FAST AND EFFICIENT EXACT SYNTHESIS OF SINGLE-QUBIT UNITARIES GENERATED BY CLIFFORD AND T GATES

VADYM KLIUCHNIKOV

Institute for Quantum Computing and School of Computer Science
University of Waterloo, 200 University Avenue West, Waterloo, Ontario, N2L 3G1, Canada

DMITRI MASLOV

National Science Foundation
4201 Wilson Boulevard, Arlington, Virginia 22230, USA
Institute for Quantum Computing, and Dept. of Physics & Astronomy
University of Waterloo, 200 University Avenue West, Waterloo, Ontario, N2L 3G1, Canada

MICHELE MOSCA

Institute for Quantum Computing, and Dept. of Combinatorics & Optimization
University of Waterloo, 200 University Avenue West, Waterloo, Ontario, N2L 3G1, Canada
Perimeter Insitute for Theoretical Physics
31 Caroline Street North, Waterloo, Ontario, N2L 2Y5, Canada

In this paper, we show the equivalence of the set of unitaries computable by the circuits over the Clifford and T library and the set of unitaries over the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$, in the single-qubit case. We report an efficient synthesis algorithm, with an exact optimality guarantee on the number of Hadamard and T gates used. We conjecture that the equivalence of the sets of unitaries implementable by circuits over the Clifford and T library and unitaries over the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ holds in the n-qubit case.

Keywords: circuit synthesis; Solovay-Kitaev algorithm

Communicated by: I Cirac & J Eisert

1 Introduction

The problem of efficient approximation of an arbitrary unitary using a finite gate set is important in quantum computation. In particular, fault tolerance methods impose limitations on the set of elementary gates that may be used on the logical (as opposed to physical) level. One of the most common of such sets consists of Clifford[a] and T:= $\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$ gates. This gate library is known to be approximately universal in the sense of the existence of an efficient approximation of the unitaries by circuits over it. In the single-qubit case, the standard solution to the problem of unitary approximation by circuits over a gate library is given by the Solovay-Kitaev algorithm [1]. The multiple qubit case may be handled via employing

---

[a]Also known as stabilizer gates/library. In the single-qubit case the Clifford library consists of, e.g., Hadamard and Phase gates. In the multiple qubit case, the two-qubit CNOT gate is also included in the Clifford library.

results from [2] that show how to decompose any n-qubit unitary into a circuit with CNOT and single-qubit gates. Given precision ε, the Solovay-Kitaev algorithm produces a sequence of gates of length $O\left(\log^{c}\left(1/\varepsilon\right)\right)$ and requires time $O\left(\log^{d}\left(1/\varepsilon\right)\right)$, for positive constants c and d.

While the Solovay-Kitaev algorithm provides a provably efficient approximation, it does not guarantee finding an exact decomposition of the unitary into a circuit if there is one, nor does it answer the question of whether an exact implementation exists. We refer to these as the problems of *exact* synthesis. Studying the problems related to exact synthesis is the focus of our paper. In particular, we study the relation between single-qubit unitaries and circuits composed with Clifford and T gates. We answer two main questions: first, given a unitary how to efficiently decide if it can be synthesized exactly, and second, how to find an efficient gate sequence that implements a given single-qubit unitary exactly (limited to the scenario when such an implementation exists, which we know from answering the first of the two questions). We further provide some intuition about the multiple qubit case.

Our motivation for this study is rooted in the observation that the implementations of quantum algorithms exhibit errors from multiple sources, including (1) algorithmic errors resulting from the mathematical probability of measuring a correct answer being less than one for many quantum algorithms [3], (2) errors due to decoherence [3], (3) systematic errors and imperfections in controlling apparatus (e.g., [4]), and (4) errors arising from the inability to implement a desired transformation exactly using the available finite gate set requiring one to resort to approximations. Minimizing the effect of errors has direct implications on the resources needed to implement an algorithm and sometimes determines the very ability to implement a quantum algorithm and demonstrate it experimentally on available hardware of a specific size. We set out to study the fourth type of error, rule those out whenever possible, and identify situations when such approximation errors cannot be avoided. During the course of this study we have also identified that we can prove certain tight and constructive upper bounds on the circuit size for those unitaries that may be implemented exactly. In particular, we report a single-qubit circuit synthesis algorithm that guarantees optimality of both Hadamard and T gate counts.

The remainder of the paper is organized as follows. In the next section, we summarize and discuss our main results. Follow up sections contain necessary proofs. In Section 2, we reduce the problem of single-qubit unitary synthesis to the problem of state preparation. In Section 3, we discuss two major technical Lemmas required to prove our main result summarized in Theorem 1. We also present an algorithm for efficient decomposition of single-qubit unitaries in terms of Hadamard, H:= $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, and T gates. Section 5 and Appendix A flesh out formal proofs of minor technical results used in Section 4. Appendix B contains a proof showing that the number of Hadamard and T gates in the circuits produced by Algorithm 1 is minimal.

2   Formulation and discussion of the results

One of our two main results reported in this paper is the following theorem:

Theorem 1 *The set of 2×2 unitaries over the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ is equivalent to the set of those*

unitaries implementable exactly as single-qubit circuits constructed using[b]H and T gates only.

The inclusion of the set of unitaries implementable exactly via circuits employing H and T gates into the set of 2×2 unitaries over the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ is straightforward, since, indeed, all four elements of each of the unitary matrices H and T belong to the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$, and circuit composition is equivalent to matrix multiplication in the unitary matrix formalism. Since both operations used in the standard definition of matrix multiplication, "+" and "×", applied to the ring elements, clearly do not take us outside the ring, each circuit constructed using H and T gates computes a matrix whose elements belong to the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$. The inverse inclusion is more difficult to prove. The proof is discussed in Sections 3-5 and Appendix A.

We believe the statement of the Theorem 1 may be extended and generalized into the following conjecture:

Conjecture 1 *For $n > 1$, the set of $2^n \times 2^n$ unitaries over the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ is equivalent to the set of unitaries implementable exactly as circuits with Clifford and T gates built using $(n + 1)$ qubits, where the last qubit, an ancillary qubit, is set to the value $|0\rangle$ prior to the circuit computation, and is required to be returned in the state $|0\rangle$ at the end of it.*

Note, that the ancillary qubit may not be used if its use is not required. However, we next show that the requirement to include a single ancillary qubit is essential—if removed, the statement of Conjecture 1 would have been false. The necessity of this condition is tantamount to the vast difference between single-qubit case and n-qubit case for $n > 1$.

We wish to illustrate the necessity of the single ancilla with the use of controlled-T gate, defined as follows:

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

where $\omega := e^{2\pi i/8}$, the eighth root of unity. The determinant of this unitary is ω. However, any Clifford gate as well as the T gate viewed as matrices over a set of two qubits have a determinant that is a power of the imaginary number i. Using the multiplicative property of the determinant we conclude that the circuits over the Clifford and T library may implement only those unitaries whose determinant is a power of the imaginary i. As such, the controlled-T, whose determinant equals ω, cannot be implemented as a circuit with Clifford and T gates built using only two qubits. It is also impossible to implement the controlled-T up to global phase. The reason is that the only complex numbers of the form $e^{i\phi}$ that belong to the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ are ω^k for integer k, as it is shown in Appendix A. Therefore, global phase can only change determinant by a multiplicative factor of ω^{4k}. However, as reported in [5] and illustrated in Figure 1, an implementation of the controlled-T over a set of three qubits, one of which is set to and returned in the state $|0\rangle$, exists. With the addition of an ancillary qubit, as described, the determinant argument fails, because one would now need to look at the determinant of a subsystem, that, unlike the whole system, may be manipulated in such a way as to allow the computation to happen.

Theorem 1 provides an easy to verify criteria that reliably differentiates between unitaries implementable in the H and T library and those requiring approximation. As an example,

---

[b]Note, that gate H may be replaced with all Clifford group gates without change to the meaning, though may help to visually bridge this formulation with the formulation of the follow-up general conjecture.

Fig. 1. Circuit implementing the controlled-T gate, with upper qubit being the control, middle qubit being the target, and bottom qubit being the ancilla. Reprinted from [5].

$R_x(\frac{\pi}{3})$ and gates such as $R_z(\frac{\pi}{2^m})$, where $m > 3$, popular in the construction of circuits for the Quantum Fourier Transform (QFT), cannot be implemented exactly and must be approximated. Thus, the error in approximations may be an unavoidable feature for certain quantum computations. Furthermore, Conjecture 1, whose one inclusion is trivial—all Clifford and T circuits compute unitaries over the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$—implies that the QFT over more than three qubits may not be computed exactly as a circuit with Clifford and T gates, and must be approximated.

Our second major result is an algorithm (Algorithm 1) that synthesizes a quantum single-qubit circuit using gates H, Z:=T^4, P:=T^2, and T in time $O(n_{opt})$, where n_{opt} is the minimal number of gates required to implement a given unitary. Technically, the above complexity calculation assumes that the operations over the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ take a fixed finite amount of time. In terms of bit operations, however, this time is quadratic in n_{opt}. Nevertheless, assuming ring operations take constant time, the efficiency has a surprising implication. In particular, it is easy to show that our algorithm is asymptotically optimal, in terms of both its speed and quality guarantees, among all algorithms (whether known or not) solving the problem of synthesis in the single-qubit case. Indeed, a natural lower bound to accomplish the task of synthesizing a unitary is n_{opt}—the minimal time it takes to simply write down an optimal circuit assuming a certain algorithm somehow knows what it actually is. Our algorithm features the upper bound of $O(n_{opt})$ matching the lower bound and implying asymptotic optimality. To state the above somewhat differently, the problem in approximating a unitary by a circuit is that of finding an approximating unitary with elements in the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$, but not composing the circuit itself. We formally show H- and T-optimality of the circuits synthesized by Algorithm 1 in Appendix B.

The T-optimality of circuit decompositions has been a topic of study of the recent paper [6]. We note that our algorithm guarantees both T- and H-optimality, whereas the one reported in [6] guarantees only T-optimality. Furthermore, our implementation allows a trade-off between the number of Phase and Pauli-Z gates (the number of other gates used, being Pauli-X and Pauli-Y, does not exceed a total of three). We shared our software implementation and circuits obtained from it to facilitate proper comparison of the two synthesis algorithms.

In the recent literature, similar topics have also been studied in [5] who concentrated on finding depth-optimal multiple qubit quantum circuits in the Clifford and T library, [7] who developed a normal form for single-qubit quantum circuits using gates H, P, and T, and [1, 8] who considered improvements of the Solovay-Kitaev algorithm that are very relevant to our work. In fact, we employ the Solovay-Kitaev algorithm as a tool to find an approximating unitary that we can then synthesize using our algorithm for exact single-qubit unitary synthesis.

