Scaling laws for Shor's algorithm with a banded quantum Fourier transform

Article *in* Physical review A, Atomic, molecular, and optical physics · February 2013

DOI: 10.1103/PhysRevA.87.032333 · Source: arXiv

Yunseong Nam
IonQ Inc.
61 PUBLICATIONS **1,786** CITATIONS

Scaling laws for Shor's algorithm with a banded quantum Fourier transform

Y. S. Nam and R. Blümel
Department of Physics, Wesleyan University,
Middletown, Connecticut 06459-0155

(Dated: February 26, 2013)

Abstract

We investigate the performance of a streamlined version of Shor's algorithm in which the quantum Fourier transform is replaced by a banded version that for each qubit retains only coupling to its b nearest neighbors. Defining the performance $P(n,b)$ of the n-qubit algorithm for bandwidth b as the ratio of the success rates of Shor's algorithm equipped with the banded and the full bandwidth ($b = n-1$) versions of the quantum Fourier transform, our numerical simulations show that $P(n,b) \approx \exp[-\varphi_{max}^2(n,b)/100]$ for $n < n_t(b)$ (non-exponential regime) and $P(n,b) \approx 2^{-\xi_b(n-8)}$ for $n > n_t(b)$ (exponential regime), where $n_t(b)$, the location of the transition, is approximately given by $n_t(b) \approx b + 5.9 + \sqrt{7.7(b+2) - 47}$ for $b \gtrsim 8$, $\varphi_{max}(n,b) = 2\pi[2^{-b-1}(n-b-2) + 2^{-n}]$, and $\xi_b \approx 1.1 \times 2^{-2b}$. Analytically we obtain $P(n,b) \approx \exp[-\varphi_{max}^2(n,b)/64]$ for $n < n_t(b)$ and $P(n,b) \approx 2^{-\xi_b^{(a)} n}$ for $n > n_t(b)$, where $\xi_b^{(a)} \approx \frac{\pi^2}{12\ln(2)} \times 2^{-2b} \approx 1.19 \times 2^{-2b}$. Thus, our analytical results predict the φ_{max}^2 scaling ($n < n_t$) and the 2^{-2b} scaling ($n > n_t$) of the data perfectly. In addition, in the large-n regime, the prefactor in $\xi_b^{(a)}$ is close to the results of our numerical simulations and, in the low-n regime, the numerical scaling factor in our analytical result is within a factor 2 of its numerical value. As an example we show that $b = 8$ is sufficient for factoring RSA-2048 with a 95% success rate.

PACS numbers: 03.67.Lx

I. INTRODUCTION

While the art of integer factoring lay dormant, literally for millennia, and not much progress beyond the crudest methods, such as trial division and looking for differences of squares, had been made [1], the advent of the widely used RSA cryptosystem [2] has recently propelled the factoring of large integers from the arcane recesses of an ancient mathematical discipline into the lime light of contemporary physics and mathematics. The reason is that a powerful factoring algorithm may be used in a frontal attack on the RSA cryptosystem, and, if successful, immediately reveals untold scores of government, military, and financial secrets [3, 4]. No wonder then, that the first substantial breakthrough in factoring in centuries, the quadratic number sieve [1, 5], occurred shortly after the initial publication of the RSA method [2]. Using the quadratic number sieve, RSA keys with up to 100 decimal digits can now routinely be cracked [6] and are not safe any more. In 1993, the general number field sieve [7] added even more power to factoring attacks on RSA and was used successfully to factor the RSA challenge number RSA-768 (232 decimal digits) [8], which prompted the US National Institute of Standards and Technology (NIST) to recommend retirement of all RSA keys with 1024 binary digits or less [9]. However, no matter how powerful these modern factoring algorithms are, they are based on classical computing algorithms, executed on classical computers and without further improvements will never be able to crack an RSA key consisting of 5000 decimal digits or more (see Sec. VIII). But not only classical computing profited from the advent of the RSA crypto-system, so did quantum computing [10]. In 1994, Shor demonstrated that a certain quantum algorithm executed on a quantum computer is exponentially more powerful than any currently known classical factoring scheme and poses a real threat to RSA-encrypted data [11]. Since its inception in 1994, Shor's algorithm has maintained its status as the gold standard in quantum computing, and progress in quantum computer implementation is frequently measured in terms of the size of semiprimes that a given quantum computer can factor [12, 13]. While, compared with classical factoring algorithms, Shor's algorithm is tremendously more powerful, it should not come as a surprise that in order to break currently employed RSA keys, an enormous number of quantum operations still need to be performed. Therefore, any advance in streamlining practical implementations of Shor's algorithm are welcome that result in reducing the number of required quantum operations. A central component of Shor's algorithm is a quantum Fourier

Transform [10] and our paper focuses on how to perform this part of Shor's algorithm with the least number of quantum gates and gate operations that still guarantee acceptable performance of the algorithm.

Our paper is organized in the following way. In Sec. II we present Shor's algorithm. This section also serves to introduce the basic notation and explains the central position of the quantum Fourier transform in Shor's algorithm. While the original version of Shor's algorithm [11] is formulated with the help of a full implementation of the quantum Fourier transform, it turns out that a reduced, approximate version of the quantum Fourier transform, the banded quantum Fourier transform [14–16], yields surprisingly good results when used in conjunction with Shor's algorithm. The banded quantum Fourier transform is introduced and discussed in Sec. III. In order to assess the influence of the banded quantum Fourier transform on the performance of Shor's algorithm, we need an objective performance measure. Our performance measure is defined in Sec. IV. In Sec. V, based on the performance measure defined in Sec. IV, we investigate numerically the performance of a quantum computer for various bandwidths b as a function of the number of qubits n. We find that for fixed b the quantum computer exhibits two qualitatively different regimes, exponential for large n and non-exponential for small n. We also find that relatively small $b \lesssim 10$ are already sufficient for excellent quantum computer performance, even for n so large as to be interesting for the factoring of semiprimes N of practical interest. These numerical findings are then investigated analytically in Sec. VI. In Sec. VI A, we show an important property of the performance measure, i.e. approximate separability, which allows us to analyze analytically the large-n behavior (Sec. VI B) and the small-n behavior (Sec. VI C) of the numerical data presented in Sec. V. In particular, we are able to predict analytically the scaling functions of the data in the large-n and small-n regimes. In Sec. VII we compare our work with the related pioneering work of Fowler and Hollenberg [15]. While the final results are similar, our approach differs substantially from the approach in [15]. Factoring actual semiprimes, our approach is more realistic than the approach taken in [15] and may serve to check the results reported in [15]. In addition, we report a host of new results. In Sec. VIII we discuss our results and conclude the paper in Sec. IX. In order not to break the flow of exposition in the main text of our paper, some technical material is relegated to three appendices. In Appendix A we prove existence and uniqueness of an order-2 element for any semiprime N. In Appendix B we compute an analytical bound for the maximal possible order ω of a given semiprime N. In Appendix C, we provide an auxiliary result on the distribution of an inverse factor of ω, needed for one of our analytical results reported in Sec. VI.

II. SHOR'S ALGORITHM

Progress in quantum computing happens in fits and starts. Periods of stagnation and pessimism are followed by unexpected breakthroughs and optimism. Shor's algorithm is a case in point. Following a lull in quantum computing during which the only known quantum algorithms were of an "academic" nature, Shor's algorithm, the first "useful" quantum algorithm, instantly revived the field when it burst on the scene, quite unexpectedly, in 1994 [11]. Shor's algorithm is quantum mechanics' answer to a task that is hard or impossible to perform on any classical computer: factoring large semiprimes N. To accomplish this task, Shor's algorithm makes use of the entire palette of quantum effects that result in an exponential speed-up of the quantum algorithm with respect to any currently known classical factoring algorithm: superposition, interference, and entanglement. Shor's algorithm is based on Miller's algorithm [17], a classical factoring algorithm. Miller's algorithm determines the factors of a semiprime $N = pq$, where $p \neq q$ are prime, according to the following procedure. First, we choose a positive integer $1 < x < N$, called the seed, relatively prime to N, i.e. $\gcd(x, N) = 1$, where gcd denotes the greatest common divisor. Then, we determine the smallest positive integer ω, called the order of x, such that

$$x^{\omega} \mod N = 1. \tag{1}$$

For Miller's algorithm to work, we require (i) that ω is even and (ii) that $(x^{\omega/2}+1) \mod N \neq 0$. Both conditions need to be fulfilled. If even one is not fulfilled, we need to choose another x and try again. There is a high probability that this will succeed after only a few trials [10, 15, 18]. Having found a seed x satisfying both conditions, we write (1) in the form

$$[(x^{\omega/2} - 1)(x^{\omega/2} + 1)] \mod N = 0, \tag{2}$$

which implies that N divides the product on the left-hand side of (2). This might be accomplished if N divides $x^{\omega/2} - 1$, which implies $x^{\omega/2} \mod N = 1$. This, however, is impossible, because $\omega/2 < \omega$ and ω, according to (1), is the smallest such exponent. Another hypothetical possibility is that N divides the second factor in (2). This, however, is excluded according to condition (ii). The only remaining possibility is that p divides one of the factors in (2) and q divides the other. Appropriately naming the factors of N, we have

$$p = \gcd(x^{\omega/2} - 1, N), \quad q = \gcd(x^{\omega/2} + 1, N), \tag{3}$$

and the factoring problem is solved. So, if Miller's classical algorithm does the job, why do we need Shor's quantum algorithm? The answer is that finding the order ω on a classical computer is an algorithmically hard problem that, for a generic seed x, is impossible to perform on a classical computer within a reasonable execution time for semiprimes N with more than 5000 digits (see Sec. VIII). This is where Shor's algorithm comes in. Using a quantum Fourier transform to find the order ω, Shor's algorithm makes order-finding tractable on a quantum computer. This is how it works.

