MaxSAT Quantum Error Correction Decoders: a new formulation and benchmark study

Mohammadreza Noormandipour[1,*] and Tobias Haug[2]

[1]*Artificial Intelligence Research Lab, Nokia Bell Labs; Broers Building, 21 J.J. Thomson Avenue, Cambridge, CB3 0FA, UK*

[2]*Quantum Research Center, Technology Innovation Institute, Abu Dhabi, UAE*

(Dated: June 6, 2024)

Quantum error correction (QEC) is essential for operating quantum computers in the presence of noise. This paper explores the application of maximum satisfiability (MaxSAT) techniques to the decoding problem in QEC, focusing on Calderbank-Shor-Steane (CSS) codes. We introduce a novel method of mapping the QEC decoding problem into a MaxSAT instance, which allows for the efficient identification and correction of errors. By leveraging the flexibility in formulating MaxSAT instances and incorporating error probabilities as soft clause weights, we transform the MaxSAT decoder into a maximum likelihood decoder. Furthermore, we discuss the computational complexity associated with clause density in k-SAT problems and its implications for QEC decoders. The proposed approach is validated through numerical simulations and scaling analysis, demonstrating improved error thresholds and decoding performance. This work provides a promising direction for enhancing the reliability and efficiency of quantum error correction using advanced classical optimization techniques.

I. INTRODUCTION

Quantum computers promise to solve problems that cannot be addressed by classical computers. To run quantum computers even in the presence of noise, quantum error correction codes suppress errors by encoding logical quantum information in many redundant physical qubits [1]. Syndrome measurements probe the physical qubits for errors [2]. Then, a decoder is used to infer the most likely error, and the corresponding correction operation is applied. Here, it is crucial not to accidentally change the logical information stored in the code. Decoding is an NP-hard problem, yet to run a quantum computer successfully we need to find a good solution within the clock speed of the quantum computer. To this end, various types of decoders have been proposed with varying speed and accuracy tradeoff. Maximum likelihood decoders are optimal, but have exponential run time. There are tensor-network based decoders, neural network decoders, Ising decoders [3, 4], belief-propagation, perfect weight-matching [5], and union-find decoders.

Here, we concentrate on Calderbank-Shor-Steane (CSS) codes [6, 7], .

II. MAXSAT FORMULATION OF THE DECODING PROBLEM

In this section we will explain how to map the decoding problem into a maximum satisfiability (MaxSAT) instance. Firstly, we will briefly review a standard and widely used format of writing a MaxSAT problem, conjunctive normal form (CNF). A k-SAT instance, is a specific case of a Boolean satisfiability problem in which you are given a Boolean expression written in CNF form where each clause is constrained to k literals. The table below summarises the important concepts/convention that we will use later on.

Boolean Variables	In k-SAT, you deal with variables that can take the values True or False.
Literals	A literal is a variable or its negation. For example, if x_i is a variable, then x_i and $\neg x_i$ (not x_i) are literals.
Clauses	A clause is a disjunction (OR) of literals. In k-SAT, each clause has exactly k literals. For instance, $(x_1 \vee \neg x_2 \vee x_3)$ is a 3-SAT clause.
Conjunctive Normal Form (CNF)	A Boolean formula is said to be in CNF if it is a conjunction (AND) of clauses. For example, $(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$ is a CNF formula.
Satisfiability	The k-SAT problem asks whether there exists an assignment of True or False to the variables such that the entire CNF formula evaluates to True.

For a MaxSAT instance, one aims to find an assignment of the variables such that it satisfies maximum number of clauses.

In quantum error correction (QEC), the parity check matrix of the QEC code, specifies which qubits can affect the state of a detector. If we assign a binary variable to the connection between the qubits and the detectors, then the core of a decoding problem is to match the parity of the product of these binary variables to the measured syndrome. This task can naturally be translated to a MaxSAT instance. There is some flexibility in formulating the problem as a MaxSAT instance, e.g. in choice of the clause lengths. However, some mappings can be much harder to solve for a typical SAT solver.

In the study of computational complexity, the clause density of a k-SAT problem is defined as the ratio of the number of clauses m to the number of variables n, denoted by $\alpha = \frac{m}{n}$. The hardness phase transition for k-SAT problems is a critical concept, highlighting a threshold clause density at which the problem transitions from being predominantly solvable to predominantly unsolvable.

For different values of k, the critical clause density where this phase transition occurs varies. For $k = 2$, the phase transition occurs at $\alpha \approx 1.0$. Problems with clause densities around this value exhibit a sharp transition from likely satisfiable to

* mohammadreza.noormandipour@nokia-bell-labs.com

likely unsatisfiable. For $k = 3$, the phase transition occurs at $\alpha \approx 4.2$, a widely studied and critical threshold in the theory of NP-completeness. For $k = 4$, the phase transition is observed at $\alpha \approx 9.9$. Beyond this density, 4-SAT instances are generally much harder to solve. These transitions are important in understanding the computational complexity and predictability of SAT-solving algorithms under different conditions. They provide a theoretical framework for expecting a sudden shift in the solvability of the problems as the clause density crosses these thresholds. For a detailed discussion see [8]. Therefore, the critical clause density can have important implications on whether it is feasible to use SAT solvers as QEC decoders and which QEC codes would be ideal for these solvers.

