Optimising quantum circuits is generally hard

John van de Wetering[1] and Matthew Amy[2]

[1]University of Amsterdam
[2]Simon Fraser University

July 28th 2024

In order for quantum computations to be done as efficiently as possible it is important to optimise the number of gates used in the underlying quantum circuits as much as possible. In this paper we find that many gate optimisation problems for approximately universal quantum circuits are NP-hard. In particular, we show that optimising the T-count or T-depth in Clifford+T circuits, which are important metrics for the computational cost of executing fault-tolerant quantum computations, is NP-hard by reducing to Boolean satisfiability. With a similar argument we show that optimising the number of CNOT gates or Hadamard gates in a Clifford+T circuit is also NP-hard. Again varying the same argument we also establish the hardness of optimising the number of Toffoli gates in a reversible classical circuit. We find an upper bound to T-count and Toffoli-count of $\mathrm{NP}^{\mathrm{NQP}}$. Finally, we also show that for any non-Clifford gate G, doing approximate optimisation of the G-count over the Clifford+G gate set, where we only have to match the target unitary within some small distance in the operator norm, is NP-hard.

1 Introduction

Applications of quantum computers can roughly be divided into two domains: those which work on relatively small and noisy devices, and those which work only on large-scale fault-tolerant machines. In the first category exist for instance a variety of variational algorithms that combine classical optimization with small, parameterized quantum circuits [9, 30]. For these circuits, noise accumulates over time and after every operation. It is then crucial to optimise the *depth* of the circuit, corresponding to the runtime of the computation, as well as the number of multi-qubit operations (like CNOT gates), as those gates tend to be much more noisy than single-qubit operations.

In the second category we find most of the proven speedups of quantum computers, like applications of Grover's [19] or Shor's algorithm [27] or solving quantum chemistry problems [5]. For such large-scale quantum computations we require a way to combat the accumulation of errors. This can be done by using *quantum error correction* [28]. This allows us to distribute the logical quantum information over multiple physical qubits, which we can combine with clever measurements in order to detect and correct potential errors. If we want to do operations on the encoded logical information, we however need to implement this in such a way on the physical hardware that it does not spread errors in an

John van de Wetering: john@vdwetering.name, http://vdwetering.name

Matthew Amy: meamy@sfu.ca, https://www.cs.sfu.ca/ meamy/

uncorrectable way. That is, we need to implement the logical operations in a *fault-tolerant* manner.

The most straightforward way to fault-tolerantly realise an operation, is to do it *transversally*, meaning that for instance we implement a logical S gate by simply applying an S gate on each individual physical qubit. Unfortunately, no quantum error correcting code can have an approximately universal gate set of transversally implementable gates [12]. This means other techniques must be used to implement the gates that don't have a transversal implementation. Two common techniques to do so, *code-switching* and *magic state distillation* both are much more expensive than doing a transversal gate [6]. An important metric to optimise in the fault-tolerant regime of quantum computing is hence the number of gates requiring such costly procedures.

To make this more concrete, let's consider the most well-studied model for fault-tolerant quantum computing: the surface code [13]. In this model a logical qubit is encoded into a 2D-grid of physical qubits known as a *patch*. Most Clifford operations can be implemented relatively cheaply in this model: Pauli gates are 'free' as they consist of updating the frame of measurement outcomes; Hadamards are implemented by a series of deformations of the patch resulting in a 'rotation' of the patch; and CNOTs can be done using *lattice surgery* [11, 21] (or in earlier proposals, by *braiding* [13]), meaning we merge and split patches by doing some smart additional measurements of the qubits. To boost this to a universal model of computation, we require a non-Clifford gate. The most common choice is the T *gate* $\text{diag}(1, e^{i\frac{\pi}{4}})$. This is implemented by injecting a $|T\rangle := T\,|+\rangle$ *magic state* into one of the logical qubits. This requires *distilling* a collection of noisy $|T\rangle$ states into increasingly less-noisy magic states, until they reach the desired precision. It is this distillation process which is very expensive. In [13] it was for instance estimated that 95% of all operations in the computer for an implementation of Shor's algorithm would be dedicated to magic state distillation factories [13]. In more recent works, distillation protocols have been improved significantly, though the distillation protocols still encompass a large part of all resources [17, 18, 23, 24].