First, we define the function

$$f(k) = x^k \mod N, \tag{4}$$

where k is an integer with $k \geq 0$. Since $f(k+\omega) = f(k)$, the function f turns order finding into period finding. Since periods may be found by a Fourier transform, the central idea of Shor's algorithm is to use a quantum Fourier transform to determine ω. To implement this idea [10, 11, 17, 18], we work with a quantum computer consisting of two quantum registers, register I and register II. We assume that both registers consist of n qubits. In order to reliably determine ω for given N, care must be taken to choose n at least twice as large as the number of binary digits of N [10, 18]. We strictly observe this requirement in Sec. V [see (64)], where we present our numerical work. We start by initializing both registers to 0 such that the initial state of the quantum computer is

$$|\psi\rangle = |0, \ldots, 0\rangle_I \, |0, \ldots, 0\rangle_{II}. \tag{5}$$

Next, we initialize register I with a superposition of all integers from 0 to $2^n - 1$ by applying a single-qubit Hadamard transform [10] to each of the n qubits of register I, resulting in the state

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle_I \, |0, \ldots, 0\rangle_{II}, \tag{6}$$

where we introduced an intuitive equivalence, whereby an integer $k \geq 0$ is mapped onto the n qubits of a register according to the binary digits of k. Now, we make use of the function f

defined in (4) to fill register II with the f-images of register I. This results in the computer state

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle_I \, |f(k)\rangle_{II}. \tag{7}$$

This step entangles registers I and II. The function f induces equivalence classes

$$[s_0] = \{s_0 + k\omega, \quad 0 \leq k \leq K(s_0) - 1\} \tag{8}$$

on $\{0, \ldots, 2^n - 1\}$ with representatives $0 \leq s_0 \leq \omega - 1$, where $K(s_0)$ is the smallest integer with $s_0 + K(s_0)\omega \geq 2^n$. In other words, $K(s_0)$ is the number of elements in the equivalence class $[s_0]$. Since the range of s values is 2^n and the spacing is ω, we obtain, approximately,

$$K(s_0) \approx \frac{2^n}{\omega}. \tag{9}$$

Because of the periodicity of f, each member of $[s_0]$ is mapped onto $f(s_0)$. Therefore, if a measurement of register II collapses this register into the state $|f(s_0)\rangle_{II}$, the quantum computer is in the state

$$|\psi_i\rangle = \frac{1}{\sqrt{K(s_0)}} \sum_{k=0}^{K(s_0)-1} |s_0 + k\omega\rangle_I \, |f(s_0)\rangle_{II}. \tag{10}$$

We may now apply a quantum Fourier transform

$$\hat{U}^{(QFT)} = \frac{1}{\sqrt{2^n}} \sum_{k,l=0}^{2^n-1} |l\rangle \; \exp(2\pi i l k/2^n) \; \langle k| \tag{11}$$

to register I of $|\psi_i\rangle$ to obtain

$$|\psi_f\rangle = \frac{1}{\sqrt{K(s_0)2^n}} \sum_{k=0}^{K(s_0)-1} \sum_{l=0}^{2^n-1} \exp[2\pi i l (s_0 + k\omega)/2^n] \, |l\rangle_I \, |f(s_0)\rangle_{II}. \tag{12}$$

A measurement of register I then collapses $|\psi_f\rangle$ into $|l\rangle$ with probability

$$\begin{aligned} \tilde{P}(n, l, \omega) &= \frac{1}{2^n K} \left| \sum_{k=0}^{K-1} \exp(2\pi i l k \omega/2^n) \right|^2 \\ &= \frac{\sin^2(K\pi\omega l/2^n)}{2^n K \sin^2(\pi\omega l/2^n)}, \end{aligned} \tag{13}$$

where here and in the following we suppressed the argument s_0 of K. Apparently, $\tilde{P}(n, l, \omega)$ is sharply peaked at l values for which $\omega l/2^n$ is close to an integer. As a consequence, these

l values will appear as a result of measurement with a high probability. Subsequent analysis of the measured peak location on a classical computer then reveals the factors of N with high probability [10]. This step is called classical post processing [10, 18]. Equation (13) is the starting point of our analysis of the performance of Shor's algorithm with a banded quantum Fourier transform in Sec. IV.

Several experimental demonstrations of Shor's algorithm have been published [12, 13, 19–21]. Since it is exceedingly difficult to experimentally control more than a handful of qubits, the numbers N factored in these experiments are very small, currently not exceeding $N = 21$ [13]. Therefore, reaching higher N is facilitated by reducing the requirements to run Shor's algorithm on a quantum computer. One such optimization is the use of an approximate, banded quantum Fourier transform [14] instead of the the full quantum Fourier transform (11). Further optimization is possible by using a banded version of the semi-classical quantum Fourier transform [22] defined in the following section.

III. BANDED QUANTUM FOURIER TRANSFORM

A direct circuit implementation of the Fourier transform defined in (11) requires $n(n+1)/2$ two-qubit quantum gates [10]. In [22], it was shown that, when followed by measurements, as required by Shor's algorithm, an equivalent quantum circuit, consisting exclusively of single-qubit gates, is exactly equivalent to the two-qubit realization of the quantum Fourier transform. Figure 1 (a) illustrates this single-qubit realization of the quantum Fourier transform for the special case of five qubits (we classify the conditional rotation gates θ in Fig. 1 as single-qubit gates since they are controlled by classical input and act coherently only on a single qubit). This circuit still requires $\sim n^2$ gate operations, but since they are performed by single-qubit gates, experimental implementation of this single-qubit circuit is considerably simpler. In contrast to the full two-qubit implementation of the quantum Fourier transform, where the measurements may occur simultaneously at the end of the quantum computation, the measurements in the single-qubit version of the quantum Fourier transform [denoted by the M gates in Fig. 1 (a)] occur sequentially and their (classical) measurement results are used to control the phase rotation gates θ. As first pointed out by Coppersmith [14], even this quantum circuit may still be optimized by working with an approximate, banded quantum Fourier transform as illustrated in Fig. 1 (b).

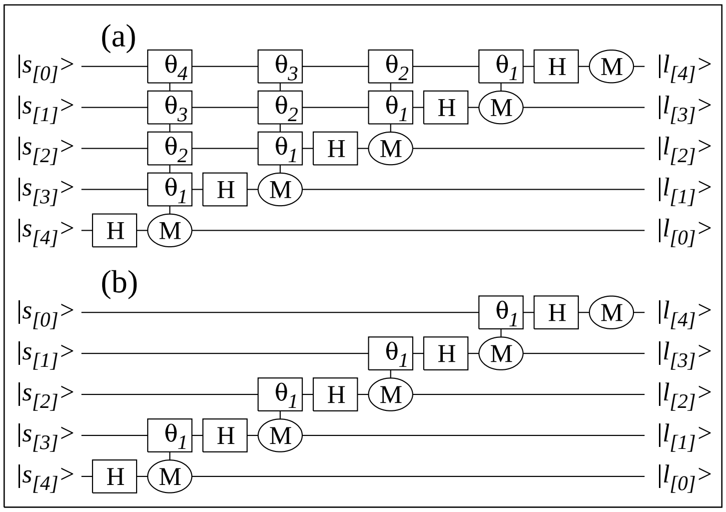

FIG. 1: Logic circuit of a five-qubit implementation of the single-qubit realization of the quantum Fourier transform [22]. (a) Full implementation (bandwidth $b = 4$); (b) truncated implementation (bandwidth $b = 1$). H, θ, and M denote Hadamard, single-qubit conditional rotation, and measurement gates, respectively.

The banded quantum Fourier transform $\hat{U}_b^{(QFT)}$ [see Fig. 1 (b)] is obtained from the full implementation of the single-qubit quantum Fourier transform [see Fig. 1 (a)] by retaining only the coupling to b nearest neighbors of a given qubit. As illustrated in Fig. 1 (b) for the case $b = 1$, this results in a banded structure of the corresponding quantum circuit [16]. The name is also justified on theoretical grounds since the unitary matrix representing the circuit shown in Fig. 1 (b) has a banded structure [23]. The banded quantum Fourier transform of bandwidth b is the basis of our work presented in the following sections.

IV. PERFORMANCE MEASURE

The key idea of Shor's algorithm is to use superposition and entanglement to steer the quantum probability into qubits that correspond to numbers encoded in binary form, which will then, as a result of classical post-processing, reveal the factors of N. Our first task, therefore, is to locate the useful peaks after the quantum Fourier transform is performed. In order to define our performance measure, we are interested in how sharp these peaks are in l. For this purpose, we notice that $\tilde{P}(n,l,\omega)$ [see (13)] (up to a factor) is of the form

$$f(z) = \frac{\sin^2(Kz)}{\sin^2(z)}, \tag{14}$$

where K is a large integer, z is a real number, and $f(z)$ is sharply peaked at integer multiples of π. Since the shape of $f(z)$ is the same for z in the vicinity of each peak, it suffices to investigate the peak at $z = 0$ to determine the width of all the other peaks of $f(z)$. We define the half width Δz of $f(z)$ by requiring

$$f(\Delta z) = \frac{1}{2}. \tag{15}$$

Inspired by a second-order Taylor-series expansion of (15), we obtain the heuristic formula

$$\Delta z \approx \frac{1.39}{K}, \tag{16}$$

which, for $K > 10$, satisfies (15) to better than 10^{-3}. Applied to $\tilde{P}(n,l,\omega)$ in (13), we have

$$z = \frac{\pi\omega l}{2^n}, \tag{17}$$

and, therefore,

$$\Delta z = \frac{\pi\omega}{2^n}\Delta l \approx \frac{1.39}{K}, \tag{18}$$

from which we obtain

$$\Delta l \approx \left(\frac{2^n}{\omega K}\right)\left(\frac{1.39}{\pi}\right) \approx 0.44, \tag{19}$$

where we used (9). This result shows that the full width at half maximum of the l-peaks is only about one state and that this width is "universal" in the sense that it is independent of K, ω, and n.

