On The Complexity of Quantum Circuit Manipulation

Vincent Liew

1 Introduction

The stabilizer class of circuits, introduced by Daniel Gottesman, consists of quantum circuits in which every gate is a controlled-NOT (CNOT), Hadamard or phase gate [5]. These circuits have several interesting properties. For example, they are robust enough to allow for entangled states, yet they are known to be simulable in polynomial time by a classical computer. These circuits also naturally arise in the study of quantum error-correcting codes.

Iwama et. al. introduced a complete set of local transformations for CNOT circuits [7]. That is, they gave a list of transformation rules, each of which involve a constant number of gates, which suffice to take any CNOT circuit to any equivalent such circuit. Such sets of rules typically have applications in developing heuristic circuit minimization techniques. In previous work, we built upon these rules to give a more general set of local transformations for the more general class of stabilizer circuits. With these rules in hand, several questions naturally arise, which we address in this paper.

Problem 1. *Let* STAB-TRANS $= \{\langle S, k\rangle\}$ *be the language of stabilizer circuits* S *which may be transformed into the identity circuit in less than* k *steps. Is* STAB-TRANS *in* **P***? Is it* **NP***-complete?*

Problem 2. *Let* TOFFLIDENTITY *be the language of circuits composed of Toffoli, CNOT and NOT gates which are equivalent to an empty circuit. This is known to be* **co-NP***-complete. Can we exhibit an explicit family of circuits which requires exponentially many local transformations to reduce to the identity?*

Problem 3. *Let* STAB-MIN $= \{\langle S, k\rangle\}$ *be the language of stabilizer circuits* S *which have an equivalent circuit consisting of less than* k *gates. Is* STAB-TRANS *in* **P***? Is it* **NP***-complete?*

2 Approaches to STAB-TRANS

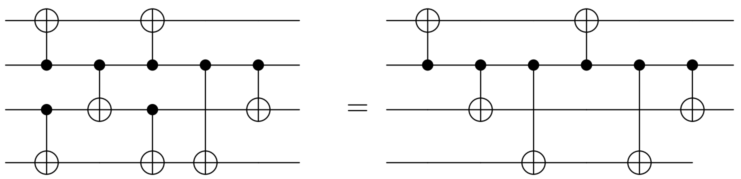
Since we have a set of local transformation rules, it is natural to ask how difficult it is to implement them optimally in order to transform two equivalent circuits A and B to each other. Because of the reversibility of each gate, we may equivalently ask how difficult it is to find the optimal transformations to the identity; the optimal series of transformations to get from AB^{-1} to the identity translates to the optimal series of transformations on A to get to B.

We previously, in proving the validity of a set of transformation rules for stabilizer circuits, gave an algorithm for transforming stabilizer circuits into the identity. This gave a polynomial upper bound for the number of transformations required. We also have a trivial lower bound of $\Omega(s)$

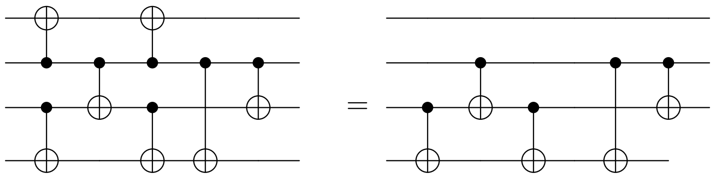
steps, where s is the number of gates in the circuit, as each gate requires a cancellation step. In the intermediate region, however, this problem becomes more difficult.

It is easy to see that STAB-TRANS is contained in **NP**. The list of k transformation steps suffices as a witness to the solution. So we would naturally like to know whether the problem is in **P** or **NP**-complete. If we attempt to come up with a polynomial time algorithm, we find that it is unclear how a dynamic programming or divide-and-conquer approach might work because it optimal transformations within in a segment of our circuit are not necessarily optimal in the rest of it. So much of our effort on this problem was on attempting to find a greedy polynomial time algorithm. Because we eventually want to eliminate all gates in the circuit, it seems that a good measure for how close we are to the identity is how many gates are left in the circuit.

