Asymptotically Optimal Topological Quantum Compiling

Vadym Kliuchnikov,[1] Alex Bocharov,[2] and Krysta M. Svore[2]
[1]Institute for Quantum Computing and David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada
[2]Quantum Architectures and Computation Group, Microsoft Research, Redmond, Washington 98052, USA
(Received 27 December 2013; published 9 April 2014)

We address the problem of compiling quantum operations into braid representations for non-Abelian quasiparticles described by the Fibonacci anyon model. We classify the single-qubit unitaries that can be represented exactly by Fibonacci anyon braids and use the classification to develop a probabilistically polynomial algorithm that approximates any given single-qubit unitary to a desired precision by an asymptotically depth-optimal braid pattern. We extend our algorithm in two directions: to produce braids that allow only single-strand movement, called weaves, and to produce depth-optimal approximations of two-qubit gates. Our compiled braid patterns have depths that are 20 to 1000 times shorter than those output by prior state-of-the-art methods, for precisions ranging between 10^{-10} and 10^{-30}.

DOI: 10.1103/PhysRevLett.112.140504 PACS numbers: 03.67.Mn

Introduction.—In a topological quantum computer, quantum information is natively protected from small local errors and universality can be achieved by braiding quasiparticles. If two quasiparticles are kept sufficiently far apart and their worldlines in $2+1$-dimensional space-time are braided adiabatically, a unitary evolution can be realized. One class of non-Abelian anyons, called Fibonacci anyons, are predicted to exist in systems in a state corresponding to the fractional quantum Hall plateau at filling fraction $\mu = 12/5$ [1,2]. It has been shown that Fibonacci anyon braids realize universal quantum computation [3,4]. Such topological systems are promising for realizing a topological quantum computer, and have the significant advantage of being intrinsically fault tolerant, reducing the need for resource-intensive quantum error correction.

The problem of approximating arbitrary unitary operations with Fibonacci braid patterns of optimal depth is essential for topological quantum computing. Previous work [5,6] has developed methods using the Solovay-Kitaev algorithm [7] for approximating a given single-qubit unitary to precision ε by a Fibonacci anyon braid pattern with depth $O(\log^c(1/\varepsilon))$, where $c \sim 3.97$ in time $t \sim \log^{2.71}(1/\varepsilon)$. For coarse precisions, one can also use brute-force search to find a braid with minimal depth $O(\log(1/\varepsilon))$ in exponential time [6]. Since the number of braids grows exponentially with the depth of the braid, this technique is infeasible for long braids required to achieve fine-grain precisions.

In this Letter, we address compilation of single- and two-qubit quantum operations into braid representations for non-Abelian quasiparticles described by the Fibonacci anyon model. We apply algebraic number theory to synthesize both exact and approximate representations of unitaries by anyon braids. We construct algorithms for such synthesis that have probabilistically polynomial runtime. We then extend our techniques to a class of braids called *weaves*, where only one of three anyons moves to carry out the braid pattern. Combined with methods in [6], this extension gives a way to construct a high-quality two-qubit gate. Finally, we use our algorithm to approximate two-qubit gates and show that in both the single- and two-qubit case, our algorithm outputs an asymptotically depth-optimal braid pattern. Furthermore, the runtime of the algorithm for finding approximations of two qubit gates is also polynomial in $\log(1/\varepsilon)$ on average. Our results significantly reduce the overhead caused by compiling quantum algorithms into braid patterns for Fibonacci anyons.

Background.—There are different ways of encoding qubits and gates with Fibonacci anyons (cf. [8,9]). In this work, we focus on the three-particle encoding of a qubit [6], where the computational basis state $|0\rangle$ corresponds to the first two anyons having topological charge zero, and state $|1\rangle$ corresponds to the first two anyons having topological charge one. The topological charge of all three anyons in both cases is one. Measurement of the topological charge of the first two anyons is a projective measurement in the computational basis. Unitary operations are realized by moving anyons around each other, where the result depends only on topological properties of the anyon worldlines.