3   Reducing unitary implementation to state preparation

In this section we discuss the connection between state preparation and implementation of a unitary by a quantum circuit. In the next section, we prove the following result:

Lemma 1 *Any single-qubit state with entries in the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ can be prepared using only H and T gates given the initial state $|0\rangle$.*

We first establish why Lemma 1 implies that any single-qubit unitary with entries in the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ can be implemented exactly using H and T gates.

Observe that any single-qubit unitary can be written in the form

$$\left(\begin{array}{cc} z & -w^* e^{i\phi} \\ w & z^* e^{i\phi} \end{array} \right),$$

where * denotes the complex conjugate. The determinant of the unitary is equal to $e^{i\phi}$ and belongs to the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ when all entries of the unitary belong to the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$. It turns out that the only elements in the ring with the absolute value of 1 are ω^k for integer k. We postpone the proof; it follows from techniques developed in Appendix A and discussed at the end of the appendix. For now, we conclude that the most general form of a unitary with entries in the ring is:

$$\left(\begin{array}{cc} z & -w^* \omega^k \\ w & z^* \omega^k \end{array} \right).$$

We next show how to find a circuit that implements any such unitary when we know a circuit that prepares its first column given the state $|0\rangle$. Suppose we have a circuit that prepares state $\left(\begin{array}{c} z \\ w \end{array} \right)$. This means that the first column of a unitary corresponding to the circuit is $\left(\begin{array}{c} z \\ w \end{array} \right)$ and there exists an integer k' such that the unitary is equal to:

$$\left(\begin{array}{cc} z & -w^* \omega^{k'} \\ w & z^* \omega^{k'} \end{array} \right).$$

We can synthesize all possible unitaries with the first column $(z, w)^t$ by multiplying the unitary above by a power of T from the right:

$$\left(\begin{array}{cc} z & -w^* \omega^{k'} \\ w & z^* \omega^{k'} \end{array} \right) T^{k-k'} = \left(\begin{array}{cc} z & -w^* \omega^k \\ w & z^* \omega^k \end{array} \right).$$

This also shows that given a circuit for state preparation of length n we can always find a circuit for unitary implementation of length $n + O(1)$ and vice versa.

4   Sequence for state preparation

We start with an example that illustrates the main ideas needed to prove Lemma 1. Next we formulate two results, Lemma 2 and Lemma 3, that the proof of Lemma 1 is based on. Afterwards, we describe the algorithm for decomposition of a unitary with entries in the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ into a sequence of H and T gates. Finally, we prove Lemma 2. The proof of Lemma 3 is more involved and it is shown in Section 5.

Table 1. First four elements of sequence $(HT)^n |0\rangle$

n	$(HT)^n |0\rangle = \begin{pmatrix} z_n \\ w_n \end{pmatrix}$	$\begin{pmatrix} |z_n|^2 \\ |w_n|^2 \end{pmatrix}$
1	$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\frac{1}{(\sqrt{2})^2} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
2	$\frac{1}{(\sqrt{2})^2} \begin{pmatrix} \omega + 1 \\ 1 - \omega \end{pmatrix}$	$\frac{1}{(\sqrt{2})^3} \begin{pmatrix} \sqrt{2} + 1 \\ \sqrt{2} - 1 \end{pmatrix}$
3	$\frac{1}{(\sqrt{2})^2} \begin{pmatrix} \omega^2 - \omega^3 + 1 \\ \omega \end{pmatrix}$	$\frac{1}{(\sqrt{2})^4} \begin{pmatrix} 3 \\ 1 \end{pmatrix}$
4	$\frac{1}{(\sqrt{2})^3} \begin{pmatrix} 2\omega^2 - \omega^3 + 1 \\ 1 - \omega^3 \end{pmatrix}$	$\frac{1}{(\sqrt{2})^5} \begin{pmatrix} 3\sqrt{2} - 1 \\ \sqrt{2} + 1 \end{pmatrix}$

Let us consider a sequence of states $(HT)^n |0\rangle$. It is an infinite sequence, since in the Bloch sphere picture unitary HT corresponds to rotation over an angle that is an irrational fraction of π. Table 1 shows the first four elements of the sequence.

There are two features in this example that are important. First is that the power of $\sqrt{2}$ in the denominator of the entries is the same. We prove that the power of the denominator is the same in the general case of a unit vector with entries in ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$. The second feature is that the power of $\sqrt{2}$ in the denominator of $|z_n|^2$ increases by 1 after multiplication by HT. We show that in general, under additional assumptions, multiplication by $H\left(T^k\right)$ cannot change the power of $\sqrt{2}$ in the denominator by more than 1. Importantly, under the same additional assumptions it is always possible to find such an integer k that the power increases or decreases by 1.

We need to clarify what we mean by power of $\sqrt{2}$ in the denominator, because, for example, it is possible to write $\frac{1}{\sqrt{2}}$ as $\frac{\omega - \omega^3}{2}$. As such, it may seem that the power of $\sqrt{2}$ in the denominator of a number from the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ is not well defined. To address this issue we consider the subring

$$\mathbb{Z}[\omega] := \left\{ a + b\omega + c\omega^2 + d\omega^3, a, b, c, d \in \mathbb{Z} \right\}$$

of ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ and the smallest denominator exponent. These definitions are also crucial for our proofs.

It is natural to extend the notion of divisibility to elements of $\mathbb{Z}[\omega]$: x divides y when there exists x' from the ring $\mathbb{Z}[\omega]$ such that $xx' = y$. Using the divisibility relation we can introduce the smallest denominator exponent and greatest dividing exponent.

Definition 1 *The* smallest denominator exponent, $\mathrm{sde}(z, x)$, *of base* $x \in \mathbb{Z}[\omega]$ *with respect to* $z \in \mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ *is the smallest integer value* k *such that* $zx^k \in \mathbb{Z}[\omega]$. *If there is no such* k, *the smallest denominator exponent is infinite.*

For example, $\mathrm{sde}(1/4, \sqrt{2}) = 4$ and $\mathrm{sde}\left(2\sqrt{2}, \sqrt{2}\right) = -3$. The smallest denominator exponent of base $\sqrt{2}$ is finite for all elements of the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$. The greatest dividing exponent is closely connected to sde.

Definition 2 *The* greatest dividing exponent, $\mathrm{gde}(z, x)$, *of base* $x \in \mathbb{Z}[\omega]$ *with respect to*

$z \in \mathbb{Z}[\omega]$ *is the integer value* k *such that* x^k *divides* z *and* x *does not divide quotient* $\frac{z}{x^k}$. *If no such* k *exists, the greatest dividing exponent is said to be infinite.*

For example, $\mathrm{gde}\,(z, \omega^n) = \infty$, since ω^n divides any element of $\mathbb{Z}[\omega]$, and $\mathrm{gde}\,(0, x) = \infty$. For any non-zero base $x \in \mathbb{Z}[\omega]$, gde and sde are related via a simple formula:

$$\mathrm{sde}\left(\frac{z}{x^k}, x\right) = k - \mathrm{gde}\,(z, x). \tag{1}$$

This follows from the definitions of sde and gde. First, the assumption $\mathrm{gde}\,(z, x) = k_0$ implies $\mathrm{sde}\left(\frac{z}{x^k}, x\right) \geq k - k_0$. Second, the assumption $\mathrm{sde}\left(\frac{z}{x^k}, x\right) = k_0$ implies $\mathrm{gde}\,(z, x) \geq k + k_0$. Since both inequalities need to be satisfied simultaneously, this implies the equality.

We are now ready to introduce two results that describe the change of the sde as a result of the application $H\,(T)^k$ to a state:

$$HT^k \begin{pmatrix} z \\ w \end{pmatrix} = \begin{pmatrix} \frac{z + w\omega^k}{\sqrt{2}} \\ \frac{z - w\omega^k}{\sqrt{2}} \end{pmatrix}.$$

Lemma 2 *Let* $\begin{pmatrix} z \\ w \end{pmatrix}$ *be a state with entries in* $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ *and let* $\mathrm{sde}\left(|z|^2\right) \geq 4$. *Then, for any integer* k:

$$-1 \leq \mathrm{sde}\left(\left|\frac{z + w\omega^k}{\sqrt{2}}\right|^2\right) - \mathrm{sde}\left(|z|^2\right) \leq 1. \tag{2}$$

The next lemma states that for almost all unit vectors the difference in (2) achieves all possible values, when the power of ω is chosen appropriately.

Lemma 3 *Let* $\begin{pmatrix} z \\ w \end{pmatrix}$ *be a state with entries in* $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ *and let* $\mathrm{sde}\left(|z|^2\right) \geq 4$. *Then, for each number* $s \in \{-1, 0, 1\}$ *there exists an integer* $k \in \{0, 1, 2, 3\}$ *such that:*

$$\mathrm{sde}\left(\left|\frac{z + w\omega^k}{\sqrt{2}}\right|^2\right) - \mathrm{sde}\left(|z|^2\right) = s.$$

These lemmas are essential for showing how to find a sequence of gates that prepares a state with entries in the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ given the initial state $|0\rangle$. Now we sketch a proof of Lemma 1. Later, in Lemma 4, we show that for arbitrary u and v from the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ the equality $|u|^2 + |v|^2 = 1$ implies $\mathrm{sde}(|u|^2) = \mathrm{sde}(|v|^2)$, when $\mathrm{sde}\left(|u|^2\right) \geq 1$ and $\mathrm{sde}\left(|v|^2\right) \geq 1$. Therefore, under assumptions of Lemma 2, we may consider sde of a single entry in any given state. Lemma 3 implies that we can prepare any state using H and T gates if we can prepare any state $\begin{pmatrix} z \\ w \end{pmatrix}$ such that $\mathrm{sde}(|z|^2) \leq 3$. The set of states with $\mathrm{sde}(|z|^2) \leq 3$ is finite and small. Therefore, we can exhaustively verify that all such states can be prepared using H and T gates given the initial state $|0\rangle$. In fact, we performed such verification using a breadth first search algorithm.

The statement of Lemma 3 remains true if we replace the set $\{0, 1, 2, 3\}$ by $\{0, -1, -2, -3\}$. Lemma 3 results in Algorithm 1 for decomposition of a unitary matrix with entries in the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ into a sequence of H and T gates. Its complexity is $O\left(\mathrm{sde}(|z|^2)\right)$, where z is an entry of the unitary. The idea behind the algorithm is as follows: given a 2×2 unitary U over the

ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ and sde ≥ 4, there is a value of k in $\{0, 1, 2, 3\}$ such that the multiplication by $H\left(T^k\right)$ reduces the sde by 1. Thus, after $n - 4$ steps, we have expressed

$$U = HT^{k_1}H \ldots HT^{k_{n-4}}U',$$

where any entry z' of U' has the property sde $\left(|z'|^2\right) < 4$. The number of such unitaries is small enough to handle the decomposition of U' via employing a breadth-first search algorithm.

