{2}$. Figure 6 illustrates that the rate 1/10 codes obtained with our methodology achieve this limit. In other words, we have a procedure to construct random sparse stabilizer codes that are essentially capacity-achieving for the erasure channel.

6 Discussion and outlook

In this work, we reformulate the search for finite-rate sparse stabilizer codes as a constraint satisfaction problem, which we call stabilizing edge coloring, that is amenable to solution with state-of-the-art CSP solvers. We believe future work can exploit the connection between quantum error correction and constraint programming that we establish here in order to guide the search for new families of random stabilizer codes.

We note that even though here we limit our analysis to CSS codes for the sake of simplicity, the method we introduce is easily adaptable to more general stabilizer codes. For example, one can assign an extra variable to each edge to represent its Pauli value instead of labelling the stabilizer vertices directly. Furthermore, the method is flexible

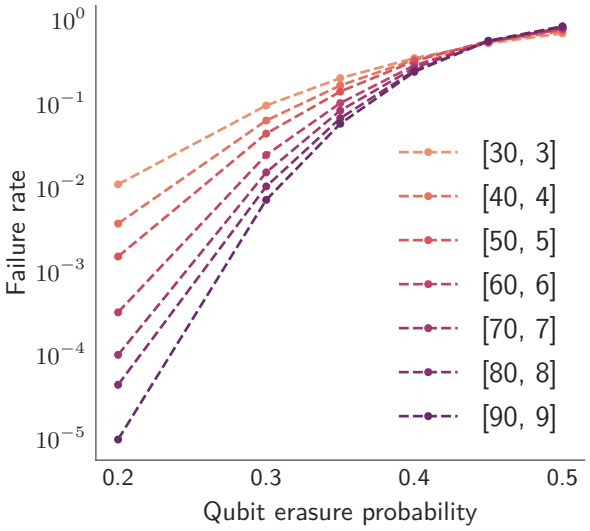

Figure 6: Failure rate for maximum-likelihood decoding of the erasure channel. For each system size, we show the lowest failure rates amongst all codes found.

in the sense that it is simple incorporate constraints that take into account multiple factors on the same footing, including, for example, hardware limitations like qubit layout and connectivity.

In this work we had to strike a balance between computational resources and a reasonable timeframe for completion of numerical calculations. Targeted searches for larger codes within the satisfiable phase using more resources are a straightforward direction for future work. Impressive progress in the performance of solvers in the last decades [1, 2] means that larger codes could soon be discoverable by our methodology with moderate resources. Customized CSP solvers that resolve the types of constraints involved in stabilizing edge coloring could also boost the search for codes with desirable properties.

It is also interesting to study different random graph constructions as input to CSP solvers instead of the uniformly sampled graphs we used. This could lead to sparser initial graphs with structure favorable to the discovery of good quantum codes.

Finally, we were able to use our approach to construct a family of finite-rate codes achieving optimal threshold for the erasure channel. This result, together with many recent stabilizer code constructions, motivate the search for more generally applicable quantum decoders for more complex noise channels such as the depolarizing channel and correlated Pauli noise. This would allow us to probe the performance of random code constructions more thoroughly.

Acknowledgments

This work was supported by the Ministère de l'Économie et de l'Innovation du Québec via its Research Chair in Quantum Computing and a Natural Sciences and Engineering Research Council of Canada Discovery grant. MT is supported by a Canada Graduate Scholarship from the Natural Sciences and Engineering Research Council of Canada. We acknowledge Calcul Québec and Compute Canada for computing resources. We thank the members of the QuICoPhy Theory Lab for valuable discussions.

References

[1] MiniZinc - Challenge, . URL https://www.minizinc.org/challenge.html.

[2] SAT Competitions, . URL http://satcompetition.org/.

[3] OR-Tools - Google Optimization Tools, March 2022. URL https://github.com/google/or-tools. original-date: 2015-02-21T01:25:35Z.

[4] Stabilizer code generation from a CSP solver, June 2022. URL https://github.com/quicophy/csp_code_gen. original-date: 2022-06-10T17:30:48Z.