Simplifying the story somewhat, we can hence say that optimising a fault-tolerant quantum circuit means optimising the number of T gates, known as the T-count, required to implement the circuit. There have hence been a multitude of results, both heuristics and optimal algorithms, for optimising the T-count of a given circuit or unitary. These can be divided into three classes. First, there are the methods that treat all non-Clifford phase rotations equivalently, and just fuse together phases that can be brought together using the sum-over-paths method or by representing the gates by a series of Pauli exponentials [3, 22, 31, 33]. These techniques are efficient, but generally don't find the optimal T-count of a circuit. Second, there are the methods that synthesise directly from the matrix, and do not optimise a circuit in place [15, 16]. These produce optimal T-counts by design, but are infeasible to run on circuits beyond just a couple of qubits in size. Third, there are the methods that use the equivalence between optimising the T-count of diagonal CNOT+T circuits and well-studied problems like symmetric 3-tensor factorisation and Reed-Muller decoding [4, 10, 20, 26].

While more powerful than the phase-fusing techniques and more efficient than the direct synthesis techniques, the methods in the third group still grow prohibitively costly for large circuits (beyond roughly 50 qubits). This is not too surprising as Reed-Muller decoding and symmetric 3-tensor factorisation are believed to be hard problems. However, to our knowledge no concrete hardness result is known for optimising the T-count of a general Clifford+T circuit. In this paper we demonstrate with a simple argument that this problem is at least NP-hard. This argument turns out to be easily adaptable to prove the hardness of optimising several other types of gates: Toffolis, CNOTs, Hadamards, and in fact any other non-Clifford gate.

1.1 Statement of results

We will define the problem T-COUNT as follows: given an integer k and a Clifford+T circuit implementing a unitary U, determine whether there exists a Clifford+T circuit implementing U using at most k T gates. Note that the optimisation version of the problem reduces to the T-COUNT problem via a binary search running logarithmically in the length of the circuit. We can then state our main result.

Theorem 1. T-COUNT is NP-hard under polynomial-time Turing reductions.

Most T gates in many applications are used inside of Toffoli gates $\mathrm{Tof}\,|x,y,z\rangle \mapsto |x,y,z\oplus(xy)\rangle$, as part of the synthesis of classical functions being applied to quantum states (such as the synthesis of the modular exponentiation needed in Shor's algorithm). An interesting related question is hence to optimise the number of Toffoli gates in such a classical circuit consisting of Toffoli, CNOT and NOT gates. Defining the problem of TOF-COUNT similarly to T-COUNT, we show how a similar argument for the hardness of T-COUNT can also be used to prove the hardness of TOF-COUNT.

Theorem 2. TOF-COUNT is NP-hard under polynomial-time Turing reductions.

We further show that optimisation of *any* non-Clifford gate is NP-hard. Let G be some non-Clifford gate. For some $\varepsilon > 0$ we define the problem of G-COUNT$_\varepsilon$ as follows: Given an integer k and a Clifford+G circuit implementing some unitary U, determine whether there exists a Clifford+G circuit containing at most k G gates which implements a unitary U' that is within distance ε of U in the operator norm.

Theorem 3. Let G be any non-Clifford gate and $\varepsilon < \sin(\frac{\pi}{16}) \approx 0.195$. Then G-COUNT$_\varepsilon$ is NP-hard.

We also find an upper bound for the T-COUNT and TOF-COUNT problems. To state this we require the complexity class NQP, standing for *non-deterministic quantum polynomial* time. This class has as a complete problem determining whether two poly-size quantum circuits are *exactly* equal (and hence should be contrasted with the more well-known complexity class QMA which has as a complete problem determining whether two circuits are *approximately* equal).

Theorem 4. The T-COUNT and TOF-COUNT problems are contained in $\mathrm{NP}^{\mathrm{NQP}}$.

Although non-Clifford gates are the majority of the cost of fault-tolerant implementations, other gates aren't free, and hence we would also like to optimise those. In particular, CNOT gates introduce connectivity constraints that might require expensive routing across distant qubits. Let CNOT-COUNT be defined analogously to T-COUNT, but with respect to CNOT gates (i.e. count the number of CNOT gates in a Clifford+T circuit).

Theorem 5. CNOT-COUNT is NP-hard under polynomial-time Turing reductions.

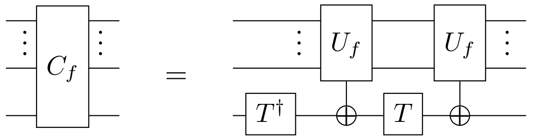
The argument in fact works for any entangling gate in the Clifford+T gate set, and also works if we let CNOT gates between different qubit pairs carry different weights (as long as all those weights are non-zero).

We could express the Clifford+T gate set succinctly as consisting of CNOT, T and Hadamard gates. We have now seen that optimising the first two types of gates is NP-hard. Optimising Hadamard gates is in fact also hard.

Theorem 6. Hadamard-COUNT is NP-hard under polynomial-time Turing reductions.