Any braid of the three particles can be represented in terms of the two generators σ_1, σ_2 of the three-strand braid group. We use σ_1, σ_2 to denote the corresponding unitary operations, where

$$\sigma_1 := \omega^6 \begin{pmatrix} 1 & 0 \\ 0 & \omega^7 \end{pmatrix}, \qquad \omega := e^{i\pi/5}$$

and σ_2 is expressed using the "fusion" matrix $\mathcal{F}$ (that describes a change of computational basis):

$$\mathcal{F} := \begin{pmatrix} \tau & \sqrt{\tau} \\ \sqrt{\tau} & -\tau \end{pmatrix}, \qquad \tau = (\sqrt{5}-1)/2, \qquad \sigma_2 := \mathcal{F}\sigma_1\mathcal{F}.$$

It has been shown [3,4] that unitaries σ_1, σ_2 are approximately universal, that is, for any single-qubit unitary $U \in U(2)$ and given precision ε there is a circuit consisting of σ_1 and σ_2 gates which approximates U within precision ε. Results [10,11] imply that it is always possible to find a circuit of size $O(\log(1/\varepsilon))$ that achieves approximation precision ε. However, no efficient algorithm for producing such a circuit was known. Here we introduce a probabilistically polynomial-time algorithm to approximate U within precision ε with a circuit over the gate set σ_1, σ_2, σ_1^{-1}, and σ_2^{-1} that has depth at most $O(\log(1/\varepsilon))$. We refer to the four basis gates as σ gates and to the circuit as a $\langle \sigma_1, \sigma_2 \rangle$ circuit. The circuit length (or equivalently depth) corresponds to the number of anyon moves needed to perform the given unitary.

We reduce the problem to approximating two types of unitaries: rotations around the Z axis

$$R_z(\phi) := \begin{pmatrix} e^{-i\phi/2} & 0 \\ 0 & e^{i\phi/2} \end{pmatrix},$$

and $R_z(\phi)X$, where X is the Pauli X gate. The reduction follows from the fact that any unitary that is not equal to $R_z(\phi)X$ can be represented as $R_z(\alpha)\mathcal{F}R_z(\beta)\mathcal{F}R_z(\gamma)$ up to a global phase (see Supplemental Material [12] for the proof).

We use a global phase-invariant distance to measure the approximation precision

$$d(U,V) := \sqrt{1 - |tr(UV^\dagger)|/2}.$$

Overview of the algorithm.—The compilation algorithm takes as input an arbitrary single-qubit unitary operation and a desired precision ε. The first step approximates the unitary with a special unitary gate, called an *exact* unitary, that can be represented exactly ($\varepsilon = 0$) by a Fibonacci anyon braid pattern. Unitaries of the following form are defined to be *exact*:

$$U[u,v,k] := \begin{pmatrix} u & v^*\sqrt{\tau}\,\omega^k \\ v\sqrt{\tau} & -u^*\omega^k \end{pmatrix}, \tag{1}$$

The numbers u and v must come from the ring of cyclotomic integers:

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

The second step applies the exact synthesis algorithm to the exact unitary gate in order to synthesize an $\langle \mathcal{F}, \mathcal{T} \rangle$-circuit. Finally, the circuit is translated into a Fibonacci anyon braid pattern and compressed using peephole optimization [13].

Exact synthesis.—We begin by describing the second step: an efficient algorithm for synthesizing a precise $\langle \mathcal{F}, \mathcal{T} \rangle$ circuit for a given *exact* unitary, where

$$\mathcal{F} = \begin{pmatrix} \tau & \sqrt{\tau} \\ \sqrt{\tau} & -\tau \end{pmatrix}, \qquad \mathcal{T} = \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix}.$$

Any $\langle \mathcal{F}, \mathcal{T} \rangle$ circuit can be expressed as a $\langle \sigma_1, \sigma_2 \rangle$ circuit using the relations $\mathcal{T} = (\omega Id)^2(\sigma_1)^3$ and $\mathcal{F} = (\omega Id)^4\sigma_1\sigma_2\sigma_1$. A $\langle \sigma_1, \sigma_2 \rangle$ circuit is more natural when considering physical implementations: σ_1 represents the first of the three quasiparticles crossing over the second, and σ_2, the second over the third.