We use $n_{opt}(U)$ to define the smallest length of the circuit that implements U.

Corollary 1 *Algorithm 1 produces circuit of length $O(n_{opt}(U))$ and uses $O(n_{opt}(U))$ arithmetic operations. The number of bit operations it uses is $O(n_{opt}^2(U))$.*

Proof. Lemma 4, proved later in this section, implies that the value of sde $\left(|\cdot|^2\right)$ is the same for all entries of U when the sde of at least one entry is greater than 0. For such unitaries we define sde$^{|\cdot|^2}(U) = $ sde $\left(|z'|^2\right)$, where z' is an entry of U. The remaining special case is unitaries of the form

$$\begin{pmatrix} 0 & \omega^k \\ \omega^j & 0 \end{pmatrix}, \begin{pmatrix} \omega^k & 0 \\ 0 & \omega^j \end{pmatrix}.$$

We define sde$^{|\cdot|^2}$ to be 0 for all of them. Consider a set $S_{opt,3}$ of optimal H and T circuits for unitaries with sde$^{|\cdot|^2} \leq 3$. This is a finite set and therefore we can define N_3 to be the maximal number of gates in a circuit from $S_{opt,3}$. If we have a circuit that is optimal and its length is greater than N_3, the corresponding unitary must have sde$^{|\cdot|^2} \geq 4$. Consider now a unitary U with an optimal circuit of length $n_g(U)$ that is larger than N_3. As it is optimal, all its subsequences are optimal and it does not include H^2. Let $N_{H,3}$ be the maximum of the number of Hadamard gates used by the circuits in $S_{opt,3}$. An optimal circuit for U includes at most $\left\lfloor \frac{n_g(U) - N_3}{2} \right\rfloor + N_{H,3}$ Hadamard gates and, by Lemma 2, sde$^{|\cdot|^2}$ of the resulting unitary is less or equal to $N_{H,3} + 3 + \left\lfloor \frac{n_g(U) - N_3}{2} \right\rfloor$. We conclude that for all unitaries except a finite set:

$$\text{sde}^{|\cdot|^2}(U) \leq N_{H,3} + 3 + \left\lfloor \frac{n_g(U) - N_3}{2} \right\rfloor.$$

From the other side, the decomposition algorithm we described gives us the bound

$$n_g(U) \leq N_3 + 4 \cdot \text{sde}^{|\cdot|^2}(U).$$

We conclude that $n_g(U)$ and sde$^{|\cdot|^2}(U)$ are asymptotically equivalent. Therefore the algorithm's runtime is $O(n_g(U))$, because the algorithm performs sde$^{|\cdot|^2}(U) - 4$ steps.

We note that to store U we need $O\left(\text{sde}^{|\cdot|^2}(U)\right)$ bits and therefore the addition on each step of the algorithm requires $O\left(\text{sde}^{|\cdot|^2}(U)\right)$ bit operations. Therefore we use $O(n_{opt}^2(U))$ bit operations in total $\square$.

This proof illustrates the technique that we use in Appendix B to find a tighter connection between sde and the circuit implementation cost, in particular we prove that circuits produced by the algorithm are H- and T-optimal.

Algorithm 1 Decomposition of a unitary matrix with entries in the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$.

Input: Unitary $U = \begin{pmatrix} z_{00} & z_{01} \\ z_{10} & z_{11} \end{pmatrix}$ with entries in the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$.

$\quad \mathbb{S}_3$ – table of all unitaries over the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$, such that *sde* of their entries is less than or equal to 3.

Output: Sequence S_{out} of H and T gates that implements U.

$\quad S_{out} \leftarrow Empty$

$\quad s \leftarrow \text{sde}(|z_{00}|^2)$

$\quad$ **while** s>3 **do**

$\quad\quad$ state←unfound

$\quad\quad$ **for all** $k \in \{0, 1, 2, 3\}$ **do**

$\quad\quad\quad$ **while** state = unfound **do**

$\quad\quad\quad\quad z'_{00} \leftarrow$ top left entry of $HT^{-k}U$

$\quad\quad\quad\quad$ **if** $sde\left(|z'_{00}|^2\right) = s - 1$ **then**

$\quad\quad\quad\quad\quad$ state = found

$\quad\quad\quad\quad\quad$ add $T^k H$ to the end of S_{out}

$\quad\quad\quad\quad\quad s \leftarrow \text{sde}\left(|z'_{00}|^2\right)$

$\quad\quad\quad\quad\quad U \leftarrow HT^{-k}U$

$\quad\quad\quad\quad$ **end if**

$\quad\quad\quad$ **end while**

$\quad\quad$ **end for**

$\quad$ **end while**

$\quad$ lookup sequence S_{rem} for U in $\mathbb{S}_3$

$\quad$ add S_{rem} to the end of S_{out}

$\quad$ **return** S_{out}

We next prove Lemma 2. In Section 5 we use Lemma 2 to show that we can prove Lemma 3 by considering a large, but finite, number of different cases. We provide an algorithm (Algorithm 2) that verifies all these cases.

We now proceed to the proof of Lemma 2. We use equation (1) connecting sde and gde together with the following properties of gde. For any base $x \in \mathbb{Z}[\omega]$:

$$\text{gde}\,(y + y', x) \geq \min\left(\text{gde}\,(y, x), \text{gde}\,(y', x)\right) \tag{3}$$

$$\text{gde}\left(yx^k, x\right) = k + \text{gde}\,(y, x) \quad \text{(base extraction)} \tag{4}$$

$$\text{gde}\,(y, x) < \text{gde}\,(y', x) \Rightarrow \text{gde}\,(y + y', x) = \text{gde}\,(y, x) \quad \text{(absorption)}. \tag{5}$$

It is also helpful to note that $\text{gde}\,(y, x)$ is invariant with respect to multiplication by ω and complex conjugation of both x and y.

All these properties follow directly from the definition of gde; the first three are briefly discussed in Appendix A. The condition $\text{gde}\,(y, x) < \text{gde}\,(y', x)$ is necessary for the third property. For example, $\text{gde}\left(\sqrt{2} + \sqrt{2}, \sqrt{2}\right) \neq \text{gde}\left(\sqrt{2}, \sqrt{2}\right)$.

There are also important properties specific to base $\sqrt{2}$. We use shorthand $\text{gde}\,(\cdot)$ for

$\text{gde}\left(\cdot, \sqrt{2}\right)$:

$$\text{gde}\left(x\right) = \text{gde}\left(|x|^2, 2\right) \tag{6}$$

$$0 \leq \text{gde}\left(|x|^2\right) - 2\text{gde}\left(x\right) \leq 1 \tag{7}$$

$$\text{gde}\left(\text{Re}\left(\sqrt{2}xy^*\right)\right) \geq \left\lfloor \frac{1}{2}\left(\text{gde}\left(|x|^2\right) + \text{gde}\left(|y|^2\right)\right)\right\rfloor \tag{8}$$

$$\text{gde}\left(|x|^2\right) = \text{gde}\left(|y|^2\right) \Rightarrow \text{gde}\left(x\right) = \text{gde}\left(y\right). \tag{9}$$

Proofs of these properties are not difficult but tedious; furthermore, for completeness they are included in Appendix A. We exemplify them here. In the second property, inequality (7), when $x = \omega$ the left inequality becomes equality and for $\omega + 1$ the right one does. When we substitute $x = \omega, y = \omega + 1$ in the second to last property, inequality (8), it turns into $0 = \left\lfloor \frac{1}{2} \right\rfloor$, so the floor function $r \mapsto \lfloor r \rfloor$ is necessary. For the third property it is important that $\text{Re}\left(\sqrt{2}xy^*\right)$ is an element of $\mathbb{Z}\left[\omega\right]$ when x, y itself belongs to the ring $\mathbb{Z}\left[\omega\right]$. In contrast, $\text{Re}\left(xy^*\right)$ is not always an element of $\mathbb{Z}\left[\omega\right]$, in particular, when $x = \omega, y = \omega + 1$. In general, $\text{gde}\left(x\right) = \text{gde}\left(y\right)$ does not imply $\text{gde}\left(|x|^2\right) = \text{gde}\left(|y|^2\right)$. For instance, $\text{gde}\left(\omega + 1\right) = \text{gde}\left(\omega\right)$, but $|\omega + 1|^2 = 2 + \sqrt{2}$ and $|\omega|^2 = 1$.

In the proof of Lemma 2 we use $x = z\left(\sqrt{2}\right)^{\text{sde}(z)}, y = w\left(\sqrt{2}\right)^{\text{sde}(w)}$ that are elements of $\mathbb{Z}\left[\omega\right]$. The next lemma shows an additional property that such x and y have.

Lemma 4 *Let z and w be elements of the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ such that $|z|^2 + |w|^2 = 1$ and $\text{sde}\left(z\right) \geq 1$ or $\text{sde}\left(w\right) \geq 1$, then $\text{sde}\left(z\right) = \text{sde}\left(w\right)$ and for elements $x = z\left(\sqrt{2}\right)^{\text{sde}(z)}$ and $y = w\left(\sqrt{2}\right)^{\text{sde}(w)}$ of the ring $\mathbb{Z}\left[\omega\right]$ it holds that $\text{gde}\left(|x|^2\right) = \text{gde}\left(|y|^2\right) \leq 1$.*

Proof. Without loss of generality, suppose $\text{sde}\left(z\right) \geq \text{sde}\left(w\right)$. Using the relation in equation (1) between sde and gde, expressing z and w in terms of x and y, and substituting the result into equation $|z|^2 + |w|^2 = 1$, we obtain

$$|y|^2 \left(\sqrt{2}\right)^{2(\text{sde}(z) - \text{sde}(w))} = \left(\sqrt{2}\right)^{2\text{sde}(z)} - |x|^2.$$

Substituting $z = x/\left(\sqrt{2}\right)^{\text{sde}(z)}$ into formula (1) relating sde and gde, we obtain $\text{gde}\left(x\right) = 0$, and using one of the inequalities (7) connecting $\text{gde}\left(|x|^2\right)$ and $\text{gde}\left(x\right)$ we conclude that $\text{gde}\left(|x|^2\right) \leq 1$. Similarly, $\text{gde}\left(|y|^2\right) \leq 1$. We use the absorption property (5) of $\text{gde}\left(\cdot\right)$ to write:

$$\text{gde}\left(|y|^2 \left(\sqrt{2}\right)^{2(\text{sde}(z) - \text{sde}(w))}\right) = \text{gde}\left(|x|^2\right).$$

Equivalently, using the base extraction property (4):

$$\text{gde}\left(|y|^2\right) + 2\left(\text{sde}\left(z\right) - \text{sde}\left(w\right)\right) = \text{gde}\left(|x|^2\right).$$

Taking into account $\text{gde}\left(|x|^2\right) \leq 1$ and $\text{gde}\left(|y|^2\right) \leq 1$, it follows that $\text{sde}\left(z\right) = \text{sde}\left(w\right)$ $\square$.