[5] Dimitris Achlioptas and Cristopher Moore. Random k-SAT: Two Moments Suffice to Cross a Sharp Threshold. *SIAM Journal on Computing*, 36(3):740–762, January 2006. ISSN 0097-5397. DOI: 10.1137/S0097539703434231. URL https://epubs.siam.org/doi/abs/10.1137/S0097539703434231. Publisher: Society for Industrial and Applied Mathematics.

[6] Dimitris Achlioptas, Assaf Naor, and Yuval Peres. Rigorous location of phase transitions in hard optimization problems. *Nature*, 435

(7043):759–764, June 2005. ISSN 1476-4687. DOI: 10.1038/nature03602. URL https://www.nature.com/articles/nature03602. Number: 7043 Publisher: Nature Publishing Group.

[7] Alexei Ashikhmin, Simon Litsyn, and Michael A. Tsfasman. Asymptotically good quantum codes. *Physical Review A*, 63(3): 032311, February 2001. DOI: 10.1103/PhysRevA.63.032311. URL https://link.aps.org/doi/10.1103/PhysRevA.63.032311. Publisher: American Physical Society.

[8] Charles H. Bennett, David P. DiVincenzo, and John A. Smolin. Capacities of Quantum Erasure Channels. *Physical Review Letters*, 78(16):3217–3220, April 1997. DOI: 10.1103/PhysRevLett.78.3217. URL https://link.aps.org/doi/10.1103/PhysRevLett.78.3217. Publisher: American Physical Society.

[9] B. Bollobás and A. G. Thomason. Threshold functions. *Combinatorica*, 7(1):35–38, March 1987. ISSN 1439-6912. DOI: 10.1007/BF02579198. URL https://doi.org/10.1007/BF02579198.

[10] S. B. Bravyi and A. Yu Kitaev. Quantum codes on a lattice with boundary. *arXiv:quant-ph/9811052*, November 1998. URL http://arxiv.org/abs/quant-ph/9811052. arXiv: quant-ph/9811052.

[11] Sergey Bravyi and Matthew B. Hastings. Homological product codes. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, STOC '14, pages 273–282, New York, NY, USA, May 2014. Association for Computing Machinery. ISBN 978-1-4503-2710-7. DOI: 10.1145/2591796.2591870. URL https://doi.org/10.1145/2591796.2591870.

[12] Sergey Bravyi, David Poulin, and Barbara Terhal. Tradeoffs for Reliable Quantum Information Storage in 2D Systems. *Physical Review Letters*, 104(5):050503, February 2010. ISSN 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.104.050503. URL https://link.aps.org/doi/10.1103/PhysRevLett.104.050503.

[13] Winton Brown and Omar Fawzi. Short random circuits define good quantum error correcting codes. In *2013 IEEE International Symposium on Information Theory*, pages 346–350, July 2013. DOI: 10.1109/ISIT.2013.6620245. ISSN: 2157-8117.

[14] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098–1105, August 1996. DOI: 10.1103/PhysRevA.54.1098. URL https://link.aps.org/doi/10.1103/PhysRevA.54.1098. Publisher: American Physical Society.

[15] Nicolas Delfosse. Tradeoffs for reliable quantum information storage in surface codes and color codes. In *2013 IEEE International Symposium on Information Theory*, pages 917–921, July 2013. DOI: 10.1109/ISIT.2013.6620360. ISSN: 2157-8117.

[16] Nicolas Delfosse and Gilles Zémor. Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel. *Physical Review Research*, 2(3):033042, July 2020. DOI: 10.1103/PhysRevResearch.2.033042. URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.033042. Publisher: American Physical Society.

[17] Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[18] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, September 2012. DOI: 10.1103/PhysRevA.86.032324. URL https://link.aps.org/doi/10.1103/PhysRevA.86.032324. Publisher: American Physical Society.

[19] R. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8 (1):21–28, January 1962. ISSN 2168-2712. DOI: 10.1109/TIT.1962.1057683. Conference Name: IRE Transactions on Information Theory.