Since a peak in $\tilde{P}(n,l,\omega)$ occurs whenever $\omega l/2^n$ is close to an integer, we define the l-integer closest to peak number j according to:

$$l_j = \left(\frac{2^n}{\omega}\right)j + \beta_j, \quad j = 0, 1, \ldots, \omega - 1, \tag{20}$$

where β_j, a rational number, ranges between $-1/2$ and $1/2$. Since the peaks in $\tilde{P}(n,l,\omega)$ are universal in the above sense and contain basically only a single state, namely l_j defined in (20), we use

$$\tilde{P}(n,l_j,\omega) \equiv \tilde{P}_j(n,\omega) \tag{21}$$

as the basis for our performance measure.

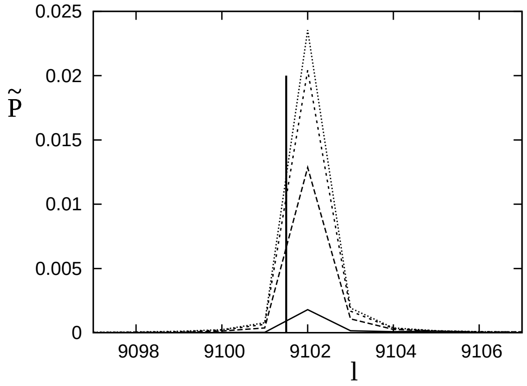

FIG. 2: Shape of a Fourier peak in l as a function of b for the semiprime $N = 247$ and order $\omega = 36$. Shown are the peaks for different bandwidths $b = 1$ (solid), $b = 2$ (long-dashed), $b = 3$ (short-dashed), and $b = 10$ (dotted). The vertical solid line is located at $l = 9101.5$.

Although the width of the peaks of $\tilde{P}(n, l, \omega)$ is narrow, according to (19) of the order of a single state, and although $|l_j\rangle$ carries most of the probability in peak number j of $\tilde{P}(n, l, \omega)$ (approximately 77% on average), there are nevertheless several states $|l\rangle$ inside of peak number j that occur with a small but still appreciable probability in a measurement of $|\psi_f\rangle$ in (12). These states are also useful for factoring during classical post-processing (see Sec. II and [10, 18]), and the question arises if these states should be included in the performance measure. Indeed, instead of determining the performance of Shor's algorithm on the basis of the single state $|l_j\rangle$, Fowler and Hollenberg [15], e.g., base their performance measure on the two closest states to the peaks in $\tilde{P}(n, l, \omega)$. We found that including more states in the performance measure is not necessary, since the width of the Fourier peaks in l is independent of the bandwidth b. At first glance this is surprising, since intuitively, we would think that the quality of the quantum Fourier transform should deteriorate with decreasing bandwidth b, possibly accompanied by a broadening of the Fourier peaks in l. That this is not so, and that the widths of the Fourier peaks are indeed independent of b, is demonstrated in Fig. 2 for the case $N = 247$ for $b = 1, 2, 3, 10$. Independent of b, the vertical line in the figure cuts each Fourier peak at approximately its midpoint, thus demonstrating that the widths of the Fourier peaks in l are indeed independent of b. Thus, upon a change in b, all l states under a Fourier peak respond in unison to the change in b. Therefore, a single l state, such as l_j, is an excellent representative of all the l states in its immediate vicinity.

Defining $\tilde{P}_j(n,b,\omega) = \tilde{P}(n,l_j,b,\omega)$ as the probability of obtaining $|l_j\rangle$ in a measurement of $|\psi_f\rangle$ if instead of the full quantum Fourier transform (11) the banded quantum Fourier transform (see Sec. III) is used and taking into account that the widths of the peaks in $\tilde{P}_j(n,b,\omega)$ do not change as b is varied, we use the ratio of the total probability of collapse into one of the states $|l_j\rangle$, given the bandwidth b, to that of the full bandwidth $b = n-1$, to capture the overall probability of obtaining the useful $|l\rangle$ states in the vicinity of $|l_j\rangle$. Thus, the normalized ratio is of the form

$$P(n,b,\omega) = \tilde{P}(n,b,\omega)/\tilde{P}(n,b=n-1,\omega), \tag{22}$$

where

$$\tilde{P}(n,b,\omega) = \sum_{j=0}^{\omega-1} \tilde{P}_j(n,b,\omega) \tag{23}$$

and $\tilde{P}(n,b=n-1,\omega)$ is the probability of collapsing into any one of the set of useful states $|l_j\rangle$ as a result of measuring $|\psi_f\rangle$, where $|\psi_f\rangle$ is generated from $|\psi_i\rangle$ by application of the full quantum Fourier transform $\hat{U}^{(QFT)}$ defined in (11). We use $P(n,b,\omega)$, defined in (22), as our performance measure throughout this paper.

Next, we derive an analytical expression for $\tilde{P}_j(n,b,\omega)$, valid for any bandwidth $0 \le b \le n-1$, that can be used in our performance measure (22). In order to find $\tilde{P}_j(n,b,\omega)$ we need to descend to the qubit-by-qubit level, since the bandwidth b in $\hat{U}_b^{(QFT)}$ refers to inter-qubit spacing on the qubit level in the circuit diagram of $\hat{U}_b^{(QFT)}$ [see Fig. 1 (b)]. We start with a representation of the quantum Fourier transform in bit-notation

$$\begin{aligned}\hat{U}^{(QFT)}|s\rangle &= \frac{1}{\sqrt{2^n}} \sum_{l=0}^{2^n-1} e^{\frac{2\pi i s l}{2^n}} |l\rangle \\ &= \frac{1}{\sqrt{2^n}} \prod_{m=0}^{n-1} \sum_{l_{[n-m-1]}=0}^{1} e^{2\pi i (.s_{[m]} s_{[m-1]} \ldots s_{[0]}) l_{[n-m-1]}} |l_{[n-m-1]}\rangle,\end{aligned} \tag{24}$$

where $s_{[\nu]}(l_{[\nu]})$ indicates the νth binary digit of s (νth binary digit of l) and

$$\left(.s_{[m]} s_{[m-1]} \ldots s_{[0]}\right) = \sum_{\nu=0}^{m} s_{[\nu]} 2^{-(m-\nu+1)}. \tag{25}$$

For bandwidth b, $\hat{U}_b^{(QFT)}|s\rangle$ then becomes

$$\hat{U}_b^{(QFT)}|s\rangle = \frac{1}{\sqrt{2^n}} \prod_{m=0}^{n-1} \sum_{l_{[n-m-1]}=0}^{1} e^{2\pi i [(.s_{[m]} s_{[m-1]} \ldots s_{[0]}) - (.00\ldots 0 s_{[m-b-1]} \ldots s_{[0]})] l_{[n-m-1]}} |l_{[n-m-1]}\rangle. \tag{26}$$

We may also write

$$\hat{U}_b^{(QFT)}|s\rangle = \sum_{l=0}^{2^n-1} B(s,l)|l\rangle, \tag{27}$$

where

$$B(s,l) = \frac{1}{\sqrt{2^n}} \exp\left\{2\pi i \sum_{m=0}^{n-1} \big[\Lambda_{m,0}(s) - \Lambda_{m,b+1}(s)\big] l_{[n-m-1]}\right\}, \tag{28}$$

and

$$\Lambda_{m,\lambda}(s) = \big(.00\ldots 0 s_{[m-\lambda]} s_{[m-\lambda-1]} \ldots s_{[0]}\big), \tag{29}$$

i.e. λ zeros are following the binary point. Defining

$$S_\lambda(s,l) = \sum_{m=0}^{n-1} \Lambda_{m,\lambda}(s) l_{[n-m-1]}, \tag{30}$$

we may express $B(s,l)$ in the form

$$B(s,l) = \frac{1}{2^{n/2}} \exp\Big\{2\pi i [S_0(s,l) - S_{b+1}(s,l)]\Big\}. \tag{31}$$

Sorting indices, $S_\lambda(s,l)$ may be written in the form

$$S_\lambda(s,l) = \frac{1}{2} \sum_{m=\lambda}^{n-1} \sum_{\mu=0}^{m-\lambda} \frac{s_{[n-m-1]} l_{[\mu]}}{2^{m-\mu}}. \tag{32}$$

We are now ready to apply the banded quantum Fourier transform to register I of the initial state $|\psi_i\rangle$[see (10)] and obtain with (27) and (31)

$$\begin{aligned}\hat{U}_b^{(QFT)}|\psi_i\rangle &= \hat{U}_b^{(QFT)} \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} |s_k\rangle \\ &= \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} \sum_{l=0}^{2^n-1} B(s_k,l)|l\rangle \\ &= \frac{1}{\sqrt{2^n K}} \sum_{k=0}^{K-1} \sum_{l=0}^{2^n-1} \exp\big\{2\pi i [S_0(s_k,l) - S_{b+1}(s_k,l)]\big\}|l\rangle. \end{aligned} \tag{33}$$