2 Proof of main result

We establish NP-hardness by reduction from Boolean satisfiability. Let $f : \{0,1\}^n \to \{0,1\}$ be some Boolean function, given as a Boolean expression. Using standard techniques we can build the classical oracle U_f implementing $U_f |\vec{x}, y\rangle = |\vec{x}, y \oplus f(\vec{x})\rangle$. Note that U_f is an $(n+1)$-qubit quantum circuit which can be constructed as a poly(n) size Clifford+T circuit (which requires potentially one borrowed ancilla). Consider then the following quantum circuit C_f:



(1)

It is straightforward to verify that C_f implements the diagonal operation

$$C_f |\vec{x}, y\rangle \;=\; e^{i\frac{\pi}{4}(1-2y)f(\vec{x})} |\vec{x}, y\rangle\,.$$

Now, if f is not satisfiable, then $f(\vec{x}) = 0$ for all $\vec{x}$, and hence we see that $C_f = \mathrm{id}$. Additionally, if f is satisfiable for all $\vec{x}$, then we have

$$C_f |\vec{x}, y\rangle = e^{i\frac{\pi}{4}(1-2y)} |\vec{x}, y\rangle \;=\; e^{i\frac{\pi}{4}} e^{-i\frac{\pi}{2}y} |\vec{x}, y\rangle \;=\; e^{i\frac{\pi}{4}} (I_n \otimes S^\dagger) |\vec{x}, y\rangle\,.$$

So, up to global phase, C_f is just an $S^\dagger$ gate in this case, and hence Clifford. In either case C_f is Clifford, so that the minimal T-count of C_f is zero.

Now suppose f is satisfiable, but that not every input is a solution. Then there exist $\vec{z}_1$ and $\vec{z}_2$ such that $f(\vec{z}_1) = 1$ and $f(\vec{z}_2) = 0$. Then it is easy to see that

$$C_f |\vec{z}_1, 0\rangle \;=\; e^{i\frac{\pi}{4}} |\vec{z}_1, 0\rangle \qquad \text{and} \qquad C_f |\vec{z}_2, 0\rangle = |\vec{z}_2, 0\rangle\,.$$

We can now observe that C_f is non-Clifford by considering the action of C_f on the n-qubit Pauli $X^{\vec{z}_1 \oplus \vec{z}_2} := X^{(\vec{z}_1 \oplus \vec{z}_2)_1} \otimes \cdots \otimes X^{(\vec{z}_1 \oplus \vec{z}_2)_n}$. In particular,

$$C_f^\dagger X^{\vec{z}_1 \oplus \vec{z}_2} C_f |\vec{z}_1, 0\rangle = e^{i\frac{\pi}{4}} C_f^\dagger X^{\vec{z}_1 \oplus \vec{z}_2} |\vec{z}_1, 0\rangle = e^{i\frac{\pi}{4}} C_f^\dagger |\vec{z}_2, 0\rangle = e^{i\frac{\pi}{4}} |\vec{z}_2, 0\rangle\,,$$

and hence $C_f^\dagger X^{\vec{z}_1 \oplus \vec{z}_2} C_f$ is not a member of the n-qubit Pauli group. By definition C_f is non-Clifford and so its minimal T-count over Clifford+T is necessarily greater than 0.

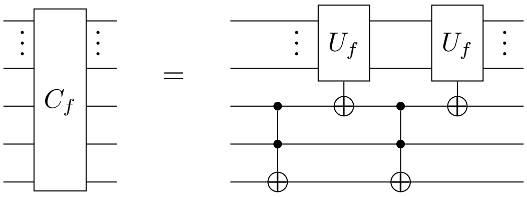
To complete the reduction, given a Boolean expression f build C_f as above in poly time and determine whether a T-count 0 implementation exists. If the minimal T-count is greater than 0, f is non-constant and hence satisfiable. If instead the minimal T-count is 0, then either f is not satisfiable or it is always satisfiable. We can distinguish between these two cases by evaluating $f(0 \cdots 0)$. If $f(0 \cdots 0) = 1$, then f is satisfiable, and otherwise we conclude that it must not be satisfiable.

Note that the exact value of $T = Z(\frac{\pi}{4})$ is not that special. The argument continues to hold for any $Z(\alpha)$, as long as the resulting Clifford+$Z(\alpha)$ gate set allows you to construct the U_f classical oracle. Hence, this argument also shows hardness of optimising the number of $Z(\frac{\pi}{2^n})$ for $n \geq 2$. In Section 3 we show how to modify the argument to prove hardness of optimising *any* non-Clifford gate.

In addition, we only had to distinguish here between a T-count of zero, and a non-zero T-count. This means that similar arguments would also hold for modified cost functions. We could for instance consider the decision problem of T-DEPTH, that asks whether a given circuit has an implementation that requires at most k layers of parallel T gates. Of course any Clifford circuit will have a T-depth of zero, while any non-Clifford circuit will have a T-depth of at least one. Hence, the same argument as above shows that T-DEPTH is also NP-hard. We can also consider the problem of IS-CLIFFORD, which asks whether the given Clifford+T circuit implements a Clifford unitary, i.e. whether it's T-count is zero. We then also see that IS-CLIFFORD is NP-hard.