Apart from the hardness phase transition as a factor to consider in choosing the clause length, it is also important to note that lower clause lengths are more desirable. This is due to the fact that designing a hardware SAT solver for lower k values is easier. In other words, you can think of a clause as a k-body interaction term which has to be realised, in order to gain the speedup in using fast hardware.

A. CNF Construction

Now we will explain the strategy to construct a CNF MaxSAT problem for each round of syndrome measurement.

B. Error Probabilities as Soft Clause Weights

Most MaxSAT algorithms allow for introducing a weight associated with each soft clause, which would be the cost of violating that particular clause. On the other hand, hard clauses are considered as infinite-weight clauses, which means that even if one of the hard clauses is violated, the solver has failed to find a satisfying assignment. As we mentioned above, in order to find the minimal solution (solution with least number of errors), we can add the negation of each of the error variables as a soft clause. Now, in order to further advance the mapping, we can add a measure of the probabilities associated with each error also as weights of these soft clauses. This in principle turns the MaxSAT decoder into a maximum likelihood decoder. More precisely, we are given a set of errors E, each with probability p_i and a set of detectors D. Given occurred errors $\boldsymbol{e} \in E$, the matrix H describes the measured syndrome $\boldsymbol{s} \in S$ via $\boldsymbol{s} = H\boldsymbol{e}$, where S is the list of possible syndromes. Decoders such as minimum-weight perfect matching (MWPM) demand that each error affects at most two decoders, i.e. each column of H has at most two non-zero entries. One important advantage of the maximum likelihood decoders is that they do not suffer from this restriction. The probability of a particular error occurring is given by

$$P(\boldsymbol{e}) = \prod_i (1-p_i)^{1-e_i} p_i^{e_i} = \prod_i (1-p_i) \prod_i \left(\frac{p_i}{1-p_i}\right)^{e_i} . \quad (1)$$

To find the most probable error $\boldsymbol{c}$ given the measured syndrome $\boldsymbol{s}$, we maximize

$$\ln(P(\boldsymbol{c})) = C - \sum_i w_i c_i \quad (2)$$

$$\text{s.t. } \boldsymbol{s} = H\boldsymbol{c} , \quad (3)$$

where we have the irrelevant constant $C = \sum_i \ln(1-p_i)$ and weights $w_i = \ln((1-p_i)/p_i)$ which represent the Bayesian prior of our knowledge on the error probabilities. Maximising the loglikelihood in Eq. 2 would be equivalent to assigning the w_i weights to the soft clauses as violation cost to be minimised.

III. CORRELATED DECODING

Commonly, in CSS codes the decoding of X and Z errors is done separately. Here, an underlying assumption is that X and Z errors are independent of each other. However, for depolarizing error this is not the case as Y errors induce both an X and Z error at the same time. The correct maximum likelihood decoder must include these correlations. With the MaxSAT decoders, these correlations can be easily included into the decoding problem as we will explain in the following.

For depolarising error, we can assign separate literals to X and Z errors and demand that both are decoded at the same time. This will increase the size of the the decoding problem, but it is the correct strategy for depolarising error as mentioned above.

Y errors were accounted for in the formulation of Ref. [9]. However, their approach is restricted to the surface code, while ours is completely generally for any CSS code.

IV. RESULTS AND DISCUSSION

In this section we present the improved scaling and error thresholds for our decoder in comparison to the BP-OSD decoder which is the state of the art for the studied codes. Solvers 2 to 5 represent various algorithms for solving MaxSAT instances. They all are integrated in our high performance decoding package called OptiSync.

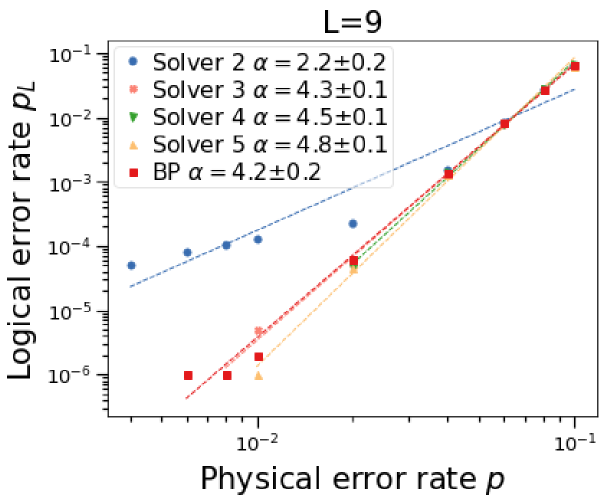
[1] P. W. Shor, in *Proceedings of 37th conference on foundations of computer science* (IEEE, 1996) pp. 56–65.