We will give a counterexample to sufficiently shortsighted greedy algorithms for STAB-TRANS. We will, in particular, show that solving the sub-problem of finding the shortest series of transformation steps that reduces the circuit size by at least one gate does not yield the shortest overall transformation to the identity. We note, however, that we do not know whether this sub-problem has an efficient algorithm. Consider the left circuit below:



In one step, by applying an identity to the bottom three qubits on the left, we may reduce the circuit size by one to get the circuit on the right. This is the shortest path to a reduction in circuit size as it is only one step. The shortest path to the identity from the circuit on the right is the obvious one from commuting and cancellation of the controlled-NOT gates. However, this was not optimal. The shortest path instead cancels the two controlled-NOT gates targeting the first qubit as shown below (this takes two steps: one to move and one to cancel), so that we do not have to commute around them. Then when we cancel the rest of the gates, we do not have to take the steps to commute through the top gates.



Although we have ruled out the most shortsighted greedy algorithm, it might be that an algorithm with more foresight, which finds the shortest path to reducing the circuit size by $O(1)$ gates instead of just 1 gate, will yield an efficient algorithm.

3 Local Transformations and TOFFLIDENTITY

It turns out, due to Alagic, Jeffery and Jordan, that TOFFLIDENTITY is **CoNP**-complete. This implies, under the assumption that **CoNP**≠**NP**, there must be some family of circuits made of CNOT, Toffoli and NOT gates which requires an exponential number of local transformations to reduce to the identity. This is because if we could carry out the transformation in a polynomial number of steps, these steps themselves constitute a certificate so that TOFFLIDENTITY $\in$ **NP**.

We will give a family of circuits which requires exponentially many steps to convert into any other form via the procedure given by Iwama et al. Although we conjecture that this a related family may require exponentially many steps in general to reduce to the identity, we do not know of a method by which we might prove this.

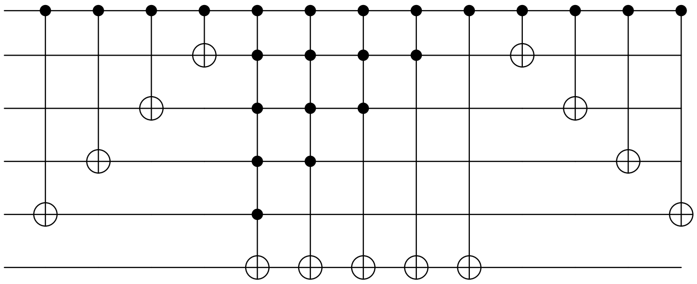
The method by which Iwama et al. proved that their local transformation rules are universal is by showing that they sufficed to transform any generalized-CNOT (where we may use arbitrarily many control qubits) circuit into a canonical form. Thus, to transform one circuit into an equivalent circuit, we can transform the first to canonical form, and then transform into the second circuit. We will show that there is a family of circuits which have a polynomial size, yet become exponential when converted into canonical form. This implies that Iwama's procedure does indeed require exponential time for certain circuits as we expected.

Definition 4. *(From [7]) A generalized-CNOT circuit is said to be in canonical form if (1) all its gates target the last qubit, (2) it does not include two or more of the same gate, (3) the gates are ordered lexicographically in terms of the indices in their control bits.*

We note that this definition restricts our attention to circuits which are the identity except on the last qubit.

Definition 5. *Let C_n, where $n > 3$, be the n-qubit quantum circuit composed of the following three sections: in the first section, we place a series of Toffoli gates so that the first has target qubit $n-3$ and control qubits $n-2$ and $n-1$, and each subsequent Toffoli gate has target has the target and control qubits of the previous gate minus one until the last gate targets the first qubit. Then we place a series of Toffoli gates which all target qubit n. The first has controls for qubits i where $i < n$. The next had controls for qubits less than $n-1$, and so on. The last segment is the same as the first, but reversed in order.*

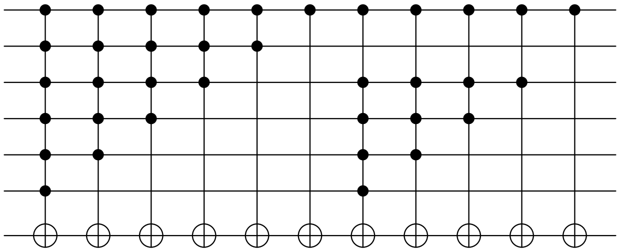
For example, C_6 is shown below.