From this we obtain

$$\tilde{P}_j(n,b,\omega) = \frac{1}{2^n K} \left| \sum_{k=0}^{K-1} \exp\big\{2\pi i [S_0(s_k,l_j) - S_{b+1}(s_k,l_j)]\big\} \right|^2, \tag{34}$$

which, using the expanded form (32) of S, can be written in the form

$$\tilde{P}_j(n,b,\omega) = \frac{1}{2^n K} \left| \sum_{k=0}^{K-1} e^{i[\Phi(n,s_k,l_j) - \varphi(n,b,s_k,l_j)]} \right|^2, \tag{35}$$

where

$$\Phi(n,s,l)=\pi\sum_{m=0}^{n-1}\sum_{\mu=0}^{m}\frac{s_{[n-m-1]}l_{[\mu]}}{2^{m-\mu}} \tag{36}$$

and

$$\varphi(n,b,s,l)=\pi\sum_{m=b+1}^{n-1}\sum_{\mu=0}^{m-b-1}\frac{s_{[n-m-1]}l_{[\mu]}}{2^{m-\mu}}. \tag{37}$$

While Φ in (36) is already in a form useful for numerical calculations, we now derive an expression for $\exp(i\Phi)$, which is more convenient for the analytical calculations in Sec. VI. We start by summing (36) in reverse order over m ($n-m-1\to m$) to obtain:

$$\begin{aligned}\Phi(n,s,l)&=\pi\sum_{m=0}^{n-1}\sum_{\mu=0}^{n-m-1}\frac{s_{[m]}l_{[\mu]}}{2^{n-1-m}2^{-\mu}}\\&=\frac{\pi}{2^{n-1}}\sum_{m=0}^{n-1}2^m s_{[m]}\sum_{\mu=0}^{n-m-1}2^{\mu}l_{[\mu]}.\end{aligned} \tag{38}$$

If we extend the μ sum in (38) to include terms ranging from $\mu=n-m$ to $\mu=n-1$, we notice that these extra terms generate even multiples of 2π in (38). Therefore, when computing $\exp(i\Phi)$, we can safely extend the μ sum to $\mu=n-1$, since the extra terms, generating even multiples of $2\pi i$ in the argument of the exponential function, do not contribute to $\exp(i\Phi)$. Therefore, we obtain:

$$\exp\left[i\Phi(n,s,l)\right]=\exp\left(\frac{\pi i}{2^{n-1}}\sum_{m=0}^{n-1}2^m s_{[m]}\sum_{\mu=0}^{n-1}2^{\mu}l_{[\mu]}\right). \tag{39}$$

Using the fact that

$$\sum_{m=0}^{n-1}2^m s_{[m]}=s \mod 2^n, \tag{40}$$

and similarly for l, we obtain

$$\exp\left[i\Phi(n,s,l)\right]=\exp\left\{\frac{2\pi i}{2^n}\big[(s \bmod 2^n)(l \bmod 2^n)\big]\right\}. \tag{41}$$

The factor $2\pi i/2^n$ in the exponent induces a modulo operation and we may also write

$$\exp\left[i\Phi(n,s,l)\right]=\exp\left\{\frac{2\pi i}{2^n}\big[(s \bmod 2^n)(l \bmod 2^n)\big] \bmod 2^n\right\}. \tag{42}$$

Using the formula

$$[(A \bmod M)(B \bmod M)] \mod M=(A\cdot B) \mod M \tag{43}$$

of elementary modular arithmetic, we may write (42) in the form:

$$\exp\left[i\Phi(n,s,l)\right] = \exp\left[\frac{2\pi i}{2^n}(s\cdot l) \bmod 2^n\right]. \tag{44}$$

Now, we use (20) and (8) with $s_0 = 0$ to obtain:

$$\exp\left[i\Phi(n,s_k,l_j)\right] = \exp\left[\frac{2\pi i}{2^n}(k2^n j + k\omega\beta_j) \bmod 2^n\right]. \tag{45}$$

The first term in parentheses contributes nothing to (45), since it is an integer and together with the prefactor in the exponent of (45), amounts to an even multiple of $2\pi i$. Therefore, (45) reduces to

$$\exp\left[i\Phi(n,s_k,l_j)\right] = \exp\left[\frac{2\pi i}{2^n}(k\omega\beta_j) \bmod 2^n\right]. \tag{46}$$

Since $k\omega \leq 2^n$ and $|\beta_j| < \frac{1}{2}$, we have $|k\omega\beta_j| < 2^n$. Therefore, the modulo operation in (46) is not needed any more and we obtain

$$\exp\left[i\Phi(n,s_k,l_j)\right] = \exp\left[2\pi i\left(\frac{k\omega\beta_j}{2^n}\right)\right]. \tag{47}$$

Thus we obtained a closed-form, analytical expression for $\exp(i\Phi)$.

Although [because of the presence of $\varphi(n,b,s_k,l_j)$ in (35)] not useful for the exact evaluation of (35), a well-justified approximation performed in Sec. VI allows us to compute

$$\Omega(n,l_j,\omega) = \sum_{k=0}^{K-1} \exp\left[i\Phi(n,s_k,l_j)\right] \tag{48}$$

separately. Using the formula for computing geometric sums, we obtain:

$$\begin{aligned}\Omega(n,l_j,\omega) &= \sum_{k=0}^{K-1} \left[\exp\left(2\pi i\omega\beta_j/2^n\right)\right]^k \\ &= \frac{1-\exp\left(2\pi i\omega\beta_j K/2^n\right)}{1-\exp\left(2\pi i\omega\beta_j/2^n\right)}.\end{aligned} \tag{49}$$

With (9) we obtain

$$\Omega(n,l_j,\omega) \approx \frac{1-\exp\left(2\pi i\beta_j\right)}{1-\exp\left(2\pi i\beta_j\omega/2^n\right)} \approx e^{i\pi\beta_j} K \frac{\sin\left(\pi\beta_j\right)}{(\pi\beta_j)}. \tag{50}$$

Since $\varphi(n,b=n-1,s,l)=0$, we note in passing that

$$\tilde{P}_j(n,b=n-1,\omega) = \frac{1}{2^n K}|\Omega(n,l_j,\omega)|^2. \tag{51}$$

We also need an analytical expression for the maximum value $\varphi_{max}(n,b)$ of $\varphi(n,b,s_k,l_j)$, defined as

$$\varphi_{max}(n,b) = \max_{k,j} \varphi(n,b,s_k,l_j). \tag{52}$$

From (37) it is clear that φ_{max} is obtained by setting all $s_{[n-m-1]}$ and $l_{[\mu]}$ values equal to 1. This procedure yields

$$\varphi_{max}(n,b) = \pi \sum_{m=b+1}^{n-1} \sum_{\mu=0}^{m-b-1} \frac{1}{2^{m-\mu}}. \tag{53}$$

Only the formula for evaluating geometric sums is needed to compute the value of φ_{max} in (53). We obtain

$$\varphi_{max}(n,b) = 2\pi[2^{-b-1}(n-b) - 2^{-b} + 2^{-n}]. \tag{54}$$

We now show that a quantum computer performs perfectly, no matter what b is, if ω is a power of 2, i.e.,

$$P(n,b,\omega) = 1, \quad \text{for } \omega = 2^{\alpha}, \quad \alpha \geq 0 \text{ integer}. \tag{55}$$

For such an ω, we notice that (i) the κth binary digit of any l_j is zero for $\kappa \leq n-\alpha$ since according to (20)

$$l_j = 2^{n-\alpha} j, \quad j = 0,1,\ldots,\omega-1 \tag{56}$$

is already integer, which implies $\beta_j = 0$, and (ii) the ιth binary digit of any equivalence class element in $[s_0]$ [see (8)] for $0 \leq \iota < \alpha$ is identical to that of s_0. Thus, we write $\varphi(n,b,s,l)$ in (37) in the form

$$\begin{aligned} \varphi(n,b,s,l) &= \pi\left(\sum_{m=n-\alpha+b+1}^{n-1} \sum_{\mu=0}^{m-b-1} \frac{s_{[n-m-1]} l_{[\mu]}}{2^{m-\mu}} + \sum_{m=b+1}^{n-\alpha+b} \sum_{\mu=0}^{m-b-1} \frac{s_{[n-m-1]} l_{[\mu]}}{2^{m-\mu}} \right) \\ &= \begin{cases} 0, & \text{if } \alpha \leq b+1, \\ \pi \sum_{m=n-\alpha+b+1}^{n-1} \sum_{\mu=n-\alpha}^{m-b-1} \frac{s_{[n-m-1]} l_{[\mu]}}{2^{m-\mu}}, & \text{if } \alpha > b+1, \end{cases} \end{aligned} \tag{57}$$

where the second equality was obtained by using (i). Now, we observe that the $n-m-1$th digit of s is bounded between 0 and $\alpha-b-2$ inclusively. Then, using (ii), we obtain

$$\begin{aligned} \varphi(n,b,s=s_k,l=l_j) &= \pi \sum_{m=n-\alpha+b+1}^{n-1} \sum_{\mu=n-\alpha}^{m-b-1} \frac{(s_k)_{[n-m-1]}(l_j)_{[\mu]}}{2^{m-\mu}} \\ &= \pi \sum_{m=n-\alpha+b+1}^{n-1} \sum_{\mu=n-\alpha}^{m-b-1} \frac{(s_0)_{[n-m-1]}(l_j)_{[\mu]}}{2^{m-\mu}} \\ &= \tilde{\varphi}_j, \end{aligned} \tag{58}$$