Two concepts are essential: the real subring of the cyclotomic ring $\mathbb{Z}[\omega]$ (2)

$$\mathbb{Z}[\tau] := \{a + b\tau | a,b \in \mathbb{Z}\},$$

and an automorphism of $\mathbb{Z}[\omega]$

$$(.)^\bullet : \mathbb{Z}[\omega] \mapsto \mathbb{Z}[\omega] \text{ such that } \omega^\bullet = \omega^3. \tag{3}$$

This implies, for example, that

$$\tau^\bullet = (\omega^2 - \omega^3)^\bullet = (\omega^\bullet)^2 - (\omega^\bullet)^3 = -(\tau + 1).$$

Hence $(.)^\bullet$ can be restricted on $\mathbb{Z}[\tau]$.

We introduce the complexity measure $\mu(u) := |u^\bullet|^2$. and extend it to exact unitaries as $\mu(U[u,v,k]) := \mu(u)$, where $\mu(u)$ takes values from $\mathbb{Z}[\tau]$.

The exact synthesis algorithm, given in Fig. 1, performs a sequence of complexity reductions by applying an $\mathcal{F}\mathcal{T}^k$ operation at each step. T'he length (depth) n of the output circuit is guaranteed to be in $\Theta(\log(\mu(U)))$. The algorithm requires at most $O(n)$ arithmetic operations and outputs an $\langle \mathcal{F}, \mathcal{T} \rangle$ circuit with $O(n)$ gates. The Supplemental Material [12] contains supporting proofs.

Input: U – exact unitary
procedure EXACT-SYNTHESIZE(U)
  $g \leftarrow \mu(U), V \leftarrow U, C \leftarrow$ (empty circuit)
  while $g \geq 2$ **do**
    $J \leftarrow \arg\min_{j \in \{1,\ldots,10\}} \mu(\mathcal{F}\mathcal{T}^j V)$
    $V \leftarrow \mathcal{F}\mathcal{T}^J V, g \leftarrow \mu(V)$
    $C \leftarrow \mathcal{F}\mathcal{T}^{10-J} C$
  end while
  Find k, j such that $V = \omega^k \mathcal{T}^j$
  $C \leftarrow \omega^k \mathcal{T}^j C$
end procedure
Output: C – circuit over $\langle \mathcal{F}, \mathcal{T} \rangle$ that implements U

FIG. 1. Exact synthesis algorithm.

Approximation.—On it first stage the compilation algorithm approximates the target $R_z(\phi)$ rotation or $R_z(\phi)X$ gate.

Approximation consists of two steps. First, we find a cyclotomic integer u that is in ε proximity to $e^{-i\phi/2}$.

The second steps solves a relative norm equation to complete cyclotomic integer u with v from $\mathbb{Z}[\omega]$ such that $U[u, v, 0]$ is a unitary matrix.

For the first step, the distance between $R_z(\phi)$ and U simplifies to

$$d(R_z(\phi), U[u, v, 0]) = \sqrt{1 - |\mathrm{Re}(ue^{i\phi/2})|},$$

where the precision depends only on u, the top left entry of $U[u, v, 0]$. Therefore, it is sufficient to find u from $\mathbb{Z}[\omega]$ such that $\sqrt{1 - |\mathrm{Re}(ue^{i\phi/2})|} \leq \varepsilon$.

However, there must also exist a v from $\mathbb{Z}[\omega]$ such that $U[u, v, 0]$ is unitary. In particular, $v \in \mathbb{Z}[\omega]$ must satisfy the norm condition

$$|v|^2 = (1 - |u|^2)/\tau = (1 - |u|^2)(1 + \tau). \tag{4}$$

We interpret the right-hand side of this condition as an element of the $\mathbb{Z}[\tau]$ ring which makes (4) a special instance of the relative norm equation

$$N_i(x) = \xi, \tag{5}$$

where $\xi \in \mathbb{Z}[\tau]$, $x \in \mathbb{Z}[\omega]$, and $N_i : \mathbb{Z}[\omega] \to \mathbb{Z}[\tau]$, $N_i(x) \mapsto xx^*$ is the relative norm map. For (5) to be solvable it is necessary that $\xi > 0$ and $\xi^\bullet > 0$, but these conditions are not sufficient.