2.1 Hardness of Toffoli-count optimisation

We can use a similar argument to the one above to show that the problem of TOFFOLI-COUNT, determining the minimal number of Toffoli gates needed to write down a classical reversible circuit (i.e. a quantum circuit consisting of NOT, CNOT and Toffoli gates), is also NP-hard. We then replace the C_f of Eq. (1) by the following:



(2)

Again if f is not satisfiable, C_f implements the identity, and if it is always satisfiable, then it implements a CNOT on the bottom two qubits. In both cases the Toffoli-count is zero. Otherwise if $f(\vec{z}_1) = 1$ and $f(\vec{z}_2) = 0$ we can check that $C_f^\dagger X^{\vec{z}_1 \oplus \vec{z}_2} C_f \left|\vec{z}_1, x, y, z\right\rangle = \left|\vec{z}_2, x, y, y \oplus z\right\rangle$, so that C_f is not Clifford, and hence its Toffoli count is not zero.

As above, this also shows that the problem TOF-DEPTH is NP-hard. We can also consider the problem IS-LINEAR, which asks whether that classical reversible circuit implements a linear Boolean function (one which can be implemented solely using XOR and NOT operations), and we then see that this is also NP-hard.

2.2 Hardness of CNOT-count optimisation

We can reuse the argument for NP-hardness of T-COUNT to prove the hardness of optimising the number of entangling gates in a Clifford+T circuit. In the case when $CNOT$ is the only multi-qubit gate over a basis of the Clifford+T operators, as in the canonical generators $\{H, T, CNOT, S := T^2\}$, this implies the hardness of $CNOT$-count optimization. Likewise, if CZ is used in place of the $CNOT$ gate as the single entangling gate, this implies optimization of CZ-count is again NP-hard.

We define the problem of E-COUNT analogously to T-COUNT, where we replace the role of the T gate by any multi-qubit (entangling) Clifford+T gate E, for instance $CNOT$, CZ, or an entangling product of Clifford and T gates. We require that E is a Clifford+T operator in order to guarantee that the circuit C_f admits implementation over single-qubit Clifford+T and E. We recall from above that for C_f in Eq. (1), if f is not satisfiable, $C_f = \mathrm{id}$, and hence it has E-COUNT zero. If instead f is always satisfiable, $C_f = e^{i\frac{\pi}{4}}(I_n \otimes S^\dagger)$ so that its E-COUNT is also zero. Now suppose f is non-constant and pick $\vec{z}_1$ and $\vec{z}_2$ such that $f(\vec{z}_1) = 1$ and $f(\vec{z}_2) = 0$. We can observe that:

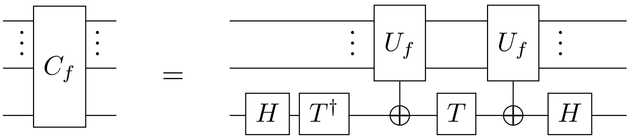
$$\begin{aligned}
C_f \left|\vec{z}_1, 0\right\rangle &= e^{i\frac{\pi}{4}} \left|\vec{z}_1, 0\right\rangle \\
C_f \left|\vec{z}_2, 0\right\rangle &= \left|\vec{z}_2, 0\right\rangle \\
C_f \left|\vec{z}_1, 1\right\rangle &= e^{-i\frac{\pi}{4}} \left|\vec{z}_2, 1\right\rangle \\
C_f \left|\vec{z}_2, 1\right\rangle &= \left|\vec{z}_2, 1\right\rangle
\end{aligned}$$

It can be observed that the right-hand side above is non-separable, and in particular C_f is entangling on this 4-dimensional subspace. Since C_f is itself entangling, it can't be written as a product of non-entangling operators, and so at least one E gate is necessary. Hence, the ability to determine the optimal E count of C_f allows us to determine whether f is satisfiable.

As with the situation for T-COUNT, the same argument can be used to argue that E-DEPTH is NP-hard.

2.3 Hardness of Hadamard-count optimization

Modifying the T-count optimization hardness argument in yet another different way also suffices to prove hardness for the H-COUNT optimisation problem — determining the minimal number of hadamard gates needed to implement a circuit over the canonical generators $\{H, T, CNOT, S\}$ of the Clifford+T gate set. In particular, by conjugating the target bit with hadamard gates as below, it can be observed that C_f can be implemented with H-count zero if and only if f is unsatisfiable.