In the proof of Lemma 2 we turn inequality (2) for difference of sde into an inequality for difference of $\text{gde}\left(|x|^2\right)$ and $\text{gde}\left(|x + y|^2\right)$. The following lemma shows a basic relation between these quantities that we will use.

Lemma 5 *If x and y are elements of the ring $\mathbb{Z}[\omega]$ such that $|x|^2 + |y|^2 = \left(\sqrt{2}\right)^m$, then*

$$\text{gde}\left(|x+y|^2\right) \geq \min\left(m, 1 + \left\lfloor \frac{1}{2}\left(\text{gde}\left(|x|^2\right) + \text{gde}\left(|y|^2\right)\right)\right\rfloor\right).$$

Proof. The first step is to expand $|x+y|^2$ as $|x|^2 + |y|^2 + \sqrt{2}Re\left(\sqrt{2}xy^*\right)$. Next, we apply inequality (3) to the gde of the sum, and then the base extraction property (4) of the gde. We use equality $\text{gde}\left(|x|^2 + |y|^2\right) = m$ to conclude that

$$\text{gde}\left(|x+y|^2\right) \geq \min\left(m, 1 + \text{gde}\left(Re\left(\sqrt{2}xy^*\right)\right)\right).$$

Finally, we use inequality (8) for $\text{gde}\left(Re\left(\sqrt{2}xy^*\right)\right)$ to derive the statement of the lemma $\square$.

Now we collected all tools required to prove Lemma 2.

Proof. Recall that, we are proving that for elements z and w of the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ and any integer k it is true that:

$$-1 \leq \text{sde}\left(\left|\frac{z + w\omega^k}{\sqrt{2}}\right|^2\right) - \text{sde}\left(|z|^2\right) \leq 1, \text{ when } \text{sde}\left(|z|^2\right) \geq 4.$$

Using Lemma 4 we can define $m = \text{sde}\,(z) = \text{sde}\left(w\omega^k\right)$ and $x = \omega^k z \left(\sqrt{2}\right)^m$ and $y = w\left(\sqrt{2}\right)^m$. Using the relation (1) between gde and sde, and the base extraction property (4) of gde we rewrite the inequality we are trying to prove as:

$$1 \leq \text{gde}\left(|x+y|^2\right) - \text{gde}\left(|x|^2\right) \leq 3.$$

It follows from Lemma 4 that $\text{gde}\left(|x|^2\right) = \text{gde}\left(|y|^2\right) \leq 1$. Taking into account $|x|^2 + |y|^2 = \sqrt{2}^{2m}$ and applying the inequality proved in Lemma 5 to x and y we conclude that:

$$\text{gde}\left(|x+y|^2\right) \geq \min\left(2m, 1 + \text{gde}\left(|x|^2\right)\right).$$

The condition $m \geq 4$ allows us to remove taking the minimum on the right hand side and replace it with $1 + \text{gde}\left(|x|^2\right)$. This proves one of the two inequalities we are trying to show, $1 \leq \text{gde}\left(|x+y|^2\right) - \text{gde}\left(|x|^2\right)$. To prove the second inequality, $\text{gde}\left(|x+y|^2\right) - \text{gde}\left(|x|^2\right) \leq 3$, we apply Lemma 5 to the pair of elements of the ring $\mathbb{Z}[\omega]$, $x+y$ and $x-y$. The conditions of the lemma are satisfied because $|x+y|^2 + |x-y|^2 = \sqrt{2}^{2(m+1)}$. Therefore:

$$\text{gde}\left(4|x|^2\right) \geq \min\left(2(m+1), 1 + \left\lfloor \frac{1}{2}\left(\text{gde}\left(|x+y|^2\right) + \text{gde}\left(|x-y|^2\right)\right)\right\rfloor\right).$$

Using the base extraction property (4), we notice that $\text{gde}\left(4|x|^2\right) = 4 + \text{gde}\left(|x|^2\right)$. It follows from $m \geq 4$ that $2(m+1) \geq 4 + \text{gde}\left(|x|^2\right)$. As such, we can again remove the minimization and simplify the inequality to:

$$3 + \text{gde}\left(|x|^2\right) \geq \left\lfloor \frac{1}{2}\left(\text{gde}\left(|x+y|^2\right) + \text{gde}\left(|x-y|^2\right)\right)\right\rfloor.$$

To finish the proof it suffices to show that $\mathrm{gde}\left(|x+y|^2\right) = \mathrm{gde}\left(|x-y|^2\right)$. We establish an upper bound for $\mathrm{gde}\left(|x+y|^2\right)$ and use the absorption property (5) of gde. Using non-negativity of gde and the definition of the floor function we get:

$$2\left(3 + \mathrm{gde}\left(|x|^2\right)\right) + 1 \geq \mathrm{gde}\left(|x+y|^2\right).$$

Since $\mathrm{gde}\left(|x|^2\right) \leq 1$, $\mathrm{gde}\left(|x+y|^2\right) \leq 9$. Observing that $2\left(m+1\right) > 9$ we confirm that

$$\mathrm{gde}\left(|x-y|^2\right) = \mathrm{gde}\left(\sqrt{2}^{2(m+1)} - |x+y|^2\right) = \mathrm{gde}\left(|x+y|^2\right)$$

$\square$.

To prove Lemma 3 it suffices to show that $\mathrm{gde}(|x+\omega^k y|^2) - \mathrm{gde}(|x|^2)$ achieves all values in the set $\{1,2,3\}$ as k varies over all values in the range from 0 to 3. We can split this into two cases: $\mathrm{gde}(|x|^2) = 1$ and $\mathrm{gde}(|x|^2) = 0$. We need to check if $\mathrm{gde}(|x+\omega^k y|^2)$ belongs to $\{1,2,3\}$ or $\{2,3,4\}$. Therefore, it is important to describe these conditions in terms of x and y. This is accomplished in the next Section.

5   Quadratic forms and greatest dividing exponent

We first clarify why it is enough to check a finite number of cases to prove Lemma 3. Recall how the lemma can be restated in terms of the elements of the ring $\mathbb{Z}[\omega]$. Next we illustrate why we can achieve a finite number of cases with a simple example using integer numbers $\mathbb{Z}$. Then we show how this idea can be extended to the elements of the ring $\mathbb{Z}[\omega]$ that are real (that is, with imaginary part equal to zero). Finally, in the proof of Lemma 3, we identify a set of cases that we need to check and provide an algorithm to perform it.

As discussed at the end of the previous Section, to prove Lemma 3 one can consider elements x and y of the ring $\mathbb{Z}[\omega]$ such that $|x|^2 + |y|^2 = 2^m$ for $m \geq 4$. We know from Lemma 2 that there are three possibilities in each of the two cases:

- when $\mathrm{gde}(|x|^2) = 0$, $\mathrm{gde}(|x+\omega^k y|^2)$ equals to $1, 2$, or 3,

- when $\mathrm{gde}(|x|^2) = 1$, $\mathrm{gde}(|x+\omega^k y|^2)$ equals $2, 3$, or 4.

We want to show that each of these possibilities is achievable for a specific choice of $k \in \{0,1,2,3\}$.

We illustrate the idea of the reduction to a finite number of cases with an example. Suppose we want to describe two classes of integer numbers:

- integer a such that the $\mathrm{gde}\left(a^2, 2\right) = 2$,

- integer a such that the $\mathrm{gde}\left(a^2, 2\right) > 2$.

It is enough to know $a^2 \bmod 2^3$ to decide which class a belongs to. Therefore we can consider 8 residues $a \bmod 2^3$ and find the classes to which they belong. We extend this idea to the real elements of the ring $\mathbb{Z}[\omega]$, being elements of the ring $\mathbb{Z}[\omega]$ that are equal to their own real part. Afterwards we apply the result to $|x+\omega^k y|^2$, that is a real element of $\mathbb{Z}[\omega]$.

We note that the real elements of $\mathbb{Z}[\omega]$ are of the form $a+\sqrt{2}b$ where a and b are themselves integer numbers. An important preliminary observation, that follows from the irrationality of $\sqrt{2}$, is that for any integer number c

$$\text{gde}(c) = 2\text{gde}(c, 2). \tag{10}$$

The next proposition gives a condition equivalent to $\text{gde}\left(a + \sqrt{2}b\right) = k$, expressed in terms of $\text{gde}(a, 2)$ and $\text{gde}(b, 2)$:

Proposition 1 *Let a and b be integer numbers. There are two possibilities:*

- $\text{gde}\left(a + \sqrt{2}b\right)$ *is even if and only if* $\text{gde}(b, 2) \geq \text{gde}(a, 2)$; *in this case,* $\text{gde}(a, 2) = \text{gde}\left(a + \sqrt{2}b\right)/2$.

- $\text{gde}\left(a + \sqrt{2}b\right)$ *is odd if and only if* $\text{gde}(b, 2) < \text{gde}(a, 2)$; *in this case,* $\text{gde}(b, 2) = \left(\text{gde}\left(a + \sqrt{2}b\right) - 1\right)/2$.

Proof.   Consider the case when $\text{gde}(b, 2) < \text{gde}(a, 2)$. Observing, from equation (10), that $\text{gde}(a)$ is always even, $\text{gde}(a) > \text{gde}\left(\sqrt{2}b\right)$, and by the absorption property (5) of gde we have $\text{gde}\left(a + \sqrt{2}b\right) = \text{gde}\left(\sqrt{2}b\right)$. Using the base extraction property (4) of gde and the relation (10) between $\text{gde}(\cdot)$ and $\text{gde}(\cdot, 2)$ for integers we obtain $\text{gde}\left(a + \sqrt{2}b\right) = 1 + 2\text{gde}(b, 2)$. The other case similarly implies $\text{gde}\left(a + \sqrt{2}b\right) = 2\text{gde}(a, 2)$. In terms of real elements of the ring $\mathbb{Z}[\omega]$, this results in the following relations:

$$A_1 = \{\text{gde}(b, 2) < \text{gde}(a, 2)\} \subseteq B_1 = \left\{\text{gde}\left(a + \sqrt{2}b\right) \text{ is even}\right\},$$

$$A_2 = \{\text{gde}(b, 2) \geq \text{gde}(a, 2)\} \subseteq B_2 = \left\{\text{gde}\left(a + \sqrt{2}b\right) \text{ is odd}\right\}.$$

We note that each pair of sets $\{A_1, A_2\}$ and $\{B_1, B_2\}$ defines a partition of real elements of the ring $\mathbb{Z}[\omega]$. This completes the proof since when for partitions $\{A_1, A_2\}$ and $\{B_1, B_2\}$ of some set the inclusions $A_1 \subseteq B_1, A_2 \subseteq B_2$ imply $A_1 = B_1$ and $A_2 = B_2$ $\square$.