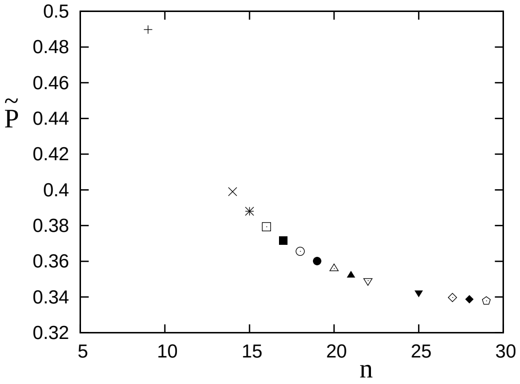

FIG. 3: Probability $\tilde{P}(n, b = 1, \omega = 6)$ as a function of n for 14 different semiprimes N with seeds chosen such that $\omega = 6$. As expected, the data clearly asymptotes to the value 1/3.

where $\tilde{\varphi}_j$ is a constant for any s_k and a given l_j. Inserting (58) in (35), $\tilde{P}_j(n, b, \omega)$ becomes

$$\begin{aligned}\tilde{P}_j(n,b,\omega) &= \frac{1}{2^n K}\left|\sum_{k=0}^{K-1} e^{i[\Phi(n,s_k,l_j)-\tilde{\varphi}_j]}\right|^2 \\ &= \frac{1}{2^n K}\left|e^{-i\tilde{\varphi}_j}\right|^2\left|\sum_{k=0}^{K-1} e^{i\Phi(n,s_k,l_j)}\right|^2 \\ &= \frac{1}{2^n K}\left|\Omega(n,l_j,\omega)\right|^2 = \tilde{P}_j(n,b=n-1,\omega), \qquad (59)\end{aligned}$$

where we used (48) and (51). With (23) and (59) we obtain

$$\tilde{P}(n,b,\omega) = \sum_{j=0}^{\omega-1} \tilde{P}_j(n,b=n-1,\omega) = \tilde{P}(n,b=n-1,\omega). \qquad (60)$$

Therefore, with (22), the normalized probability (the performance measure) $P(n, b, \omega)$ reads

$$P(n,b,\omega) = \frac{\tilde{P}(n,b=n-1,\omega)}{\tilde{P}(n,b=n-1,\omega)} = 1, \qquad (61)$$

which completes the proof.

Since $\omega = 2$ always exists (see Appendix A), this is an important observation, since the corresponding quantum computer works perfectly in this case for any n and any b. The trick, of course, is to find the seed x that yields $x^2 \mod N = 1$. This, however, is an unsolved problem for large N.

If ω is not a power of 2, we write it in the form

$$\omega = r2^\alpha, \quad r, \alpha \text{ integer}, \qquad (62)$$

where r is odd. For such an ω, according to (20), we may write l_j as

$$l_j = \left(\frac{2^{n-\alpha}}{r}\right) j + \beta_j. \tag{63}$$

Therefore, if j is a multiple of r, we have $\beta_j = 0$ and $\tilde{P}_j(n, b, \omega) = 1/\omega$, which is proved by following the corresponding steps for the case where ω is a power of 2. This means that the contribution of these j values to $\tilde{P}(n, b, \omega)$ is $1/r$. This is a constant contribution, which does not depend on either n or b. Therefore, if for large n the contributions to $\tilde{P}(n, b, \omega)$ tend to zero for the l_j peaks for which j is not a multiple of r, we expect $\tilde{P}(n, b, \omega)$ to approach $1/r$ for large n. This is demonstrated in Fig. 3, which shows $\tilde{P}(n, b = 1, \omega = 6)$ as a function of n. Since in this case $\omega = 3 \times 2^1$, we expect $\tilde{P}(n, b = 1, \omega = 6)$ to approach 1/3, which is clearly confirmed in Fig. 3.

V. NUMERICAL RESULTS

In this section we explore, numerically, the performance of Shor's algorithm supplied with a banded quantum Fourier transform of bandwidth b. The performance is measured objectively with the help of the quantitative performance measure $P(n, b, \omega)$ defined in (22). In contrast to a similar investigation by Fowler and Hollenberg [15], who use an effective ω for the investigation of the performance of the banded Shor algorithm, we opted for a more realistic simulation of the performance of Shor's algorithm using ensembles of semiprimes N together with their exact associated orders ω. Thus, our procedure for computing the performance measure is as follows. For given n we choose an ensemble of semiprimes $N = pq$ such that

$$n = \lfloor 2\log_2(N) + 1 \rfloor, \tag{64}$$

where $\lfloor \ldots \rfloor$ is the floor function [24]. This ensures that n is at least twice as large as the number of binary digits of N, as required to reliably determine the order ω with an n-qubit quantum computer [18, 25, 26]. For each N we compute its set of orders $\{\omega_1, \ldots, \omega_{a(N)}\}$, where $a(N)$ is the number of orders for given N. We also define the multiplicity of a given order ω as the number $\nu(\omega)$ of seeds x of order ω. Thus equipped, we compute the performance $P_N(n, b)$ as the properly weighted average

$$P_N(n, b) = \frac{1}{\varphi_E(N)} \sum_{j=1}^{a(N)} \nu(\omega_j) P(n, b, \omega_j), \tag{65}$$

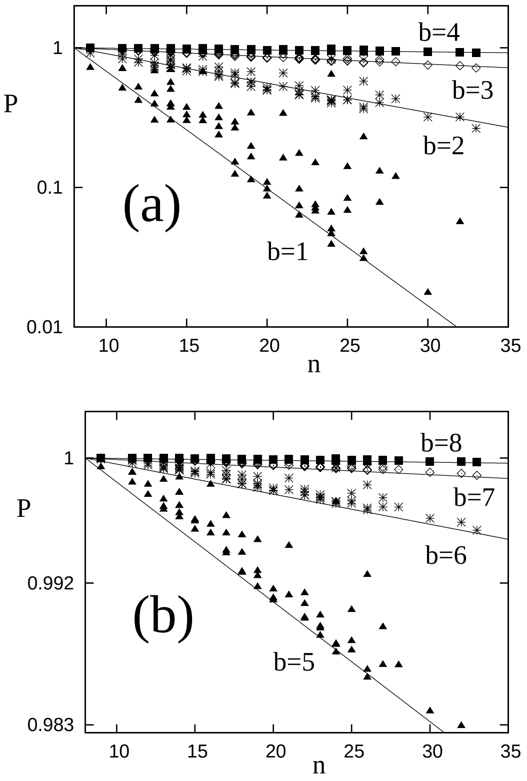

FIG. 4: Normalized probability P, represented by the properly averaged performance measure (65), for successful factorization of sample semiprimes N of binary length $\log_2(N) \sim n/2$ as a function of n for several bandwidths b, ranging from $b = 1$ to $b = 8$. (a) $b = 1$ (triangles), $b = 2$ (stars), $b = 3$ (diamonds), and $b = 4$ (squares). (b) $b = 5$ (triangles), $b = 6$ (stars), $b = 7$ (diamonds), and $b = 8$ (squares). The solid lines through the data points are the fit functions (66). Notice the visual similarity of (a) and (b), which illustrates the exponential scaling of ξ_b in b.

where $P(n, b, \omega)$ is defied in (22) and $\varphi_E(N)$ is Euler's totient function [27].

In Fig. 4 (a) we show $P_N(n, b)$ for various choices of N for $b = 1, \ldots, 4$ and n ranging from $n = 9$ to $n = 33$. Plot symbols correspond to particular N values and there are up to 7 semiprimes N per n. Overall we see that the data exhibit exponential behavior on average, which is well represented by the fit lines

$$P_>(n, b) = 2^{-\xi_b(n-8)}, \quad \xi_b = 1.1 \times 2^{-2b} \tag{66}$$

drawn through the data points. In Sec. VI B we present an analytical model that explains the b-scaling of (66) and in addition reproduces the pre-factor in (66) within 10%. Figure 4 (b) shows corresponding data for $b = 5, \ldots, 8$. Again, the data points behave exponentially

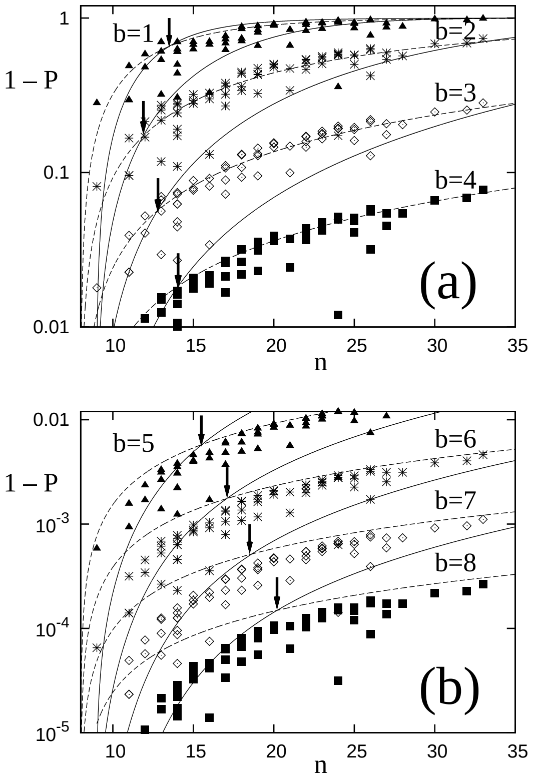

FIG. 5: Small-n behavior of $1 - P$ [see (65)] for several sample semiprimes N (plot symbols) with proper average over $\{\omega(N)\}$. The bandwidth b ranges from $b = 1$ to $b = 8$. (a) $b = 1$ (triangles), $b = 2$ (stars), $b = 3$ (diamonds), and $b = 4$ (squares). (b) $b = 5$ (triangles), $b = 6$ (stars), $b = 7$ (diamonds), and $b = 8$ (squares). The solid lines are the non-exponential fit functions (67). The dashed lines are the fit functions (66). The cross-over points between the small-n, non-exponential behavior and the large-n, exponential behavior [i.e. the intersections of (66) and (67)] are marked by arrows.

and are well approximated by the fit lines defined in (66). This illustrates that the b and n scaling in (66) holds over a considerable range of b and n values.