(3)

In particular, if f is unsatisfiable, then C_f is the identity. If however f is satisfiable, then there exists at least one bit string $\vec{z}$ such that $f(\vec{z}) = 1$, and so where the control register is in the state $|\vec{z}\rangle$, C_f implements the following transformation on the target bit:

$$HXTXT^\dagger = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

which can not be implemented over $\{H, T, CNOT, S\}$ without at least one hadamard gate (and hence neither can its controlled version), as the hadamard-free circuits over the canonical basis correspond to generalized permutations (permutations of computational basis states together with added phases on these basis states). More generally, optimising the H-count is hard over a gate set $\mathcal{G}$ whenever it generates the Clifford+T circuits, and $\mathcal{G} \setminus \{H\}$ contains only Z- and $X-$basis transformations.

The fact that the H-COUNT optimisation problem is NP-hard is not surprising given its close connection and applications to T-count optimisation. In particular, the number of distinct n-qubit unitaries over Clifford+T with a fixed number of hadamard gates is finite [2] and the minimal number of T-gates in an n-qubit Clifford+T circuit with k hadamard gates is bounded above by both $O(k \cdot n)$ and $O((n+k)^2)$ [4]. Optimisation of the H-count over Clifford+T hence has significant application to the problem of T-count optimisation.

3 Hardness of general approximate circuit optimisation

We can adapt the above arguments to show hardness of optimising the number of any specific non-Clifford gate.

Definition 7. Let G be any unitary quantum gate. We define the decision problem G-COUNT$_\varepsilon$ as follows: given the inputs

- C, a circuit specified in the Clifford+G gate set;

- k, an integer;
- ε; an error bound in $\mathbb{R}_{>0}$;

determine whether there exists a circuit C' over the inverse-closed Clifford+G gate set using at most k G or $G^\dagger$ gates, such that for some global phase α we have $\|C - e^{i\alpha}C'\|_\infty \leq \epsilon$.

Note that G-COUNT$_\varepsilon$ is asking whether there exists any circuit close to the target circuit with some G-count. It hence allows us to determine the exact optimal G-count of circuits close to the target. This is different then asking to approximate the G-count itself.

Theorem 8. For any non-Clifford gate G, G-COUNT$_\varepsilon$ is NP-hard under polynomial-time Turing reductions for error bounds $\varepsilon < \sin(\frac{\pi}{16}) \approx 0.195$.

Proof. We modify the argument we used above: reducing from Boolean satisfiability, by asking about the approximate G-count of the circuit C_f in Eq. (1). Note that as a circuit over Clifford+T, C_f might not be exactly expressible as a circuit over Clifford+G. Suppose first that this is the case — for instance, if $G = \sqrt{T}$. We show that if f is non-constant, then at least one G gate is required to approximate C_f to distance ϵ over the Clifford+G gate set. In particular, assume that $f(\vec{z}_1) = 1$ and $f(\vec{z}_2) = 0$ for some vectors $\vec{z}_1$ and $\vec{z}_2$, and let U be a Clifford. We will find a lower bound on $\|C_f - e^{i\alpha}U\|_\infty$.

First we note that any global phase multiple of a non-diagonal Clifford has distance at least $\frac{1}{2}$ from C_f. In particular, assume U is not diagonal and let $|\vec{x}, y\rangle$ be some computational basis state such that $U|\vec{x}, y\rangle \not\propto |\vec{x}, y\rangle$. Since $U|\vec{x}, y\rangle$ is a stabiliser state, $\||\vec{x}, y\rangle - U|\vec{x}, y\rangle\|_2 \geq \frac{1}{2}$ [14], and in particular since C_f is diagonal,

$$\|C_f - e^{i\alpha}U\|_\infty \geq \|C_f|\vec{x}, y\rangle - e^{i\alpha}U|\vec{x}, y\rangle\|_2 \geq \frac{1}{2}.$$

Now suppose U is diagonal. Then U has the form $U|\vec{x}, y\rangle = e^{i\phi(\vec{x},y)}|\vec{x}, y\rangle$ where ϕ is some quadratic function of $\vec{x}$ and y taking values in $\frac{\pi}{2}\mathbb{Z}$. Then for any $\vec{x}$ and y:

$$\|C_f - e^{i\alpha}U\|_\infty \;\geq\; \|C_f|\vec{x}, y\rangle - e^{i\alpha}U|\vec{x}, y\rangle\|_2 \;=\; |e^{i\frac{\pi}{4}(1-2y)f(\vec{x})} - e^{i\alpha}e^{i\phi(\vec{x},y)}|.$$

Plugging in $\vec{x} = \vec{z}_1$ and $\vec{x} = \vec{z}_2$, we then find that for U to be an ε-approximation of C_f, we must at least satisfy:

$$|e^{i\frac{\pi}{4}}e^{-i\frac{\pi}{2}y} - e^{i\alpha}e^{i\phi(\vec{z}_1,y)}| \;\leq\; \varepsilon \qquad \text{and} \qquad |1 - e^{i\alpha}e^{i\phi(\vec{z}_2,y)}| \;\leq\; \varepsilon.$$