Equation (4) is only solvable for a fraction of randomly generated values of u. However, a significant fraction of such values allows for "easy" solutions of the norm equation and we can efficiently test if a given case is easy.

Powerful algorithms exist for solving relative norm equations in algebraic number rings [14–16]. In the Supplemental Material [12], we give an algorithm to solve the norm equation that runs in probabilistically polynomial time provided the right-hand side of (5) is easy to factor. We prove, under a given number theory conjecture (Conjecture 9 in the Supplemental Material [12]), that an easy instance can be found after $O(\log(1/\varepsilon))$ random trials. That is, given $\xi > 0$, $\xi^\bullet > 0$, and $p = \xi\xi^\bullet$ is an integer prime, we prove that (5) is solvable if and only if p mod 5 is either 1 or 0. The probability of encountering an easily solvable norm equation is thus no less than the probability of encountering a prime number of the form $5m + 1$ in a stream of random integers of a certain size.

The compilation algorithm is given in Fig. 2 (procedures listed in the Supplemental Material [12]). Each iteration uses RANDOM-SAMPLE to randomly pick a $u \in \mathbb{Z}[\omega]$ that is in ε proximity of $e^{-i\varphi/2}$. The corresponding norm equation is then tested for easy solution (line 9). An easy solution v is found in $O(1/\varepsilon)$ random trials.

Input: ϕ – defines $R_z(\phi)$, ε – precision
1: $C \leftarrow \sqrt{\varphi/4}$
2: $m \leftarrow \lceil \log_\tau(C\varepsilon) \rceil + 1$
3: Find k such that $\theta = -\phi/2 - \pi k/5 \in [0, \pi/5]$
4: *not-found* $\leftarrow$ true, $u \leftarrow 0, v \leftarrow 0$
5: **while** *not-found* **do**
6: $u_0 \leftarrow$ RANDOM-SAMPLE$(\theta, \varepsilon, 1)$
7: $\xi \leftarrow \varphi\left(\varphi^{2m} - |u_0|^2\right)$
8: $fl \leftarrow$ EASY-FACTOR(ξ)
9: **if** EASY-SOLVABLE(fl) **then**
10: *not-found* $\leftarrow$ false
11: $u \leftarrow \omega^k \tau^m u_0$
12: $v \leftarrow \tau^m$SOLVE-NORM-EQUATION(ξ)
13: **end if**
14: **end while**
15: $C \leftarrow$ EXACT-SYNTHESIZE$(U[u, v, 0])$
Output: Circuit C such that $d(C, R_z(\phi)) \leq \varepsilon$

FIG. 2. Compilation algorithm.

Once v is found the unitary $U[u, v, 0]$ is constructed (lines 11–12). It can be proved that the complexity measure μ of $U[u, v, 0]$ is in $O(1/\varepsilon)$. Finally, the exact synthesis algorithm (line 15) outputs the corresponding circuit. The depth of the circuit output by EXACT-SYNTHESIZE (Fig. 1) is bound by log of the complexity measure of the sample, making the depth $O(\log(1/\varepsilon))$.

Experimental results.—We evaluate the approximation quality of our algorithm on several sets of inputs. We obtain empirical evidence that the depth of the compiled circuits scales as $O(\log(1/\varepsilon))$. Experiment details are reported in the Supplemental Material [12] where approximations of frequently used single qubit gates are also given.