While on the large scale of Fig. 4 the data show an exponential behavior, looking more closely at the small-n regime, we see definite deviations from exponential behavior. Plotting $1 - P(n, b)$ magnifies the $P(n, b)$ behavior in the small-n region and clearly brings out the deviations from exponential behavior. This is illustrated in Fig. 5, which shows the data of Fig. 4, plotted as $1 - P(n, b)$. The dashed lines in Fig. 5 are the exponential fit lines defined in (66). We see that even on this magnified scale and in the large-n regime the data are well represented by the exponentials (66). For small n, however, the data clearly deviate from exponential, but are well fit by the solid lines representing the function [16]

$$P_<(n,b) = \tilde{P}_<(n,b)/\bar{f}, \tag{67}$$

where

$$\bar{f} = \int_{-1/2}^{1/2} \frac{\sin^2(\pi\beta)}{(\pi\beta)^2} d\beta \approx 0.774 \tag{68}$$

and

$$\tilde{P}_<(n,b) = \langle\frac{1}{r}\rangle + \Big(1 - \langle\frac{1}{r}\rangle\Big)\left(\frac{\bar{f} - \langle\frac{1}{r}\rangle}{1 - \langle\frac{1}{r}\rangle}\right) \exp\left[-\varphi_{max}^2(n,b)/100\right], \tag{69}$$

where φ_{max} is given in (54), r is defined in (62), and $\langle\frac{1}{r}\rangle = 2^{-(n-8)/2.6}$ (see Appendix C). Based on our numerical evidence, we conclude that $P(n,b)$ shows a clear transition from non-exponential behavior for small n to exponential behavior for large n. The arrows in Fig. 5 point to the locations of the transition between the two regimes and are the intersection points between the functions defined in (66) and (67).

Combining expressions (66) and (67), we derive an analytical expression, $n_t(b)$, for the transition points between the two different regimes for given b. The transition points n_t are defined as the n-value at which (66) equals (67). A useful analytical formula, approximately valid for $b \gtrsim 8$, is obtained in the following way. For $b \gtrsim 8$, we noticed numerically that the $1/r$ terms in (69) may be neglected, resulting only in a small shift of n_t of about 2 units in n. Therefore, to lowest order, $P_<(n_t,b) = P_>(n_t,b)$ results in

$$\frac{\varphi_{max}^2(n_t,b)}{100} = \xi_b \ln(2)(n_t - 8), \tag{70}$$

which implies

$$1.1 \times 2^{-2b} \ln(2)(n_t - 8) = \frac{4\pi^2}{100}\left[2^{-b-1}(n_t - b - 2) + 2^{-n_t}\right]^2. \tag{71}$$

At this point we notice that the transitions n_t between the two regimes occur at n values for which

$$2^{-n_t} \ll 2^{-b}, \tag{72}$$

which implies that we can safely neglect the 2^{-n_t} term in (71). This turns (71) into the quadratic equation

$$n_t^2 - 2n_t(C + b + 2) + 16C + (b+2)^2 = 0, \tag{73}$$

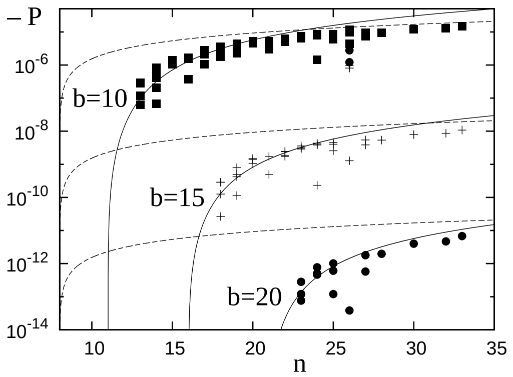

FIG. 6: Small-n behavior of semiprimes N for $b = 10$ (squares), $b = 15$ (crosses), and $b = 20$ (bullets). The full lines are the non-exponential performance functions $P_<(n, b)$ [see (67)]. The dashed lines are the corresponding large-n, exponential fit functions (66).

where we defined

$$C = \frac{55 \ln(2)}{\pi^2}. \tag{74}$$

Solving (73) yields

$$n_t = b + 5.9 + \sqrt{7.7(b+2) - 47}. \tag{75}$$

The expression (75) for the transition points shows that the onset of exponential behavior is shifted toward larger n for larger b. Formula (75) for the transition points $n_t(b)$ is useful for extrapolating into the practically relevant qubit regime $n \gtrsim 4000$, where classical computers cannot follow any more. In this classically inaccessible regime, we can then decide on the basis of (75), e.g., whether for given b and very large n, formula (66) or formula (67) should be used to predict the performance of the quantum computer. For $b = 1, \ldots, 4$, as shown in Fig. 5 (a), the transition is poorly defined, whereas, as shown in Fig. 5 (b), the transition is progressively better defined as b increases. That this trend continues is shown in Fig. 6, which shows data for $b = 10$, 15, and 20. We also see that the quality of the fit of the data with (67) improves for increasing b. The sharp cut-off displayed by $P_<(n, b)$ in Fig. 6 at $n = 11$ ($b = 10$), $n = 16$ ($b = 15$), and $n = 22$ ($b = 20$) is also understood since, according to (54), $\varphi_{max}(n, b) = 0$ for $n = b + 1$.

VI. ANALYTICAL RESULTS

Our analytical investigation of the performance measure starts with (35). Analytically and numerically we found that $\Phi(n, s_k, l_j)$ is a slow function of k, whereas $\varphi(n, b, s_k, l_j)$ is a fast, erratic function of k. Therefore, we can write approximately,

$$\begin{aligned}\tilde{P}_j(n, b, \omega) &\approx \frac{1}{2^n K}\left|\left[\sum_{k=0}^{K-1} e^{i\Phi(n,s_k,l_j)}\right]\langle e^{-i\varphi}\rangle_{n,b,l_j}\right|^2 \\ &= \frac{1}{2^n K}\left|\Omega(n, l_j, \omega)\right|^2\left|\langle e^{-i\varphi}\rangle_{n,b,l_j}\right|^2, \qquad (76)\end{aligned}$$

where $\Omega(n, l_j, \omega)$ is defined in (48) and

$$\langle e^{-i\varphi}\rangle_{n,b,l_j} = \frac{1}{K}\sum_{k=0}^{K-1} e^{-i\varphi(n,b,s_k,l_j)}. \qquad (77)$$

With (22), (23), and (51) we now obtain

$$P(n, b, \omega) = \frac{\sum_{j=0}^{\omega-1}\left|\Omega(n, l_j, \omega)\right|^2\left|\langle e^{-i\varphi}\rangle_{n,b,l_j}\right|^2}{\sum_{j=0}^{\omega-1}\left|\Omega(n, l_j, \omega)\right|^2}. \qquad (78)$$

We now proceed with a slightly less but still extremely accurate approximation by separating (78) in j, which then yields

$$P(n, b, \omega) = \frac{1}{\omega}\sum_{j=0}^{\omega-1}\left|\langle e^{-i\varphi}\rangle_{n,b,l_j}\right|^2 = \langle\left|\langle e^{-i\varphi}\rangle_k\right|^2\rangle_j, \qquad (79)$$

where $\langle\ldots\rangle_k$ and $\langle\ldots\rangle_j$ are averages over k and j, respectively. This expression for the performance measure $P(n, b, \omega)$ is the basis of our analytical work.

Since (79) is based on the validity of the separation in k and j, both are investigated in detail in Sec. VI A. A random model is used in Sec. VI B to evaluate (79) analytically in the large-n regime. This yields an analytical explanation for the b-scaling in (66) and excellent agreement with the prefactor of the exponential term in (66). In Sec. VI C, again assuming separation in k and j, we then arrive at an analytical formula describing the small-n regime, which predicts the functional form and the b-scaling of (67) very well, and also provides an estimate of the overall scaling factor.

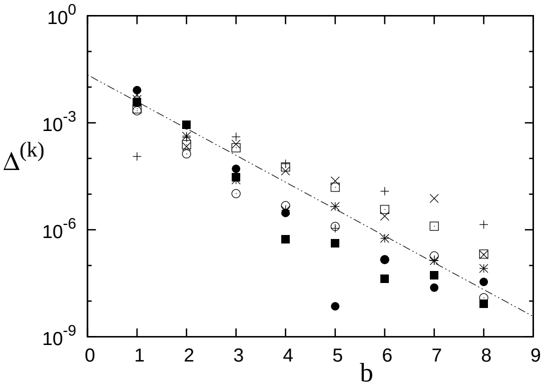

FIG. 7: Relative error $\Delta^{(k)}$ of k separation as a function of b for several semiprimes N. The data shows that the error is negligible. The fit line $\Delta = 2^{-2.5b-5.5}$ (dashed line) shows that the relative error vanishes exponentially in b.