Assuming without loss of generality that $0 \leq \alpha \leq \frac{\pi}{4}$ (since if it is outside this bound, a $\frac{\pi}{2}$ phase can be extracted into the Clifford circuit itself), we see that to minimise the value of both of these terms, we need to have $\phi(\vec{z}_2, y) = 0$, $\phi(\vec{z}_1, y) = -\frac{\pi}{2}y$ and $\alpha = \frac{\pi}{8}$. Any other value of ϕ would just increase the value of the expression, while changing α would decrease one at the cost of the other. Hence, the closest a global phase multiple of a Clifford can get to approximating C_f is at least $|1 - e^{i\frac{\pi}{8}}| = 2\sin(\frac{\pi}{16}) \approx 0.39$.

However, all of this is assuming that C_f can be exactly implemented as a Clifford+G circuit, so that it can be given as an input to G-COUNT$_\varepsilon$. This is not the case in general. Note though that if G is non-Clifford that Clifford+G will be approximately universal, and hence that it can approximate any unitary. In particular, due to the Solovay-Kitaev theorem it can approximate any other finite gate set with polylogarithmic overhead in the error. We can then translate every T gate in C_f into a circuit over the Clifford+G gate set, giving a circuit C'_f that is within ε of C_f. Since G and T are fixed, and there are a polynomial number of them, this takes polynomial time in $\log 1/\varepsilon$. Now, if f is always satisfiable or not satisfiable, then C_f is Clifford, so that C'_f is within ε to being a Clifford. Since we only have to estimate up to ε, a Clifford circuit will do, and hence the G-count is zero. Conversely, suppose the G-count of C'_f is zero, so that it is within ε of some Clifford. Using the triangle inequality, C_f must then be within 2ε of some Clifford. However, by assumption $2\varepsilon < 2\sin(\frac{\pi}{16}) = |1 - e^{i\frac{\pi}{8}}|$ so this is only possible if C_f is Clifford itself by the argument above, so that f must be not satisfiable or everywhere satisfiable. Hence, if the G-count of C'_f is greater than zero, then f is satisfiable.

The full reduction is then as follows: for a Boolean formula, construct C_f, written in the Clifford+T gate set. Using the Solovay-Kitaev algorithm approximate the T gate using Clifford+G closely enough so that we get an approximation of C_f that is within ε in the operator norm. Ask whether $G\text{-COUNT}_\varepsilon(C'_f, 0)$ for some $\varepsilon < \sin(\frac{\pi}{16})$ is true. If it is, then the G-count is zero, and hence the circuit is well-approximated by a Clifford, which is only possible if f is not satisfiable, or if f is always satisfiable. Test $f(0\cdots 0)$ to see whether it is indeed always satisfiable. If not, then it is not satisfiable. If $G\text{-COUNT}_\varepsilon(C'_f, 0)$ is false, then any approximation to C'_f and hence to C_f can only be a non-Clifford circuit. Hence, f must be satisfiable. □

The only thing we needed for this argument to work, was for the cost function (in this case: the number of G gates) to distinguish between Clifford unitaries, and non-Clifford unitaries. Hence, we can generalise the above problem to gate sets containing multiple non-Clifford gates $G_1, \ldots, G_m$, and any cost function

$$f : \{\text{circuits over Clifford}+\{G_1, \ldots, G_m\}\} \to \mathbb{R}_{\geq 0}$$

as long as $f(C) = 0$ means C implements a Clifford unitary, and $f(C) > 0$ means C implements a non-Clifford unitary. Hence, determining the optimal G-depth, instead of G-count, will also be NP-hard.

4 An upper bound to the hardness of T-count and Toffoli optimisation

The arguments above show that T-COUNT and its associated optimisation problem are at least NP-hard (and the same for TOFFOLI-COUNT). Let us also demonstrate a simple upper bound to the T-COUNT problem.

Proposition 9. The T-COUNT problem is contained in NP^{NQP}.

We first recall that determining whether two poly-size quantum circuits are exactly equal is a coNQP-complete problem [29] (non-deterministic quantum polynomial time). Note that a QMA oracle [7] is not enough since we care about exact equality. Now to determine whether a given n-qubit circuit C has an implementation with at most k T gates, we realise first that such a circuit can be made to have at most $O(n^2 k)$ Clifford+T gates: any pure Clifford circuit can be represented by a normal form consisting of $O(n^2)$ gates [1], and hence a general Clifford+T circuit containing k T gates can be written as a series of Clifford normal forms followed by a T gate, with this structure repeated k times (for the specific case of Clifford+T this can actually be improved to $O(nk + n^2)$ by the use of Pauli exponentials [23]). Hence, we can non-deterministically choose any circuit with up to k T gates in non-deterministic poly-time, and then use an NQP oracle to determine whether this circuit is equal to C. Hence T-COUNT is in NP^{NQP}.