Figure 3(a) shows precision ε versus the circuit depth for rotations by angles $\pi/2^k$ used in the quantum Fourier transform. Figure 3(b) compares the circuit depth of our algorithm (NTA) versus brute force search (BFS) for $R_Z(\phi)$ rotations where angles ϕ are chosen to cover the interval [0, 2π] uniformly. BFS is performed over a database of optimal $\langle \sigma_1, \sigma_2 \rangle$ circuits of length up to 25 gates; BFS can only achieve precisions around $10^{-2.5}$ or coarser. The figures show that NTA produces circuits that are within 18% of the minimum circuit depth for $R_z(\phi)$ rotations. We also find that the NTA implementaion (in C++) has a runtime that scales slower than $\log^2(1/\varepsilon)$ when ε tends to zero.

Weaves.—A *weave* corresponds to a circuit generated by σ_1^2, σ_2^2 gates and their inverses.

Approximating unitaries with weaves requires replacing EXACT-SYNTHESIZE in Fig. 2 with EXACT-SYNTHESIZE-W (Fig. 4), which efficiently finds the first easy norm instance that also satisfies an extra condition—$U[u, v, 0]$ must be representable by a weave—and outputs the corresponding circuit (proof in [12]).

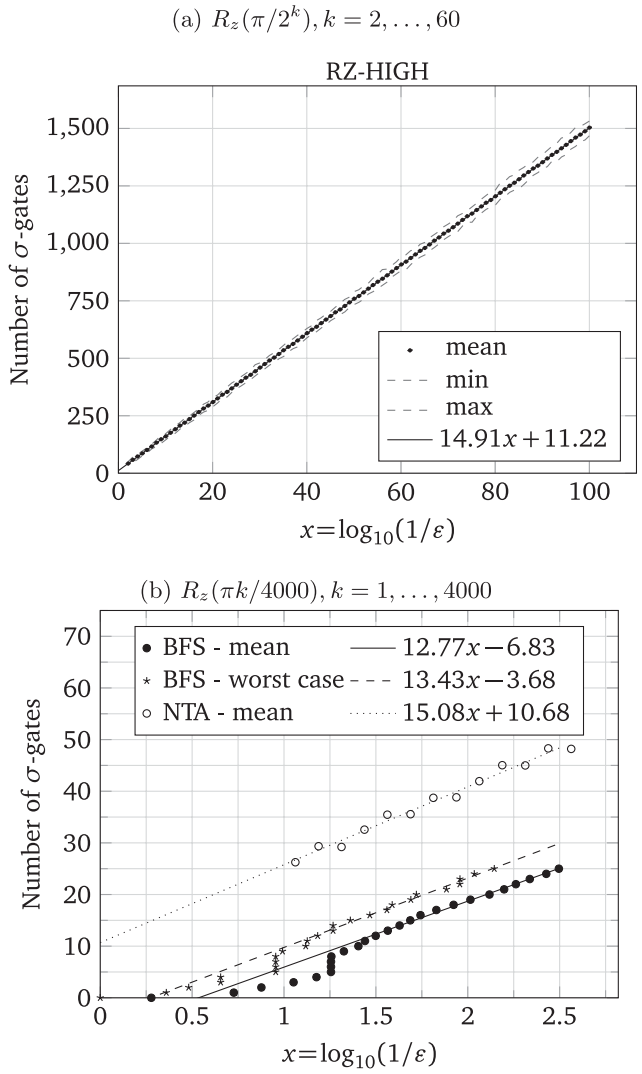

FIG. 3. (a) Number of σ gates needed to achieve approximation precision ε using our number theoretic algorithm (NTA) on $R_Z(\phi)$ rotations with $\phi = \pi/2^k$. (b) Comparison of the number of σ gates needed to achieve approximation precision ε using NTA versus brute force search (BFS), for $R_Z(\phi)$ rotations,where ϕ covers uniformly the range of angles $[0, 2\pi]$.

Experiments (see the Supplemental Material [12]) indicate that the number of random trials required to find an easy instance increases by a constant factor over the generic braid case. We also find that the number of $\langle \sigma_1, \sigma_2 \rangle$ gates needed scales as $2(9.67\log_{10}(1/\varepsilon) + 6.61)$. The factor of 2 reflects that each $\langle \sigma_1^2, \sigma_2^2 \rangle$-circuit generator is equivalent to two $\langle \sigma_1, \sigma_2 \rangle$-circuit generators. Thus, the number of elementary gates required to achieve a desired precision with weaves is 30% larger than with generic braid patterns.