To express $\left|x + \omega^k y\right|^2$ in the form $a + \sqrt{2}b$ in a concise way, we introduce two quadratic forms $P(\cdot)$ and $Q(\cdot)$ with the property:

$$|x|^2 = P(x) + \sqrt{2}Q(x). \tag{11}$$

Given that x, an element of $\mathbb{Z}[\omega]$, can be expressed in terms of the integer number coordinates as follows, $x = x_0 + x_1\omega + x_2\omega^2 + x_3\omega^3$, we define the quadratic forms as:

$$P(x) := x_0^2 + x_1^2 + x_2^2 + x_3^2, \tag{12}$$

$$Q(x) := x_0(x_1 - x_3) + x_2(x_1 + x_3). \tag{13}$$

Let us rewrite equality $\text{gde}\left(|x + y|^2\right) = 4$ in terms of these quadratic forms and the gde of base 2. Using Proposition 1 we can write:

$$\text{gde}\left(P\left(x + \omega^k y\right), 2\right) = 2,$$

$$\text{gde}\left(Q\left(x + \omega^k y\right), 2\right) \geq 2.$$

Similar to the example given at the beginning of this section, we see that it suffices to know the values of the quadratic forms modulo 2^3. To compute them, it suffices to know the values of the integer coefficients of x and y modulo 2^3. This follows from the expression of the product ωy in terms of the integer number coefficients:

$$\omega \left(y_1 + y_2\omega + y_3\omega^2 + y_4\omega^3\right) = -y_4 + y_1\omega + y_2\omega^2 + y_3\omega^3,$$

and from the following two observations:

- integer number coefficients of the sum of two elements of the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ is the sum of their integer number coefficients,

- for any element of $\mathbb{Z}[\omega]$, x, the values of quadratic forms $P(x)$ and $Q(x)$ modulo 2^3 are defined by the values modulo 2^3 of the integer number coefficients of x.

In summary, to check the second part of Lemma 2 we need to consider all possible values for the integer coefficients of x and y modulo 2^3. There are two additional constraints on them. The first one is $|x|^2 + |y|^2 = 2^m$. Since we assumed $m \geq 4$, we can write necessary conditions to satisfy this constraint, in terms of the quadratic forms, as:

$$P(x) \equiv -P(y) \left(mod\, 2^3\right),$$

$$Q(x) \equiv -Q(y) \left(mod\, 2^3\right).$$

The second constraint is $\mathrm{gde}\left(|x|^2\right) = \mathrm{gde}\left(|y|^2\right)$ and $\mathrm{gde}\left(|x|^2\right) \leq 1$. To check it, we use the same approach as in the example with $\mathrm{gde}\left(|x+y|^2\right) = 4$.

We have now introduced the necessary notions required to prove Lemma 3.

Proof.   Our proof is an exhaustive verification, assisted by a computer search. We rewrite the statement of the lemma formally as follows:

$$\mathcal{G}_j = \left\{ \begin{array}{l} (x, y) \in \mathbb{Z}[\omega] \times \mathbb{Z}[\omega] \;\left|\; \begin{array}{l} \exists m \geq 4 \text{ s.t. } |x|^2 + |y|^2 = 2^m, \\ \mathrm{gde}(x) = \mathrm{gde}(y) = j \end{array} \right., j \in \{0, 1\}, \\ \text{for all } (x, y) \in \mathcal{G}_j, \text{for all } s \in \{1, 2, 3\} \text{ there exists } k \in \{0, 1, 2, 3\} \\ \text{such that } \mathrm{gde}\left(|x + \omega^k y|^2\right) = s + j. \end{array} \right\} \tag{14}$$

The sets $\mathcal{G}_j$ are infinite, so it is impossible to perform the check directly. As we illustrated with an example, equality $\mathrm{gde}\left(|x + \omega^k y|^2\right) = s + j$ depends only on the values of the integer coordinates of x and y modulo 2^3. If the sets $\mathcal{G}_j$ were also defined in terms of the residues modulo 2^3 we could just check the lemma in terms of equivalence classes corresponding to different residuals. More precisely, the equivalence relation $\sim$ we would use is:

$$\sum_{p=0}^{3} x_p\omega^p \sim \sum_{p=0}^{3} y_p\omega^p \overset{def}{\iff} \text{ for all } p \in \{0, 1, 2, 3\} : x_p \equiv y_p \left(\mathrm{mod}\, 2^3\right).$$

To address the issue, we introduce sets $\mathcal{Q}_j$ that include $\mathcal{G}_j$ as subsets:

$$\mathcal{Q}_j = \left\{ (x, y) \in \mathbb{Z}[\omega] \times \mathbb{Z}[\omega] \;\left|\; \begin{array}{l} \mathrm{gde}(x) = \mathrm{gde}(y) = j \\ P(x) + P(y) \equiv 0 \left(\mathrm{mod}\, 2^3\right) \\ Q(x) + Q(y) \equiv 0 \left(\mathrm{mod}\, 2^3\right) \end{array} \right.\right\}, j \in \{0, 1\}.$$

Therefore, in terms of the equivalence classes with respect to the above defined relation $\sim$ the more general problem can be verified in a finite number of steps. However, the number of equivalence classes is large. This is why we employ a computer search that performs verification of all cases. To rewrite (14) into conditions in terms of the equivalence classes it suffices to replace $\mathcal{G}_j$ by $\mathcal{Q}_j$, replace x and y by their equivalence classes, and replace $\mathbb{Z}\left[\omega\right]$ by the set of equivalence classes $\mathbb{Z}\left[\omega\right]/\sim$.

Algorithm 2 verifies Lemma 3. We use bar (e.g., $\overline{x}$ and $\overline{y}$) to represent 4-dimensional vectors with entries in $\mathbb{Z}_8$, the ring of residues modulo 8. The definition of bilinear forms, multiplication by ω and the relations $\mathrm{gde}\left(\left|\cdot\right|^2\right) = 1, 2, 3, 4$ extend to $\overline{x}$ and $\overline{y}$. We implemented Algorithm 2 and the result of its execution is *true*. This completes the proof $\square$.

Algorithm 2 Verification of Lemma 3.

Output: Returns *true* if the statement of Lemma 3 is correct; otherwise, returns *false*.

   ▷ Here, $G_{j,a,b}$ is the set of all residue vectors $\overline{x}$ such that $\mathrm{gde}(\overline{x}) = j, P\left(\overline{x}\right) = a, Q\left(\overline{x}\right) = b$.

  for all $x_1, x_2, x_3, x_4 \in \{0, \ldots, 7\}$ **do**   ▷ generate possible residue vectors;

    $\overline{x} \leftarrow (x_1, x_2, x_3, x_4)$

    $j \leftarrow \mathrm{gde}(\left|\overline{x}\right|^2), a \leftarrow P\left(\overline{x}\right), b \leftarrow Q\left(\overline{x}\right)$

    if $j \in \{0, 1\}$ **then**

        add $\overline{x}$ to $G_{j,a,b}$

    end if

  end for

  for all $j \in \{0, 1\}, a_x \in \{0, 7\}, b_x \in \{0, 7\}$ **do**

    $a_y \leftarrow -a_x \bmod 8, b_y \leftarrow -b_x \bmod 8$ ▷ consider only those pairs that

    for all $(\overline{x}, \overline{y}) \in G_{j,a_x,b_x} \times G_{j,a_y,b_y}$ **do**   ▷ satisfy necessary conditions;

      for all $d \in \{1, 2, 3\}$ **do**

        state $\leftarrow$ unfound

        for all $k \in \{0, 1, 2, 3\}$ **do**

            $\overline{t} \leftarrow \overline{x} + \omega^k \overline{y}$

            if $\mathrm{gde}(\left|\overline{t}\right|^2) = \mathrm{d} + \mathrm{j}$ **then**

                state $\leftarrow$ found

            end if

        end for

        if state = unfound **then**

            return *false*

        end if

      end for

    end for

  end for

  return *true*

---

6   Implementation

Our C++ implementation of Algorithm 1 is available online at `http://code.google.com/p/sqct/`.
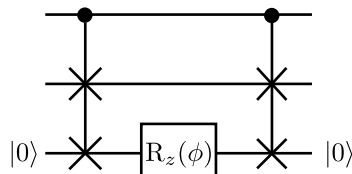
Fig. 2. Circuit implementing the controlled-$R_z(\phi)$ gate. Upper qubit is the control, middle qubit is the target, and bottom qubit is the ancilla.

7   Experimental results

Table 2 summarizes the results of first obtaining an approximation of the given rotation matrix by a unitary over the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ using our implementation of the Solovay-Kitaev algorithm [1, 9], and then decomposing it into a circuit using the exact synthesis Algorithm 1 presented in this paper. We note that the implementation of our synthesis Algorithm 1 (runtimes found in the column t_{decomp}) is significantly faster than the implementation of the Solovay-Kitaev algorithm used to approximate the unitary (runtimes reported in the column t_{approx}). Furthermore, we were able to calculate approximating circuits using 5 to 7 iterations of the Solovay-Kitaev algorithm followed by our synthesis algorithm. The total runtime to approximate and decompose unitaries ranged from approximately 11 to 600 seconds, correspondingly, featuring best approximating errors on the order of 10^{-50}, and circuits with up to millions of gates. Actual specifications of all circuits reported, as well as those synthesized but not explicitly included in the Table 2, due to space constraints, may be obtained from `http://qcirc.iqc.uwaterloo.ca/`.

On each step Algorithm 1 chooses from one of four small circuits: H, HT, HT^2 (=HP), and HT^3 to reduce sde. In practice, Pauli-Z gate is often easier to implement than either Phase or T gate. The cost of $P^\dagger$ and $T^\dagger$ gates is usually the same as that of the respective P and T gates. We took this into account by writing circuit HT^3 using an equivalent and cheaper form $HZT^\dagger$. This significantly reduces the number of Phase gates required to implement a unitary. If there is no preference between the choice of P or Z, or P is preferred to Z, the HPT could be used in place of HT^3.