Theorem 6. *$\{C_n\}$ is a family of circuits which have a canonical form exponential in n*

We note that the size of each circuit is polynomial in n even if we restrict ourselves to just using Toffoli gates because we only need $O(n)$ Toffoli gates to simulate a generalized-CNOT (henceforth called a CNOT) gate with n controls [4]. Further it is straightforward to see that the only qubit which is changed by this circuit is the last one; the effect of the gates in the first section is cancelled by the gates in the last section.

Proof. The idea is that in order to convert this circuit to its canonical form, we must eliminate the gates which target qubits other than n. This means that we must move the gates from the first section of the circuit through the second, and into the third in order to use the cancellation rule. We will start with the gates closest to section two. Notice that when we bring the first gate through, it will create new gates which target the last qubit and which have controls at the union of the controls of the gates we are commuting. So upon commuting the innermost gate through to the other side of the circuit, section two for, say, C_7 will look like:



if we rearrange the CNOTs into the order in which they are generated (the CNOTs in section two all commute so we may arrange them as we like). We see that moving the CNOT from section 1 through essentially "copies" section 2, but without the controls corresponding to the target of the CNOT we are moving. Notice that we have generated a copy of a CNOT already present; we will account for this in our analysis. Also, we do not end up copying the gates which commuted with the CNOT we moved through.

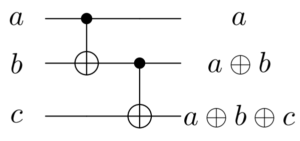
So when we move the next CNOT through, we will copy the above circuit, but without the controls corresponding to qubit 3. We can see that each time we move a CNOT through, we "almost" double the size of section 2. We now check just how large this section becomes. Let us assume, as a simplification, that we do not cancel alike CNOTs yet. Notice that when we copy section 2 as outlined above, we also copy all of the gates which will not commute with the next CNOT we move through. The initial configuration of section 2 had $n-1$ gates which did not commute with the first CNOT, $n-2$ noncommuting with the second CNOT and so on. On step i, we will have 2^{i-1} "copies" of the initial $n-i+1$ gates which do not commute with the CNOT targeting qubit $n-i$. On this step, then, we will generate $(n-i+1)\times 2^{i-1}$ gates. It is clear from this that the total number of gates generated will be exponential in n. To account for generating cancelling CNOTs, we note that only those CNOTs with a single control at the top and a target at the bottom ever induce a cancellation (we may see this by inspection). There is at most one single-control CNOT created per copy. So if we throw all of these away, we remove at most 2^{i-1} gates which does not affect the exponentiality of the number of gates. □

4 Approaches to STAB-MIN

To start, we note that there are two parameters by which we might measure complexity of an input: the number of qubits used by the circuit, and the number of gates. It turns out that any stabilizer circuit on n qubits has an equivalent stabilizer circuit consisting of $O(n^2/\log n)$ gates [2]. Therefore the appropriate measure of input complexity is the number of qubits a circuit uses.

It is clear the STAB-MIN is in **NP**. Given a solution circuit, we may verify in polynomial time that it is equivalent to the circuit we began with. Much of the nontrivial structure in stabilizer circuits is due to the CNOT gates; if we restricted ourselves to only the single-qubit Hadamard and phase gates, we get a very simple circuit class. But if we restrict ourselves to only CNOT gates, we obtain a possibly simpler optimization problem (which we will call CNOT-MIN).

Given a CNOT circuit, we may represent its action in the form of a matrix. This matrix tracks the outputs of the circuit; the nth row is a vector containing the output of the nth qubit. This output will consist of a subset of the input variables added modulo 2 so the vector will have a 1 in its entries corresponding to the variables present in this sum. A CNOT operation, in this matrix, corresponds to row addition. So in this representation, the following circuit is represented by the below matrix.



$$\downarrow$$

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

It is apparent from this formulation that CNOT-MIN is equivalent to the following problem: given an invertible matrix over GF(2), find the fewest number of row operations which performs this reduction. This problem has been previously posed in [3] as an open problem.