[20] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. PhD Thesis, California Institute of Technology, 1997.

[21] Daniel Gottesman. Fault-Tolerant Quan-

tum Computation with Constant Overhead. *arXiv:1310.2984 [quant-ph]*, October 2013. URL http://arxiv.org/abs/1310.2984.

[22] Antoine Grospellier, Lucien Grouès, Anirudh Krishna, and Anthony Leverrier. Combining hard and soft decoders for hypergraph product codes. *arXiv:2004.11199 [quant-ph]*, April 2020. URL http://arxiv.org/abs/2004.11199.

[23] Michael J. Gullans, Stefan Krastanov, David A. Huse, Liang Jiang, and Steven T. Flammia. Quantum Coding with Low-Depth Random Circuits. *Physical Review X*, 11(3): 031066, September 2021. DOI: 10.1103/PhysRevX.11.031066. URL https://link.aps.org/doi/10.1103/PhysRevX.11.031066. Publisher: American Physical Society.

[24] Matthew B. Hastings, Jeongwan Haah, and Ryan O'Donnell. Fiber Bundle Codes: Breaking the $N^{1/2} \operatorname{polylog}(N)$ Barrier for Quantum LDPC Codes. *arXiv:2009.03921 [quant-ph]*, October 2020. URL http://arxiv.org/abs/2009.03921. arXiv: 2009.03921.

[25] Aleksander Kubica, Michael E. Beverland, Fernando Brandão, John Preskill, and Krysta M. Svore. Three-Dimensional Color Code Thresholds via Statistical-Mechanical Mapping. *Physical Review Letters*, 120(18):180501, May 2018. DOI: 10.1103/PhysRevLett.120.180501. URL https://link.aps.org/doi/10.1103/PhysRevLett.120.180501. Publisher: American Physical Society.

[26] Andrew J. Landahl, Jonas T. Anderson, and Patrick R. Rice. Fault-tolerant quantum computing with color codes. *arXiv:1108.5738 [quant-ph]*, August 2011. URL http://arxiv.org/abs/1108.5738. arXiv: 1108.5738.

[27] Pavel Panteleev and Gleb Kalachev. Asymptotically Good Quantum and Locally Testable Classical LDPC Codes. *arXiv:2111.03654 [quant-ph]*, January 2022. URL http://arxiv.org/abs/2111.03654. arXiv: 2111.03654.

[28] Tom Richardson and Ruediger Urbanke. *Modern Coding Theory*. Cambridge University Press, Cambridge, 2008. ISBN 978-0-511-79133-8. DOI: 10.1017/CBO9780511791338. URL http://ebooks.cambridge.org/ref/id/CBO9780511791338.

[29] A. M. Steane. Simple quantum error-correcting codes. *Physical Review A*, 54(6):4741–4751, December 1996. DOI: 10.1103/PhysRevA.54.4741. URL https://link.aps.org/doi/10.1103/PhysRevA.54.4741. Publisher: American Physical Society.

[30] Ashley M. Stephens. Fault-tolerant thresholds for quantum error correction with the surface code. *Physical Review A*, 89(2):022321, February 2014. DOI: 10.1103/PhysRevA.89.022321. URL https://link.aps.org/doi/10.1103/PhysRevA.89.022321. Publisher: American Physical Society.

[31] Jean-Pierre Tillich and Gilles Zémor. Quantum LDPC Codes With Positive Rate and Minimum Distance Proportional to the Square Root of the Blocklength. *IEEE Transactions on Information Theory*, 60(2):1193–1202, February 2014. ISSN 1557-9654. DOI: 10.1109/TIT.2013.2292061. Conference Name: IEEE Transactions on Information Theory.

[32] Maxime A. Tremblay, Nicolas Delfosse, and Michael E. Beverland. Constant-overhead quantum error correction with thin planar connectivity. *arXiv:2109.14609 [quant-ph]*, September 2021. URL http://arxiv.org/abs/2109.14609. arXiv: 2109.14609.