The RAM memory requirement during unitary approximation stage for our implementation is 2.1GB. In our experiments we used a single core of the Intel Core i7-2600 (3.40GHz) processor.

The experimental results reported may be utilized in the construction of an approximate implementation of the Quantum Fourier Transform (QFT). Figure 2 shows a circuit that employs the technique from [10] to implement the controlled-$R_z(\phi)$ using a single ancillary qubit. Note that $|0\rangle$ is the eigenvector of $R_z(\phi)$ with the eigenvalue 1, thus this construction works correctly; moreover, no phase is introduced. Such controlled rotations are used in the standard implementation of the QFT [3]. The advantage of such a circuit is that it introduces only a small additive constant overhead on the number of gates required to turn an uncontrolled rotation into a controlled rotation. Indeed, the number of T gates (those are more complex in the fault tolerant implementations than the Clifford gates [8]) required for the exact implementation of the Fredkin gate, being 7 [5], is small compared to the number of T

Table 2. Results of the approximation of $R_z(\varphi) = \begin{pmatrix} e^{-i\varphi} & 0 \\ 0 & e^{i\varphi} \end{pmatrix}$ by our implementation.
Column N_I contains the number of iterations used by the Solovay-Kitaev algorithm, n_g—total number of gates (sum of the next four columns), n_T—number of T and T† gates, n_H—number of Hadamard gates, n_P—number of P and P† gates, n_{Pl}—number of Pauli gates (note that the combined number of Pauli-X and Pauli-Y is never more than three for any of the circuits, so n_{Pl} is dominated by Pauli-Z gates), $dist$—trace distance to approximation, t_{approx}—time spent on the unitary approximation using the Solovay-Kitaev algorithm (in seconds), t_{decomp}—time spent on the decomposition of the approximating unitary into circuit, per Algorithm 1 (in seconds). Circuit specifications are available at `http://qcirc.iqc.uwaterloo.ca/`.

U	N_I	n_g	n_T	n_H	n_P	n_{Pl}	$dist$	t_{approx}	t_{decomp}
$R_z\left(\frac{\pi}{16}\right)$	0	74	28	27	2	17	1.34296×10^{-3}	0.09180	0.00022
	1	342	132	132	1	77	4.61204×10^{-5}	0.12238	0.00079
	2	1683	670	670	1	342	5.68176×10^{-7}	0.22369	0.00391
	3	8197	3284	3283	0	1630	2.97644×10^{-10}	0.83023	0.02077
	4	35819	14312	14311	3	7193	3.64068×10^{-15}	2.91120	0.12453
$R_z\left(\frac{\pi}{32}\right)$	0	64	24	23	2	15	3.92540×10^{-4}	0.01805	0.00021
	1	314	124	123	2	65	1.34267×10^{-5}	0.05238	0.00074
	2	1388	556	556	0	276	4.65743×10^{-7}	0.31252	0.00321
	3	7493	3000	2999	1	1493	1.10252×10^{-10}	0.90493	0.01950
	4	35113	14054	14053	2	7004	2.69806×10^{-15}	2.96710	0.12122
$R_z\left(\frac{\pi}{64}\right)$	0	54	22	23	2	7	8.05585×10^{-4}	0.08827	0.00020
	1	344	136	136	3	69	9.57729×10^{-6}	0.12163	0.00080
	2	1414	564	564	3	283	1.97877×10^{-7}	0.38928	0.00326
	3	7769	3086	3087	3	1593	1.08884×10^{-10}	0.98522	0.01954
	4	35456	14170	14171	2	7113	3.00267×10^{-15}	3.05440	0.12384
$R_z\left(\frac{\pi}{128}\right)$	0	72	28	29	1	14	9.59916×10^{-4}	0.08822	0.00023
	1	344	136	137	3	68	1.79353×10^{-5}	0.12143	0.00081
	2	1588	634	634	4	316	3.67734×10^{-7}	0.39048	0.00368
	3	7519	3004	3005	2	1508	4.23657×10^{-10}	0.98045	0.01890
	4	34388	13722	13722	2	6942	1.32046×10^{-14}	2.86740	0.11832
$R_z\left(\frac{\pi}{256}\right)$	0	71	28	29	2	12	5.06207×10^{-4}	0.01819	0.00023
	1	326	136	136	2	52	1.08919×10^{-5}	0.05474	0.00079
	2	1389	566	567	3	253	2.00138×10^{-7}	0.30498	0.00332
	3	7900	3174	3175	3	1548	2.91716×10^{-10}	0.91405	0.02060
	4	38188	15290	15291	1	7606	8.87785×10^{-15}	2.98030	0.13545
$R_z\left(\frac{\pi}{512}\right)$	0	76	30	29	2	15	3.62591×10^{-4}	0.01749	0.00023
	1	319	126	126	2	65	1.95491×10^{-5}	0.05171	0.00075
	2	1722	680	680	2	360	2.76529×10^{-7}	0.30618	0.00396
	3	8122	3242	3242	2	1636	1.87476×10^{-10}	0.92576	0.02109
	4	34974	13992	13992	1	6989	5.66762×10^{-15}	3.16060	0.11920
$R_z\left(\frac{\pi}{1024}\right)$	0	0	0	0	0	0	2.16938×10^{-3}	0.08622	0.00005
	1	264	106	105	2	51	5.57373×10^{-5}	0.13615	0.00063
	2	1541	622	622	3	294	1.74595×10^{-7}	0.23445	0.00366
	3	6791	2722	2722	1	1346	5.39912×10^{-11}	0.82811	0.01703
	4	32983	13188	13188	1	6606	5.54995×10^{-16}	2.98480	0.11494

gates required (the T-counts in the circuits we synthesize are provably minimal for the unitary being implemented) in the approximations of the individual single-qubit rotations, Table 2. In comparison to other approaches to constructing a controlled gate out of an uncontrolled gate, such as replacing each gate in the circuit by its controlled version, or using the decomposition provided by Lemma 5.1 in [2] (achieving a two-qubit controlled gate using three single-qubit uncontrolled gates, that has the expected effect of roughly tripling the number of T gates), the proposed approach where a single ancilla is employed appears to be beneficial.

We approximate the controlled-$R_z(\phi)$ by replacing $R_z(\phi)$ with its approximation $R'_z(\phi)$. To evaluate the quality of such approximation we need to take into account that ancillary qubit is always initialized to $|0\rangle$ and that the controlled rotation is a part of a larger circuit. For this reason, we computed the completely bounded trace norm [11] of the difference of the channels corresponding to the controlled-$\Psi_{R_z(\phi)}$ and its approximation using $\Psi_{R'_z(\phi)}$. Both channels map the space of two qubit density matrices into the space of three qubit density matrices, as one of the inputs is fixed.

To compute the completely bounded trace norm we used the semidefinite program (SDP) described in [11]. Usual 64-bit machine precision was not enough for our purposes, so we used package SDPA-GMP [12] that employs The GNU Multiple Precision Arithmetic Library to solve SDP. Also, Mathematica was used to generate files with the description of SDP problems in the input format of SDPA-GMP. We found that for all unitaries in Table 2 the ratio of completely bounded trace norm of $\Psi_{R_z(\phi)} - \Psi_{R'_z(\phi)}$ and trace distance between $R_z(\frac{\pi}{2^n})$ and its approximation belongs to the interval $[2.82842, 2.98741]$. In other words, the numerical value of the approximation error for the controlled rotations using the trace norm is roughly three times that for the corresponding single-qubit rotations using the trace distance (as per Table 2).

We also performed a comparison to Dawson's implementation of the Solovay-Kitaev algorithm (see Figure 3) available at `http://gitorious.org/quantum-compiler/`. We ran Dawson's code using the gate library {H, T} with the maximal sequence length equal to 22 and tile width equal to 0.14. During this experiment, the memory usage was around 6 GB. For the purpose of the comparison gate counts for our implementation are also provided in the {H, T} library ($Z=T^4$ and $P=T^2$). We used projective trace distance to measure quality of approximation as it is the one used in Dawson's code. Because of the larger epsilon net used in our implementation we were able to achieve better approximation quality using fewer iterations of the Solovay-Kitaev algorithm. Usage of The GNU Multiple Precision Arithmetic Library allowed us to achieve precision up to 10^{-50} while Dawson's code encounters convergence problem when precision reaches 10^{-8}. The latter explains behaviour of the last set of points in the experimental results for Dawson's code reported in Figure 3.

Two other experiments that we performed with Dawson's code are a resynthesis of the circuits generated by it using our exact decomposition algorithm. We first resynthesised circuits that were generated by Dawson's code using the {H,T} library. In most cases, the gate counts reduced by about 10-20% (our resulting circuits were further decomposed such as to use the {H, T} gate library). In the other experiment we used {H, T, P, Z} gate library with Dawson's implementation. In this case, we were able to run Dawson's code with the sequences of length 9 only, and it used 6 GB of memory. The gate counts, using our algorithm, decreased by about 40-60%.
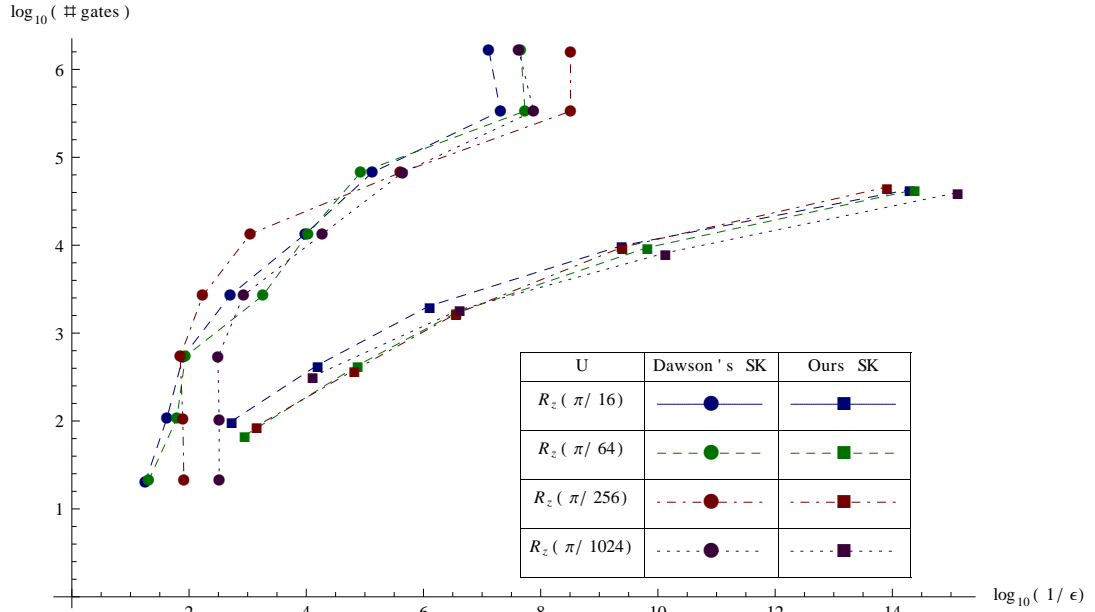
Fig. 3. Comparison between ours and Dawson's implementations of the Solovay-Kitaev algorithm. Vertical axis shows $\log_{10}$ of the number of gates and horizontal axis shows $\log_{10}(1/\epsilon)$, where ϵ is the projective trace distance between unitary and its approximation.