Note that classical Boolean circuit minimisation is complete for $\Sigma_2^P := \text{NP}^{\text{NP}}$ [8], so that the only difference with this bound is that we replace the coNP problem of determining whether Boolean circuits are equal, with the coNQP problem of doing the same for quantum circuits.

The argument above works for determining an upper bound of exact optimisation of Clifford+G circuits for any non-Clifford G, which hence includes TOF-COUNT.

For G-COUNT$_\varepsilon$ one might expect that we would require instead $\mathrm{NP}^{\mathrm{QMA}}$, as QMA allows one to check whether two quantum circuits are approximately equal. However, this only works as a promise problem where either the circuits have to be closer than a certain bound ε *or* more different then some other bound ε' and these bounds need to be 'far apart'. However, in this case where we are generating candidate circuits non-deterministically, we have no such promise. Instead, we wish to solve the non-promise problem of whether given circuits U and V are ε-close. The contravariant form is then determining whether there exists a normalised state $|\psi\rangle$ and global phase α such that $\|(U - e^{i\alpha}V)|\psi\rangle\|_2 > \varepsilon$. If $|\psi\rangle$ is efficiently representable as a tensor, this calculation of the norm can be represented as a tensor contraction, and hence is in $P^{\#P}$. The overall problem would then be in $\mathrm{NP}^{\#P}$, since we are non-deterministically choosing a candidate state $|\psi\rangle$. An upper bound to the hardness of G-COUNT$_\varepsilon$ would then be $\mathrm{NP}^{\mathrm{NP}^{\#P}}$. However, we do not have such a promise that $|\psi\rangle$ is efficiently preparable. Whether $|\psi\rangle$ can always be chosen in such a manner is in fact exactly the question of whether QMA = QCMA [32]. We have not managed to find any non-conditional upper bounds to G-COUNT$_\varepsilon$ and leave this for future work.

5 Conclusion and outlook

We have shown that, as has long been suspected, many problems in quantum circuit optimisation are indeed hard. In particular, the following problems are all NP-hard:

- Optimising T-count or T-depth.
- Optimisation of T-count or T-depth of unitaries only matching the target unitary within some error-bound on the norm.
- Optimising Toffoli count or depth of classical reversible circuits.
- For any non-Clifford gate G, optimising the G-count or G-depth of Clifford+G unitaries within some error bound on a target unitary.
- Optimising the number of CNOT gates in a Clifford+T circuit.
- Optimising the number of Hadamard gates in a Clifford+T circuit.

A number of open questions remain:

- What is the exact hardness of T-COUNT? It would make sense for this to be a complete problem for $\mathrm{NP}^{\mathrm{NQP}}$, but there are some problems with NQP requiring exact equality, while Clifford+T is only approximately universal, so what is the right conclusion to make here. We could somewhat artificially restrict NQP to NQP_T where it only deals with Clifford+T circuits, but even then adapting the classical proof of the completeness of Boolean circuit optimisation for $\mathrm{NP}^{\mathrm{NP}}$ seems difficult.
- Is it still hard to approximate the optimal T-count within a certain small error? Does it matter whether this error bound is additive or multiplicative?
- For Clifford+T we know that exact optimisation and approximate optimisation are both NP-hard, but for general non-Clifford gates G we only know that approximate optimisation of Clifford+G circuits is NP-hard (for small enough error bounds). Is exact optimisation of the G-count of Clifford+G circuits NP-hard?

- Optimising circuits consisting of only CNOT gates is a well-studied problem [25] and is closely related to optimising steps in a Gaussian elimination of a linear system. It seems likely that this problem is also at least NP-hard, but our methods do not suffice to prove that.

Acknowledgments: The authors wish to thank Robin Kothari for pointing out that our original upper bound to the T-count problem of $\mathrm{NP}^{\mathrm{NP}^{\#\mathrm{P}}}$ can be improved to $\mathrm{NP}^{\mathrm{NQP}}$, and Tuomas Laakkonen for suggesting that our hardness argument might also apply to Toffoli gate optimisation.

References

[1] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004.

[2] Matthew Amy, Jianxin Chen, and Neil J. Ross. A Finite Presentation of CNOT-Dihedral Operators. In *Proceedings of the 14th International Conference on Quantum Physics and Logic*, QPL '17, pages 84–97, 2017.

[3] Matthew Amy, Dmitri Maslov, and Michele Mosca. Polynomial-time T-depth optimization of Clifford+ T circuits via matroid partitioning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(10):1476–1489, 2014.