[2] S. M. Girvin, SciPost Physics Lecture Notes , 070 (2023).

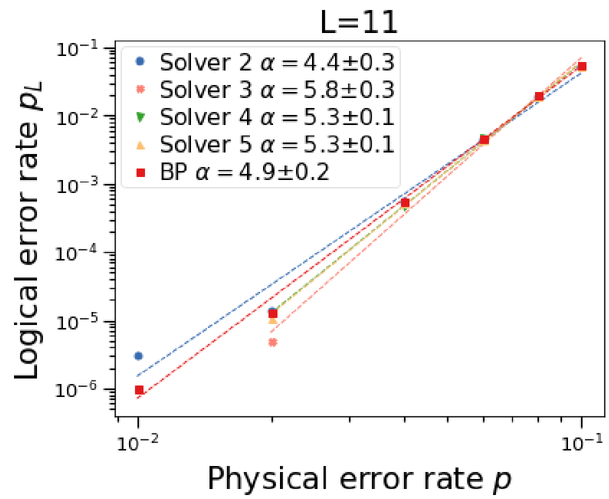
[3] J. Fujisaki, H. Oshima, S. Sato, and K. Fujii, Physical Review Research **4**, 043086 (2022).

TABLE I: Comparison of Error Thresholds (%) for MaxSAT and BP-OSD Decoders

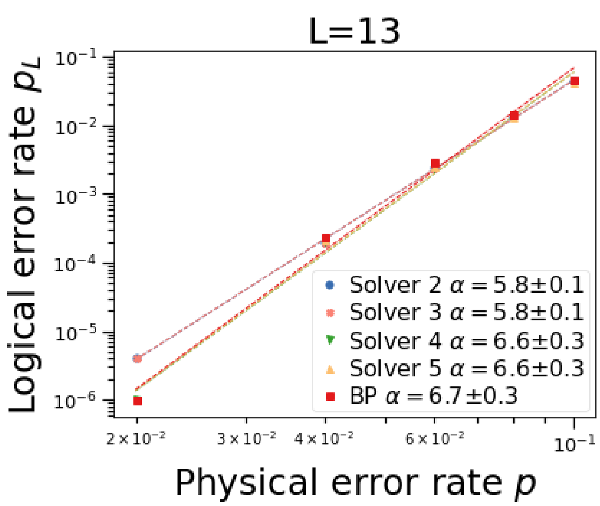
Code	Solver 2	Solver 3	Solver 4	Solver 5	BP-OSD
Toric Code (2D)	15.65 ± 0.34	5.57 ± 0.35	15.64 ± 0.33	15.67 ± 0.24	15.25 ± 0.51
Color Code (on hexagonal lattice)	15.26 ± 0.23	15.41 ± 0.20	15.36 ± 0.16	15.22 ± 0.16	13.37 ± 0.13



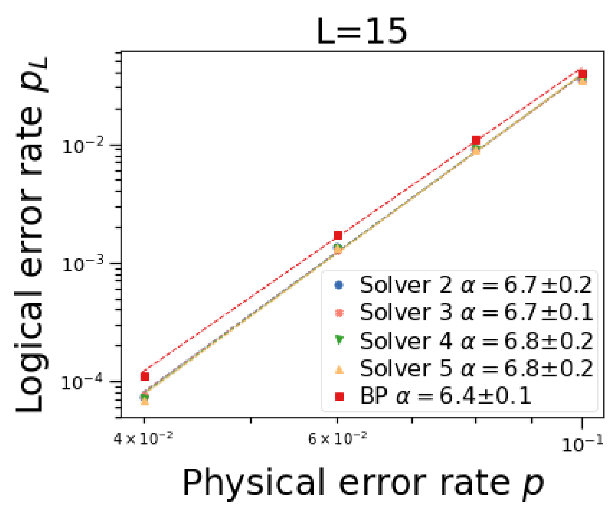
(a) Code distance 9



(b) Code distance 11



(c) Code distance 13



(d) Code distance 15

FIG. 1: The scaling of logical error rate with physical error rate for various solvers within OptiSync given with the relation $p_L \propto p_{phys}^{\alpha}$. These results are for the Toric 2D code.

[4] Y. Takada, Y. Takeuchi, and K. Fujii, Phys. Rev. Res. **6**, 013092 (2024).

[5] O. Higgott and C. Gidney, arXiv:2303.15933 (2023).

[6] A. Steane, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **452**, 2551 (1996).

[7] A. R. Calderbank and P. W. Shor, Physical Review A **54**, 1098 (1996).

[8] G. Bresler and B. Huang, "The algorithmic phase transition of random k-sat for low degree polynomials," (2021), arXiv:2106.02129 [cs.CC].

[9] Y. Takeuchi, Y. Takada, T. Sakashita, J. Fujisaki, H. Oshima, S. Sato, and K. Fujii, arXiv:2311.07973 (2023).