8   Conclusion

In this paper, we studied quantum circuits over the Clifford and T library. We proved that in the single-qubit case the set of unitaries over the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ is equivalent to the set of unitaries computable by the Clifford and T circuits. We generalized this statement to conjecture that in the n-qubit case the sets of unitaries computable by the Clifford and T circuits and those over the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ are equivalent, as long as a single ancillary qubit residing in the state $|0\rangle$ is supplied. While we did not prove this conjecture, we showed the necessity of ancilla.

We have also presented a single-qubit synthesis algorithm that uses Pauli, H, P, and T gates. Our algorithm is asymptotically optimal in both its performance guarantee, and its complexity. The algorithm generates circuits with a provably minimal number of Hadamard and T gates. Furthermore, our experiments suggest that the P-counts may also be minimal. The total number of times Pauli-X and Pauli-Y gates are used in any given circuit generated by the algorithm is limited to at most three. As such, our algorithm is likely optimal (up to, possibly, a small additive constant on some of the gate counts, that, in turn, may be corrected by re-synthesizing the lookup table using a different/suitable circuit cost metric) in all parameters, except, possibly, Pauli-Z count.

Acknowledgements

References

1. C. Dawson, and M. Nielsen. *The Solovay-Kitaev algorithm.* Quantum Information and Computation **6**:81–95, 2006, quant-ph/0505030.
2. 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.* Physical Review A 52, 34573467, 1995, quant-ph/9503016.
3. M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.
4. H. K. Cummins, G. Llewellyn, and J. A. Jones. *Tackling Systematic Errors in Quantum Logic Gates with Composite Rotations.* Physical Review A 67, 042308, 2003, quant-ph/0208092.
5. M. Amy, D. Maslov, M. Mosca, and M. Roetteler. *A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits.* IEEE Trans. CAD, 2013, arXiv:1206.0758.
6. A. Bocharov and K. M. Svore. *A depth-optimal canonical form for single-qubit quantum circuits.* Physical Review Letters 109, 190501, 2012, arXiv:1206.3223.
7. K. Matsumoto and K. Amano. *Representation of Quantum Circuits with Clifford and $\pi/8$ Gates.* 2008, arXiv:0806.3834.
8. A. G. Fowler. *Towards Large-Scale Quantum Computation.* Ph.D. Thesis, University of Melbourne, 2005, quant-ph/0506126.
9. A. Kitaev, A. Shen, and M. Vyalyi. *Classical and Quantum Computation.* American Mathematical Society, Providence, RI, 2002.
10. A. Kitaev. *Quantum measurements and the Abelian Stabilizer Problem.* 1995, quantph/9511026.
11. J. Watrous. *Semidefinite programs for completely bounded norms.* 2009, arXiv:0901.4709.
12. M. Nakata. *A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver:SDPA-GMP, -QD and -DD.* the proceedings of 2010 IEEE Multi-Conference on Systems and Control, 29-34, 2010.
13. A. G. Fowler. *Constructing Arbitrary Steane Code Single Logical Qubit Fault-tolerant Gates.* Quantum Information and Computation **11**:867–873, 2011, quant-ph/0411206.

Appendix A

Here we prove properties of the greatest dividing exponent that was defined and used in Section 4. We first discuss the base extraction property (4) of gde and then proceed to the proof of special properties of $\text{gde}\left(\cdot, \sqrt{2}\right)$. The base extraction property simplifies proofs of all

statements related to gde $\left(\cdot, \sqrt{2}\right)$.

Proposition 2 (Base extraction property) *If $x, y \in \mathbb{Z}[\omega]$, then for any non negative integer number k*

$$\text{gde}\left(yx^k, x\right) = k + \text{gde}\left(y, x\right).$$

Proof. Follows directly from the definition of gde $\ \Box$.

The base extraction property together with non-negativity of gde provide a simple formula to lower bound the value of gde: if x^k divides y then $\text{gde}\left(y, x\right) \geq k$. Inequality for gde of a sum (3) follows directly from this—$x^{\min\left(\text{gde}(y,x),\text{gde}\left(y',x\right)\right)}$ divides $y + y'$. The proof of absorption property (5) follows easily, as well.

Now we prove properties of gde specific to base $\sqrt{2}$. Instead of proving them for all elements of $\mathbb{Z}[\omega]$ it suffices to prove them for elements of $\mathbb{Z}[\omega]$ that are not divisible by $\sqrt{2}$. We illustrate this with an example $\text{gde}\left(x, \sqrt{2}\right) = \text{gde}\left(|x|^2, 2\right)$. We can always write $x = x'\left(\sqrt{2}\right)^{\text{gde}(x)}$. By the definition of gde, $\sqrt{2}$ does not divide x'. By substituting the expression for x into $\text{gde}\left(|x|^2, 2\right)$ and then using the base extraction property we get:

$$\text{gde}\left(|x|^2, 2\right) = \text{gde}\left(|x'|^2, 2\right) + \text{gde}\left(x, \sqrt{2}\right).$$

Therefore, it suffices to show that $\text{gde}\left(|x'|^2, 2\right) = 0$ when $\sqrt{2}$ does not divide x', or, equivalently, when $\text{gde}\left(x'\right) = 0$.

The quadratic forms defined in Section 5 will be a useful tool for later proofs. Bilinear forms that generalize them are important for the proof of relation for $\text{gde}\left(\text{Re}\left(xy^*\right)\right)$. Effectively, we only need the values of mentioned forms modulo 2. For this reason, we also introduce forms that are equivalent modulo 2 and more convenient for the proofs.

We define function $F\left(\cdot, \cdot\right)$ for $x, y \in \mathbb{Z}[\omega]$ as follows:

$$F\left(x, y\right) := x_0 y_0 + x_1 y_1 + x_2 y_2 + x_3 y_3.$$

Note that the following equality holds, and provides some intuition behind the choice to introduce $F\left(\cdot, \cdot\right)$:

$$\text{Re}\left(xy^*\right) = F\left(x, y\right) + \frac{1}{\sqrt{2}} F\left(\sqrt{2}x, y\right).$$

Using formula $\sqrt{2} = \omega - \omega^3$ we can rewrite multiplication by $\sqrt{2}$ as a linear operator:

$$\sqrt{2}x = \sqrt{2}(x) : x_0 + x_1\omega + x_2\omega^2 + x_3\omega^3 \mapsto (x_1 - x_3) + (x_0 + x_2)\omega + (x_1 + x_3)\omega^2 + (x_2 - x_0)\omega^3. \tag{A.1}$$

In particular, it is easy to verify that:

$$F\left(\sqrt{2}x, y\right) = (x_1 - x_3)y_0 + (x_0 + x_2)y_1 + (x_1 + x_3)y_2 + (x_2 - x_0)y_3,$$

and, substituting $y = x$,

$$F\left(\sqrt{2}x, x\right) = 2(x_1 - x_3)x_2 + 2(x_1 + x_3)x_0 = 2Q\left(x\right),$$

which corresponds to the earlier definition shown in equation (13). The definition of $F(\cdot,\cdot)$ written for $x=y$ results in an earlier definition (12). This shows how $F(\cdot,\cdot)$ generalizes and ties together previously introduced $P(\cdot)$ and $Q(\cdot)$.

Furthermore, in modulo 2 arithmetic the following expressions hold true:

$$P(x) \equiv (x_1 + x_3) + (x_0 + x_2) \,(\mathrm{mod}\,2) \tag{A.2}$$

$$Q(x) \equiv (x_1 + x_3)(x_0 + x_2) \,(\mathrm{mod}\,2) \tag{A.3}$$

$$F\left(\sqrt{2}x, y\right) \equiv (x_1 + x_3)(y_0 + y_2) + (x_0 + x_2)(y_1 + y_3) \,(\mathrm{mod}\,2). \tag{A.4}$$

It is easy to verify these equations by expanding the left and right hand sides.

The next proposition shows how we use equivalent quadratic and bilinear forms.

Proposition 3 *If* $\mathrm{gde}(x) = 0$ *there are only two alternatives:*

- $P(x)$ *is even and* $Q(x)$ *is odd,*

- $P(x)$ *is odd and* $Q(x)$ *is even.*

Proof. The equality $\mathrm{gde}(x) = 0$ implies that 2 does not divide $\sqrt{2}x$. Using expression (A.1) for $\sqrt{2}x$ in terms of integer coefficients we conclude that at least one of the four numbers $x_1' \pm x_3'$, $x_0' \pm x_2'$ must be odd. Suppose that $x_1' + x_3'$ odd. Using formulas (A.2,A.3) we conclude that the values of $P(x)$ and $Q(x)$ must have different parity. The remaining three cases are similar $\square$.

An immediate corollary is: $\mathrm{gde}(x) = 0$ implies $\mathrm{gde}\left(|x|^2, 2\right) = 0$. To show this it suffices to use expression (11) for $|x|^2$ in terms of quadratic forms.

We can also conclude that $\sqrt{2}$ divides x if and only if 2 divides $|x|^2$. Sufficiency follows from the definition of gde. To prove that 2 divides $|x|^2$ implies $\sqrt{2}$ divides x, we assume that 2 divides $|x|^2$ and $\sqrt{2}$ does not divide x, which leads to a contradiction. This also results in the inequality $\mathrm{gde}\left(|x|^2\right) \le 1$ when $\mathrm{gde}(x) = 0$.

We use the next two propositions to prove the inequality for $\mathrm{Re}\left(\sqrt{2}xy^*\right)$.