A. Separability

In this section we investigate in detail the quality of the separations in k and in j, which lead to our jump-off point (79) for the analytical calculations reported in Sec. VI B and Sec. VI C.

We start with justifying the separation in k. To this end we define

$$A^{(k)} = \sum_{j=0}^{\omega-1} \left| \sum_{k=0}^{K-1} e^{i\Phi(n,s_k,l_j)-i\varphi(n,b,s_k,l_j)} \right|^2 \tag{80}$$

and

$$\begin{aligned} B^{(k)} &= \sum_{j=0}^{\omega-1} \left| \left[\sum_{k=0}^{K-1} e^{i\Phi(n,s_k,l_j)} \right] \frac{1}{K} \sum_{k'=0}^{K-1} e^{-i\varphi(n,b,s_{k'},l_j)} \right|^2 \\ &= \sum_{j=0}^{\omega-1} \left| \Omega(n,l_j,\omega) \right|^2 \left| \langle e^{-i\varphi} \rangle_{n,b,l_j} \right|^2 \end{aligned} \tag{81}$$

and compute the relative error

$$\Delta^{(k)} = \frac{\left| A^{(k)} - B^{(k)} \right|}{\left| A^{(k)} \right|} \tag{82}$$

incurred by the k separation. Figure 7 shows $\Delta^{(k)}$ as a function of b for various choices of N. We clearly see that k separation is an excellent approximation, which produces negligible, exponentially small errors. We plotted the line $\Delta = 2^{-2.5b-5.5}$ through the data to guide the eye. This line shows that the relative error of k separation vanishes exponentially in b.

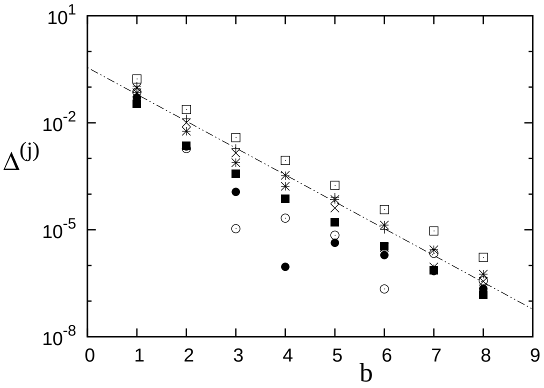

FIG. 8: Relative error $\Delta^{(j)}$ of j separation as a function of b for several semiprimes N. A fit line, $\Delta = 2^{-2.5b-1.5}$ (dashed line) is also shown. Compared with k separation (see Fig. 7) the error decays with the same exponent, only the overall scale factor is different.

Turning now to the j separation, we define

$$A^{(j)} = B^{(k)} \tag{83}$$

and

$$B^{(j)} = \left[\sum_{j=0}^{\omega-1} \left|\Omega(n, l_j, \omega)\right|^2\right] \frac{1}{\omega} \sum_{j=0}^{\omega-1} \left|\langle e^{-i\varphi}\rangle_{n,b,l_j}\right|^2 \tag{84}$$

and compute the relative error of j separation

$$\Delta^{(j)} = \frac{\left|A^{(j)} - B^{(j)}\right|}{\left|A^{(j)}\right|}. \tag{85}$$

Figure 8 shows $\Delta^{(j)}$ as a function of b for various choices of N. Apparently, while a bit less accurate than k separation, j separation is still highly accurate, improving exponentially with b. This is seen from the fit line $\Delta = 2^{-2.5b-1.5}$ through the data in Fig. 8, which also shows that $\Delta^{(k)}$ and $\Delta^{(j)}$ decay with the same exponential factor in b, and are offset by a constant only.

B. Large-n, exponential regime

In this section we evaluate (79) analytically in a model in which we treat s_k and l_j as independent random variables. This model, obviously, cannot capture the correlations between s_k and l_j introduced by ω and yields $P(n, b, \omega)$ that is independent of ω. Therefore, the ω-average in (65) is trivial and $P_N(n,b)$ does not depend on N either. Therefore, we write $P_N(n,b) \to P(n,b)$ as the prediction of the random model. However, even in this model, where ω-correlations are entirely neglected, it is hard to evaluate the expectation value of the exponential. Therefore, we proceed to evaluate (79) via its moment expansion

$$\langle \left|\langle e^{-i\varphi}\rangle_k\right|^2\rangle_j = 1 - \left[\langle\varphi^2\rangle_{kj} - \langle\langle\varphi\rangle_k^2\rangle_j\right] + \left[\frac{1}{12}\langle\varphi^4\rangle_{kj} + \frac{1}{4}\langle\langle\varphi^2\rangle_k^2\rangle_j - \frac{1}{3}\langle\langle\varphi\rangle_k\langle\varphi^3\rangle_k\rangle_j\right] \pm \ldots, \tag{86}$$

where we used $\langle\ldots\rangle_{kj} = \langle\langle\ldots\rangle_k\rangle_j = \langle\langle\ldots\rangle_j\rangle_k$ in cases where the averages commute. We start by computing

$$\langle\varphi^2\rangle_{kj} = \pi^2 \sum_{m,m'=b+1}^{n-1} \sum_{\mu=0}^{m-b-1} \sum_{\mu'=0}^{m'-b-1} \frac{\langle s_{[n-m-1]} s_{[n-m'-1]}\rangle_k \langle l_{[\mu]} l_{[\mu']}\rangle_j}{2^{m+m'-\mu-\mu'}}, \tag{87}$$

where we made use of the assumed independence of s and l. Taking into account that the binary digits of s and l can only take the values 0 and 1, we obtain

$$\langle s_{[\alpha]} s_{[\beta]}\rangle_k = \frac{1}{2}\delta_{\alpha\beta} + \frac{1}{4}(1-\delta_{\alpha\beta}), \tag{88}$$

and a similar expression for $\langle l_{[\mu]} l_{[\mu']}\rangle_j$. Because of (88), the evaluation of the quadruple sum (87) is lengthy, but can be performed analytically. The result is

$$\langle\varphi^2\rangle_{kj} = \left(\frac{\pi^2}{144}\right) 2^{-2b}\Big[9x^2 + 21x - 10 + 9(2+x)2^{-x} + 2^{-2x}\Big], \tag{89}$$

where

$$x = n - b - 2. \tag{90}$$

Next, we evaluate $\langle\langle\varphi\rangle_k^2\rangle_j$. With (88) and following the same procedures that lead to (89), we obtain

$$\langle\langle\varphi\rangle_k^2\rangle_j = \left(\frac{\pi^2}{96}\right) 2^{-2b}\Big[6x^2 + 6x - 4 + 6(1+x)2^{-x} + 2^{-2x}\Big], \tag{91}$$

where x is defined in (90). We define

$$\hat{\sigma}^2 = \langle\varphi^2\rangle_{kj} - \langle\langle\varphi\rangle_k^2\rangle_j, \tag{92}$$

which, on the basis of the results (89) and (91), is explicitly given by

$$\hat{\sigma}^2 = \left(\frac{\pi^2}{288}\right) 2^{-2b}\Big(24x - 8 + 18\times 2^{-x} - 2^{-2x}\Big). \tag{93}$$

With (79) and up to second order in the moment expansion (86), the performance measure is now given by

$$P(n,b) \approx 1 - \hat{\sigma}^2. \tag{94}$$

Comparing (94) with the fit function (66) and using (90), we see that (94), to leading order in n, is the first-order expansion of

$$P^{(a)}(n,b) \sim 2^{-\xi_b^{(a)} n}, \tag{95}$$

where

$$\xi_b^{(a)} = \left[\frac{\pi^2}{12\ln(2)}\right] \times 2^{-2b} \approx 1.19 \times 2^{-2b}. \tag{96}$$

This analytical result recovers the 2^{-2b} scaling of the fit line (66), and is within 10% of the exponential prefactor in (66).

The analytical evaluation of the 4th order terms in (86) is technically straightforward, but tedious, and not essential at this point. Our numerical calculations show that the 4th order terms are approximately given by $(\hat{\sigma}^2)^2/2$, and are therefore very small. This has two consequences: (i) it shows that up to 4th order in φ the probability measure $P(n,b)$ for fixed b is consistent with exponential decay in n and (ii) that because of their smallness it is currently not necessary to evaluate the 4th order terms analytically.

To conclude this section, we compute

$$\langle\varphi\rangle_{kj} = \frac{\pi}{4} \sum_{m=b+1}^{n-1} \sum_{\mu=0}^{m-b-1} \frac{1}{2^{m-\mu}}, \tag{97}$$

which is needed in the following section. Using the summation formula for the evaluation of geometric sums, we obtain

$$\langle\varphi\rangle_{kj} = \frac{\pi}{4}[2^{-b}(n-b-2) + 2^{1-n}] = \frac{1}{4}\varphi_{max}, \tag{98}$$

where we related $\langle\varphi\rangle_{kj}$ to φ_{max} via (54).