Two-qubit gate synthesis.—As suggested in [17], the existence of an entangling two-qubit gate that can be exactly represented as a unitary anyon braid remains open for future work. However, our techniques can be used to efficiently ε approximate two-qubit gates with asymptotically depth-optimal braid patterns.

Input: U – exact unitary
1: **procedure** EXACT-SYNTHESIZE-W(U)
2: $g \leftarrow \mu(U), V \leftarrow U, C \leftarrow$ (empty circuit)
3: **while** $g \geq 4$ **do**
4: $J \leftarrow \arg\min_{j \in \{1,\ldots,5\}} \mu(\mathcal{F}\mathcal{T}^{2j}V)$
5: $V \leftarrow \mathcal{F}\mathcal{T}^{2J}V, g \leftarrow \mu(V)$
6: $C \leftarrow \mathcal{F}\mathcal{T}^{10-2J}C$
7: **end while**
8: **if** C has an odd number of $\mathcal{F}$ gates **then**
9: Remove the first $\mathcal{F}$ gate from C, $V \leftarrow \mathcal{F}V$
10: **end if**
11: **if** V can be represented as "weave" **then**
12: $C \leftarrow VC$, $V \leftarrow Id$
13: **end if**
14: **return** (C, V)
15: **end procedure**
Output: C – "weave" circuit over $\langle \mathcal{F}, \mathcal{T} \rangle$, V – non-"weave" part of the unitary, $CV = U$.

FIG. 4. Exact synthesis algorithm for weaves.

Consider the braid pattern found using BFS in [6] that approximates the gate controlled-σ_2^2 $[\Lambda(\sigma_2^2)]$. We can reduce its implementation to compilation of a global phase matrix and then use a variation of our approximation algorithm to yield a braid circuit of depth $O(\log(1/\varepsilon))$.

The algorithm to approximate nontrivial global phases is as follows. Make a sequence of calls to RANDOM-SAMPLE (0, $\delta/4$, 1) (Supplemental Material [12]). For any $u \in \mathbb{Z}[\omega]$ generated by a call, $|u|^2 < 1$ while $1 - |u|^2 < (\delta/4)^2$. Check whether the relative norm equation $|v|^2 = (1 - |u|^2)(1 + \tau)$ is an easy instance. Once an easy instance is obtained, use the exact Fibonacci matrix $U = U[u, v, 0]$ which is in the $\delta/4$-vicinity of Id.

Our experiments indicate that U can be decomposed into a weave of the form $\omega^k b$, $b = b(\sigma^2)$, $k \in \{0, \ldots, 9\}$ in probabilistically polynomial runtime. If k is 0 or 5, we reject the solution and continue drawing random samples. We observe empirically that $k \in \{0, 5\}$ is 4 times less likely than the opposite case, so it will take on average 5/4 successful weave decompositions to find one with $k \notin \{0, 5\}$. If k is even then b^{-1} is a $\delta/4$ approximation of one of the global phases of interest. If k is odd then b^{-2} is a $\delta/2$ approximation of one such phase [18]. Details and examples of braid patterns approximating $\Lambda(\sigma_2^2)$ are given in the Supplemental Material [12].

Conclusions.—We have developed an algorithm for optimal representation of single-qubit unitaries as braid patterns in the Fibonacci anyon basis in topological quantum computing. Our algorithm enables efficient compilation of single-qubit unitaries into circuits of asymptotically optimal depth $O(\log(1/\varepsilon))$ for an arbitrary target precision ε.

Compilation of an axial rotation to precision 10^{-30} takes less than 80 ms on average on a regular classical desktop

computer. Extensions of our algorithm can be used to produce both weaves and two-qubit gate approximations. Consequently, our topological compiler significantly improves on state-of-the-art solutions in both an asymptotic and practical sense. Future work includes proving (or disproving) the existence of a nonleaking two-qubit gate in the anyon braid representation.