Proposition 4 *Let* $\mathrm{gde}(x) = 0$:

- *if* $\sqrt{2}$ *divides* $|x|^2$ *then* $P(x)$ *is even and* $Q(x)$ *is odd,*

- *if* $\sqrt{2}$ *does not divide* $|x|^2$ *then* $P(x)$ *is odd and* $Q(x)$ *is even.*

Proof. As discussed, the previous proposition implies that $\sqrt{2}$ divides y if and only if 2 divides $|y|^2$. We apply this to $|x|^2$. By expressing $|x|^4$ in terms of quadratic forms we get:

$$|x|^4 = P(x)^2 + 2Q(x)^2 + 2\sqrt{2}P(x)Q(x).$$

We see that 2 divides $|x|^4$ if and only if 2 divides $P(x)^2$, or, equivalently, $\sqrt{2}$ divides $|x|^2$ if and only if $P(x)$ even. Using the previous proposition again, this time for x, we obtain the required result $\square$.

Proposition 5 *Let* $\mathrm{gde}(x) = 0$ *and* $\mathrm{gde}(y) = 0$. *If* $\sqrt{2}$ *divides* $|x|^2$ *and* $\sqrt{2}$ *divides* $|y|^2$ *then* $\sqrt{2}$ *divides* $\mathrm{Re}\left(\sqrt{2}xy^*\right)$.

Proof. By the previous proposition, $\sqrt{2}$ divides $|x|^2$ and $\sqrt{2}$ divides $|y|^2$ implies that $Q(x)$ and $Q(y)$ are odd. Formula (A.3) implies that in terms of the integer number coefficients of x and y integer numbers $x_1 + x_3$, $x_0 + x_2$, $y_1 + y_3$, $y_0 + y_2$, are all odd. Expressing $\mathrm{Re}\left(\sqrt{2}xy^*\right)$ in terms of $F(\cdot,\cdot)$,

$$\mathrm{Re}\left(\sqrt{2}xy^*\right) = \sqrt{2}F(x,y) + F\left(\sqrt{2}x,y\right),$$

and using expression (A.4), we conclude that 2 divides $F\left(\sqrt{2}x,y\right)$; therefore $\sqrt{2}$ divides $\mathrm{Re}\left(\sqrt{2}xy^*\right)$ $\square$.

Now we show $\mathrm{gde}\left(\mathrm{Re}\left(\sqrt{2}xy^*\right)\right) \geq \left\lfloor \frac{1}{2}\left(\mathrm{gde}\left(|x|^2\right) + \mathrm{gde}\left(|y|^2\right)\right)\right\rfloor$. As we discussed in the beginning, we can assume $\mathrm{gde}(x) = 0$ and $\mathrm{gde}(y) = 0$ without loss of generality. This implies $\mathrm{gde}\left(|x|^2\right) \leq 1$ and $\mathrm{gde}\left(|y|^2\right) \leq 1$. The expression $\left\lfloor \frac{1}{2}\left(\mathrm{gde}\left(|x|^2\right) + \mathrm{gde}\left(|y|^2\right)\right)\right\rfloor$ can only be equal to 0 or 1. The second one is only possible when $\mathrm{gde}\left(|x|^2\right) = 1$ and $\mathrm{gde}\left(|y|^2\right) = 1$, in which case the previous proposition implies $\mathrm{gde}\left(\mathrm{Re}\left(\sqrt{2}xy^*\right)\right) \geq 1$. In the first case inequality is true because of the non-negativity of gde.

We can also use quadratic forms to describe all numbers z in the ring $\mathbb{Z}[\frac{1}{\sqrt{2}},i]$ such that $|z|^2 = 1$. Seeking a contradiction, suppose $\mathrm{sde}(z) \geq 1$. We can always write $z = \frac{x}{(\sqrt{2})^k}$ where $k = \mathrm{sde}(z)$ and $\mathrm{gde}(x) = 0$. From the other side $|x|^2 = P(x) + \sqrt{2}Q(x) = 2^k$. Thus we have a contradiction with the statement of Proposition 3. We conclude that z is an element of $\mathbb{Z}[\omega]$. Therefore we can write z in terms of its integer number coordinates, $z = z_0 + z_1\omega + z_2\omega^2 + z_3\omega^3$. Equality $|z|^2 = 1$ implies that $F(z,z) = z_0^2 + z_1^2 + z_2^2 + z_3^2 = 1$. Taking into account that z_j are integer numbers we conclude that $z \in \left\{\omega^k, k = 0,1,\ldots,7\right\}$.

Appendix B

Here we prove that Algorithm 1 produces circuits with the minimal number of Hadamard and T gates over the gate library $\mathcal{G}$ consisting of Hadamard, T, $\mathrm{T}^\dagger$, P, $\mathrm{P}^\dagger$, and Pauli-X, Y, and Z gates. We say that a circuit implements a unitary U if the unitary corresponding to the circuit is equal to U up to global phase. We define integer-valued quantities $h(U)$ and $t(U)$ as the minimal number of Hadamard and T gates over all circuits implementing U. We call a circuit H- or T-optimal if it contains the minimal number of H or T gates, correspondingly.

Theorem 2 *Let U be a 2×2 unitary over the ring $\mathbb{Z}[\frac{1}{\sqrt{2}},i]$ with a matrix entry z such that $\mathrm{sde}(|z|^2) \geq 4$. Algorithm 1 produces a circuit that implements U over $\mathcal{G}$ with:*

1. *the minimal number of Hadamard gates and $h(U) = \mathrm{sde}(|z|^2) - 1$, and*

2. *the minimal number of T gates and $t(U) = h(U) - 1 + (l \mod 2) + (j \mod 2)$, where l and j are chosen such that $h(HT^lUT^jH) = h(U) + 2$.*

Proof.
1: H-optimality. Using brute force, we explicitly verified that the set of H-optimal circuits with precisely 3 Hadamard gates is equal to the set of all unitaries over the ring $\mathbb{Z}[\frac{1}{\sqrt{2}},i]$ with $\mathrm{sde}(|z|^2) = 4$. Suppose we have a unitary U with $\mathrm{sde}(|z|^2) = n \geq 4$. With the help of Algorithm 1 we can reduce it to a unitary with $\mathrm{sde}(|z|^2) = 4$ while using $n - 4$ Hadamard

gates to accomplish this. As such, there exists a circuit with $n-1$ Hadamard gates that implements U.

Now consider an H-optimal circuit C that implements U. Using brute force, we established that if C has less than 3 Hadamard gates, then $\mathrm{sde}(|z|^2)$ is less than 4. Suppose C contains $m \geq 3$ Hadamard gates. Its prefix, containing 3 Hadamard gates, must also be H-optimal, and therefore $\mathrm{sde}(|z|^2)$ of the corresponding unitary is 4. Now, using the inequality from Lemma 2, we conclude that $\mathrm{sde}(|z|^2)$ of the unitary corresponding to C is less than $m+1$. This implies $n \leq m+1$. Since we already know that $m \leq n-1$, we may conclude that $m = n-1$ and m is the number of Hadamard gates in the circuit produced by Algorithm 1 in combination with the brute force step.

2: T-optimality. To prove T-optimality we introduce a normal form for circuits over $\mathcal{G}$. We call a circuit HT-normal if there is precisely one T gate between every two H gates and, symmetrically, precisely one H gate between every two T gates. It is not difficult to modify Algorithm 1 to produce a circuit in HT-normal form while preserving its H-optimality. To accomplish that, first, recall that $\mathrm{HT}^3 = \mathrm{HZT}^\dagger$ and that all circuits generated during the brute force stage are both H-optimal and in the HT-normal form. Second, any circuit produced by the algorithm is H-optimal and does not contain a non-H-optimal (up to global phase) subcircuit $\mathrm{HT}^2\mathrm{H} = \mathrm{HPH} = \omega\mathrm{PHP}$.

We will show that any H-optimal circuit in HT-normal form is also T-optimal. We start with a special case of HT-normal circuits—those that begin and end with the Hadamard gate, in other words, those that can be written as $\mathrm{HS}_1\mathrm{H}\ldots\mathrm{HS}_k\mathrm{H}$, and are H-optimal. Let U be a unitary corresponding to this circuit. Due to HT-normality, each S_i contains exactly one T gate, the number of T gates in the circuit is k, and $h(U) = k+1$; therefore, $t(U) \leq h(U) - 1$. To prove that $t(U) = h(U) - 1$, it suffices to show that $t(U) \geq h(U) - 1$. Let us write a T-optimal circuit for U as $\mathrm{C}_0\mathrm{TC}_1\mathrm{T}\ldots\mathrm{TC}_k$. Each subcircuit C_k implements a unitary from the Clifford group. Each unitary from the single-qubit Clifford group can be implemented using at most one H gate (recall, that we are concerned with the implementations up to global phase), therefore $h(U) \leq t(U) + 1$, as required.

In the general case, consider a circuit obtained by Algorithm 1 and implementing a unitary V with $h(V) \geq 3$ that is H-optimal and written in HT-normal form and show that it is T-optimal. We can write it as $\mathrm{S}_0\mathrm{HS}_1\mathrm{H}\ldots\mathrm{HS}_k\mathrm{HS}_{k+1}$. By Lemma 3 we can always find such l and j that $C := \mathrm{HT}^l\mathrm{S}_0\mathrm{HS}_1\mathrm{H}\ldots\mathrm{HS}_k\mathrm{H}\,\mathrm{S}_{k+1}\mathrm{T}^j\mathrm{H}$ is also an H-optimal circuit. Indeed, according to Lemma 3, using the connection between $\mathrm{sde}(\cdot)$ and $h(\cdot)$ described in the first part of the proof, given $h(V) = k+1$ we can always find l such that $h(\mathrm{HT}^l V) = k+2$. From the other side, circuit $\mathrm{HT}^l\mathrm{S}_0\mathrm{HS}_1\mathrm{H}\ldots\mathrm{HS}_k\mathrm{HS}_{k+1}$ contains $k+2$ Hadamard gates and therefore is H-optimal. We repeat the same procedure to find j.

Considering the different possible values of l and j allows to complete the proof of the Theorem. This is somewhat tedious, and we illustrate how to handle different cases with a representative example of $l = 3$ and $j = 2$. In such a case, we can rewrite circuit C as $C' = \mathrm{HTPS}_0\mathrm{HS}_1\mathrm{H}\ldots\mathrm{HS}_k\mathrm{HS}_{k+1}\mathrm{PH}$. We conclude that S_0 must have zero T gates and S_{k+1} must have one T gate. Otherwise subcircuits $\mathrm{HTPS}_0\mathrm{H}$ and $\mathrm{HS}_{k+1}\mathrm{PH}$ will not be H-optimal. As such, we reduced the problem to the special case considered above, therefore circuit C' is T-optimal and $\mathrm{S}_0\mathrm{HS}_1\mathrm{H}\ldots\mathrm{HS}_k\mathrm{HS}_{k+1}$ is T-optimal as its subcircuit. In the general case, the following formula may be developed $t(V) = h(U) - 1 + (l \mod 2) + (j \mod 2)$ $\square$.