[4] Matthew Amy and Michele Mosca. T-count optimization and Reed-Muller codes. *Transactions on Information Theory*, 2019.

[5] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan. Quantum algorithms for quantum chemistry and quantum materials science. *Chemical Reviews*, 120(22):12685–12717, 2020.

[6] Michael E Beverland, Aleksander Kubica, and Krysta M Svore. Cost of universality: A comparative study of the overhead of state distillation and code switching with color codes. *PRX Quantum*, 2(2):020341, 2021.

[7] Adam D Bookatz. QMA-complete problems. Preprint, 2012.

[8] David Buchfuhrer and Christopher Umans. The complexity of boolean formula minimization. *Journal of Computer and System Sciences*, 77(1):142–153, 2011.

[9] Jaeho Choi and Joongheon Kim. A tutorial on quantum approximate optimization algorithm (qaoa): Fundamentals and applications. In *2019 international conference on information and communication technology convergence (ICTC)*, pages 138–142. IEEE, 2019.

[10] Niel de Beaudrap, Xiaoning Bian, and Quanlong Wang. Fast and Effective Techniques for T-Count Reduction via Spider Nest Identities. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*, volume 158 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:23, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[11] Niel de Beaudrap and Dominic Horsman. The ZX calculus is a language for surface code lattice surgery. *Quantum*, 4, 2020.

[12] Bryan Eastin and Emanuel Knill. Restrictions on transversal encoded quantum gate sets. *Physical review letters*, 102(11):110502, 2009.

[13] 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, 2012.

[14] Héctor J. García, Igor L. Markov, and Andrew W. Cross. On the geometry of stabilizer states. *Quantum Information and Computation*, 14:683–720, 2014.

[15] Vlad Gheorghiu, Michele Mosca, and Priyanka Mukhopadhyay. A (quasi-) polynomial time heuristic algorithm for synthesizing t-depth optimal circuits. *npj Quantum Information*, 8(1):110, 2022.

[16] Vlad Gheorghiu, Michele Mosca, and Priyanka Mukhopadhyay. T-count and t-depth of any multi-qubit unitary. *npj Quantum Information*, 8(1):141, 2022.

[17] Craig Gidney and Martin Ekerå. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*, 5:433, April 2021.

[18] Craig Gidney and Austin G. Fowler. Efficient magic state factories with a catalyzed $|CCZ\rangle$ to $2|T\rangle$ transformation. *Quantum*, 3:135, 4 2019.

[19] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[20] Luke E Heyfron and Earl T Campbell. An efficient quantum compiler that reduces T count. *Quantum Science and Technology*, 4(015004), 2018.

[21] Clare Horsman. Quantum picturalism for topological cluster-state computing. *New Journal of Physics*, 13(9):095011, 2011.

[22] Aleks Kissinger and John van de Wetering. Reducing the number of non-Clifford gates in quantum circuits. *Physical Review A*, 102:022406, 8 2020.

[23] Daniel Litinski. A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery. *Quantum*, 3:128, 3 2019.

[24] Daniel Litinski. Magic State Distillation: Not as Costly as You Think. *Quantum*, 3:205, December 2019.

[25] Ketan Patel, Igor Markov, , and John Hayes. Optimal synthesis of linear reversible circuits. *Quantum Information and Computation*, 8(3&4):0282–0294, 2008.

[26] Francisco JR Ruiz, Tuomas Laakkonen, Johannes Bausch, Matej Balog, Mohammadamin Barekatain, Francisco JH Heras, Alexander Novikov, Nathan Fitzpatrick, Bernardino Romera-Paredes, John van de Wetering, et al. Quantum circuit optimization with alphatensor. *arXiv preprint arXiv:2402.14396*, 2024.

[27] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

[28] Peter W Shor. Scheme for reducing decoherence in quantum computer memory. *Physical review A*, 52(4):R2493, 1995.

[29] Yu Tanaka. Exact non-identity check is nqp-complete. *International Journal of Quantum Information*, 8(05):807–819, 2010.

[30] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H Booth, et al. The variational quantum eigensolver: a review of methods and best practices. *Physics Reports*, 986:1–128, 2022.

[31] John van de Wetering, Richie Yeung, Tuomas Laakkonen, and Aleks Kissinger. Optimal compilation of parametrised quantum circuits. *arXiv preprint arXiv:2401.12877*, 2024.

[32] Pawel Wocjan, Dominik Janzing, and Thomas Beth. Two qcma-complete problems. *arXiv preprint quant-ph/0305090*, 2003.

[33] Fang Zhang and Jianxin Chen. Optimizing T gates in Clifford+T circuit as $\pi/4$ rotations around Paulis. Preprint, 2019.