We wish to thank Andreas Blass, Michael Freedman, Yuri Gurevich, Matthew Hastings, Martin Roeteller, Cameron Stewart, Zhenghan Wang, and Jon Yard for useful discussions.

[1] S. Sarma, M. Freedman, and C. Nayak, Phys. Rev. **94**, 166802 (2005).
[2] C. Nayak, S. Stern, A. Vatan, M. Freedman, and S. Sarma, Rev. Mod. Phys. **80**, 1083 (2008).
[3] M. Freedman, M. Larsen, and Z. Wang, Commun. Math. Phys. **227**, 605 (2002).
[4] M. Freedman, M. Larsen, and Z. Wang, Rev. Mod. Phys. **228**, 177 (2002).
[5] S. Simon, N. Bonesteel, M. Freedman, N. Petrovic, and L. Hormozi, Rev. Mod. Phys. **80**, 1083 (2006).
[6] L. Hormozi, G. Zikos, N. Bonesteel, and S. Simon, Phys. Rev. B **75**, 165310 (2007).
[7] C. Dawson and M. Nielsen, Rev. Mod. Phys. **228**, 177 (2002).
[8] S. Trebst, M. Troyer, Z. Wang, and A. Ludwig, Prog. Theor. Phys. Suppl. **176**, 384 (2008).
[9] J. Preskill, *Lecture Notes for Physics 219:Quantum Computation* (Springer, New York, 2004).
[10] A. Harrow, B. Recht, and I. Chuang, J. Math. Phys. (N.Y.) **43**, 4445 (2002).
[11] J. Bourgain and A. Gamburd, Inventiones Mathematicae **171**, 83 (2007).
[12] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.112.140504, which includes Refs. [19–30], for details of the mathematical apparatus and external algorithms used in the research presented in the Letter, it also contains significant number of examples and the bulk of experimental results and experimental statistics from which practical conclusions presented in the Letter had been inferred.
[13] A. Prasad, V. Shende, I. Markov, J. Hayes, and K. Patel, ACM J. Emerging Technol. Comput. Syst. **2**, 277 (2006).
[14] The Pari Group, User's Guide to PARI/GP (Institut de Mathematiques de Bordeaux, 2011), http://pari.math.u-bordeaux.fr/pub/pari/manuals/2.5.1/users.pdf.
[15] D. Simon, Math. Comput. **71**, 1287 (2002).
[16] H. Cohen, *Advanced Topics in Computational Algebraic Number Theory* (Springer, New York, 1999).
[17] R. Ainsworth and J. Slingerland, arXiv:1102.5029.
[18] An obvious version of the algorithm is to continue drawing the random samples until ω^k with nonzero even k is encountered. We may need to cycle through $5/2$ successful weave decompositions on average for this to happen.
[19] H. Lenstra, Jr., J. Lond. Math. Soc. **s2-10**, 457 (1975).
[20] V. Kliuchnikov, A. Bocharov, and K. Svore, arXiv:1310.4150.
[21] H. Cohen, *Number Theory, Volume I: Tools and Diophantine Equations* (Springer, New York, 2000).
[22] H. Cohen, *A Course in Computational Algebraic Number Theory* (Springer, New York, 1996).
[23] L. Washington, *Introduction to Cyclotomic Fields* (Springer, New York, 1997).
[24] N. Jacobson, *Basic Algebra I* (Dover, New York, 2009).
[25] D. Shanks, in *Proceedings of the Second Manitoba Conference on Numerical Mathematics* (Dover Publications, Mineola, 1973), p. 51.
[26] I. Damgard, G. Frandsen, J. Symb. Comput. **39**, 643 (2005).
[27] E. Bach, Math. Comput. **55**, 355 (1990).
[28] D. Wikstrom, Lect. Notes Comput. Sci. **3580**, 1189 (2005).
[29] J. Neukirch, *Algebraic Number Theory* (Springer, Berlin, 1999).
[30] J. Brylinski and R. Brylinski, arXiv:quant-ph/0108062.