The problem of row reduction is closely related to the following open problem posed by Gowers in [6] and attributed by him to Widgerson:

Problem 7. *Let $D(M_n)$ be the minimum number of row operations to reduce M_n to the identity. Find an explicit construction for a sequence of $n \times n$ invertible matrices over GF(2), $\{M_n\}$, such that $n = o(D(M_n))$.*

Here, Gowers interprets explicit construction to mean that there is a polynomial time algorithm which might generate each matrix. This problem is thought to be fairly challenging and has remained open for a significant time. If we could solve CNOT-MIN in polynomial time, then we would be significantly closer to a solution to this problem. A counting argument (there are less than $2n^2$ row operations and roughly 2^{n^2} invertible matrices) reveals that almost every invertible matrix over GF(2) will require a superlinear number of row operations to reduce to the identity. So for each n, it would suffice to sample uniformly and then verify, using the algorithm for CNOT-MIN, that our matrix satisfied a lower bound. It would remain to derandomize this algorithm. Thus, it seems unlikely the CNOT-MIN would be in **P**, based on the difficulty of the above problem.

Although we have not shown that CNOT-MIN is **NP**-complete, we can give an **NP**-complete problem which is very closely related to it, due to Forbes [1].

Definition 8. *Let RRD be the problem of taking as input $m \times n$ matrices M, N over $\mathbb{F}_2$ and an integer k and outputting whether M row-reduces to N in less than k steps.*

Essentially, if we allow for arbitrary matrices over $\mathbb{F}_2$ then we may obtain an **NP**-completeness result. Unfortunately the reduction inherently makes use of these extra matrices, so we have not been able to adapt it to our problem. However, that this problem is **NP**-complete does hint that

CNOT-MIN is also **NP**-complete. And it may be that we can reduce the more general STAB-MIN problem to RRD to prove **NP**-completeness.

We end with a discussion of a heuristic method for reducing the size of a stabilizer circuit which makes use of the local transformation rules for these circuits. Following the work of Maslov, Young, Dueck and Miller [8], we use a greedy algorithm to reduce the size of the circuit. Each transformation rule corresponds to several different rules which might make the circuit smaller. If we had a rule of the form $abcdef = I$, for example, we could apply $abcde = f^{-1}$ in addition to our original rule. Given a list of rules and a circuit, we attempt to apply the smallest rules which will decrease the size of the circuit first. The larger rules will be attempted after the smaller ones.

This approach was proposed as a general way of using local transformation rules for circuit simplification by [8], and was found to successfully give better implementations of previously synthesized circuits. We are currently writing a program which implements this algorithm in order to obtain experimental data on its efficacy. It would be interesting to see if known stabilizer circuits implementing error correction could be reduced in size.

References

[1] http://mathoverflow.net/questions/69873/what-is-the-complexity-of-this-problem.

[2] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70(052328), 2004. quant-ph/0406196.

[3] Daniel Andrn, Lars Hellstrm, and Klas Markstrm. On the complexity of matrix reduction over finite fields. *Advances in Applied Mathematics*, 39(4):428 – 452, 2007.

[4] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52(3457), 1995. quant-ph/9503016.

[5] Daniel Gottesman. The heisenberg representation of quantum computers. *Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics*, pages 32–43, 1998.

[6] W.T. Gowers. Rough structure and classification. In N. Alon, J. Bourgain, A. Connes, M. Gromov, and V. Milman, editors, *Visions in Mathematics*, Modern Birkhuser Classics, pages 79–117. Birkhuser Basel, 2010.

[7] Kazuo Iwama, Shigeru Uamashita, and Yahiko Kambayashi. Transformation rules for cnot-based quantum circuits and their applications. *New Generation Computing*, 21(4):297–317, 2003.

[8] D. Maslov, C. Young, D. M. Miller, and G. W. Dueck. Quantum circuit simplification using templates. In *Proceedings of the conference on Design, Automation and Test in Europe - Volume 2*, DATE '05, pages 1208–1213, Washington, DC, USA, 2005. IEEE Computer Society.