C. Small-n, non-exponential regime

Our starting point is again equation (79), but in this section we focus on the small-n regime, i.e. $n < n_t(b)$ [see (75)]. We first derive some useful relations that can then be used to evaluate (79) approximately in this regime. We start by inspecting $\varphi(n,b,s,l)$ in (37). We notice that

$$\varphi(n,b,s,l) = \frac{\pi}{2^{n-1}} \sum_{i=0}^{n-b-2} \left[(2^i s_{[i]} l) \mod 2^{n-b-1}\right]. \tag{99}$$

Since the modulus of the product of two numbers is smaller than or equal to the product of the moduli of two numbers, we obtain

$$\begin{aligned}\varphi(n,b,s,l) &\leq \frac{\pi}{2^{n-1}} \sum_{i=0}^{n-b-2} \left[(2^i s_{[i]} \bmod 2^{n-b-1})(l \bmod 2^{n-b-1})\right] \\ &= \frac{\pi}{2^{n-1}} \left[(s \bmod 2^{n-b-1})(l \bmod 2^{n-b-1})\right], \end{aligned} \tag{100}$$

where the equality is obtained by using

$$\left(\sum_{i=0}^{n-b-2} 2^i s_{[i]}\right) \bmod 2^{n-b-1} = (s \bmod 2^{n-b-1}) \bmod 2^{n-b-1} = s \bmod 2^{n-b-1}. \tag{101}$$

In order to compensate for the difference between (99) and (100), we introduce an effective parameter $\bar{l}$ in (100) such that

$$\varphi = \frac{\pi}{2^{n-1}} (s \bmod 2^{n-b-1}) \bar{l} \leq \varphi_{max}, \tag{102}$$

where the inequality is obtained from the definition of φ_{max} in (52). Since this inequality must hold for any s, the inequality (102) implies

$$\pi 2^{-b} \bar{l} < \varphi_{max}, \tag{103}$$

where we used $\max(s \bmod 2^{n-b-1}) \approx 2^{n-b-1}$. Assuming the random model used in Sec. VI B, in particular its assumption of statistical independence of s and l, we compute the average of (102). With (98) we obtain

$$\langle \varphi \rangle_{kj} = \frac{\varphi_{max}}{4} = \frac{\pi}{2^{n-1}} \langle s \bmod 2^{n-b-1} \rangle_k \langle \bar{l} \rangle_j = \frac{\pi}{2} 2^{-b} \langle \bar{l} \rangle_j. \tag{104}$$

Hence, solving for $\langle \bar{l} \rangle_j$, dropping the small term 2^{-n} in (54), we expect

$$\langle \bar{l} \rangle_j \simeq \frac{n-b-2}{2}. \tag{105}$$

We note that $\langle \bar{l} \rangle_j$ in (105) fulfills (103). Next, by writing the order of a seed as $\omega = 2^\alpha r$ [see (62)], and by using the form of an element s_k of an equivalence class $[s_0]$ defined in (8), we obtain

$$\begin{aligned} s_k \bmod 2^{n-b-1} &= kr 2^\alpha \bmod 2^{n-b-1} \\ &= (kr \bmod 2^{n-\alpha-b-1}) 2^\alpha, \end{aligned} \tag{106}$$

where we assumed $s_0 = 0$ for analytical simplicity. We note that $(kr \bmod 2^{n-\alpha-b-1})$ is a random integer variable in k for k an integer, which spans the entire integer space $0 \leq k \leq 2^{n-\alpha-b-1} - 1$. Now, we compute $\frac{\varphi}{\varphi_{max}}$, using (54), (102), and (106):

$$\begin{aligned}\frac{\varphi(n,b,s_k,l)}{\varphi_{max}} &= \frac{\pi}{2^{n-1}} \frac{(s_k \bmod 2^{n-b-1}) \times \bar{l}}{2\pi[2^{-b-1}(n-b) - 2^{-b} + 2^{-n}]} \\ &\approx \frac{\bar{l}}{n-b-2} \frac{kr \bmod 2^{n-\alpha-b-1}}{2^{n-\alpha-b-1}},\end{aligned} \tag{107}$$

where we again dropped the small 2^{-n} term. Thus, we write

$$\varphi(n,b,s_k,l) \approx \frac{\bar{l}\varphi_{max}}{n-b-2}\bar{R}_k, \tag{108}$$

where we used

$$\bar{R}_k = \frac{kr \bmod 2^{n-\alpha-b-1}}{2^{n-\alpha-b-1}}, \tag{109}$$

which is a random variable in k whose range is $[0,1)$.

We are now ready to evaluate (79). Inserting (108) in (79), we obtain

$$P(n,b) = \langle |\langle \exp\left(-i\bar{R}_k \frac{\varphi_{max}\bar{l}}{n-b-2}\right)\rangle_k|^2\rangle_j. \tag{110}$$

Assuming that $\bar{R}_k$ is uniformly distributed in $[0,1)$, we turn the k average into an integral and obtain

$$P(n,b) \approx \langle \left| \int_0^\eta e^{-i\bar{R}} \frac{1}{\eta} d\bar{R} \right|^2 \rangle_j, \tag{111}$$

where we defined

$$\eta = \frac{\bar{l}\varphi_{max}}{n-b-2}. \tag{112}$$

Evaluation of (111) yields

$$P(n,b) \approx \langle \frac{2}{\eta^2}[1 - \cos(\eta)]\rangle_j. \tag{113}$$

Since η defined in (112) is small for $n < n_t$, we Taylor-expand (113), which results in

$$P(n,b) \approx \langle \frac{2}{\eta^2}\left[1 - \left(1 - \frac{\eta^2}{2} + \frac{\eta^4}{24}\right)\right]\rangle_j = 1 - \frac{\langle \eta^2 \rangle_j}{12}. \tag{114}$$

Inserting η defined in (112) into (114), we obtain

$$P(n,b) \approx 1 - \frac{\varphi_{max}^2 \langle \bar{l}^2 \rangle_j}{12(n-b-2)^2}. \tag{115}$$

We compute $\langle \bar{l}^2 \rangle_j$ in the following way. Computing the average of the square of (102), we obtain

$$\begin{aligned}\langle \varphi^2 \rangle_{kj} &= \frac{\pi^2}{2^{2n-2}} \langle (s \bmod 2^{n-b-1})^2 \rangle_k \langle \bar{l}^2 \rangle_j \\ &= \left(\frac{\pi^2}{3}\right) 2^{-2b} \langle \bar{l}^2 \rangle_j, \end{aligned} \tag{116}$$

where we used the assumed independence of s and l of the random model. According to (89), and to leading order in x [defined in (90)], we have

$$\langle \varphi^2 \rangle_{kj} \approx \left(\frac{\pi^2}{16}\right) 2^{-2b} (n-b-2)^2. \tag{117}$$

Equating (116) and (117), we obtain

$$\langle \bar{l}^2 \rangle_j = \frac{3}{16} (n-b-2)^2. \tag{118}$$

Inserting (118) into (115), we obtain

$$P(n,b) \approx 1 - \frac{\varphi_{max}^2}{64} \approx \exp[-\varphi_{max}^2(n,b)/64]. \tag{119}$$

Compared with the numerical fit line (67) [in particular equation (69)], this analytical result predicts the functional form of the b-scaling exactly and the overall scaling factor within a factor of 2.

VII. COMPARISON WITH THE WORK OF FOWLER AND HOLLENBERG

Our work is closely related to the work of Fowler and Hollenberg [15] (in the following abbreviated to FH). The purpose of this section is to discuss similarities and differences between the two approaches. The notation in [15] differs from ours. In order to avoid confusion, we translate the notation in [15] into our notation. As argued in [15] and here, because of the sensitivity of quantum gates to noise and decoherence, it is important to reduce the number of gates and gate operations as much as possible. This provides the motivation for studying the performance of Shor's algorithm as a function of bandwidth b of the quantum Fourier transform, since a small b results in substantial savings in gates to be implemented and gate operations to be executed. Both works conclude that for large n the period-finding part of Shor's algorithm scales exponentially in n, $P(n,b) \sim 2^{-\xi_b n}$, where $\xi_b = \gamma 2^{-2b}$ and γ a constant. FH quote $\gamma = 2$; we find $\gamma = 1.1$. Thus, while the research

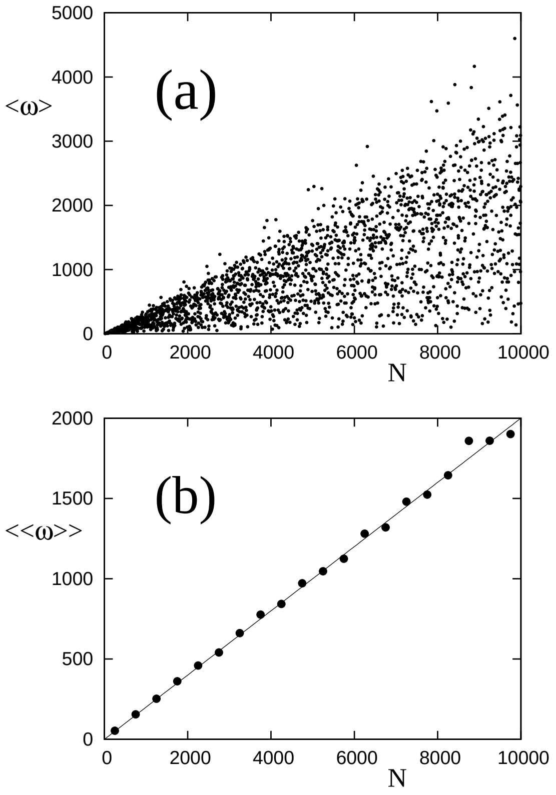

FIG. 9: Average ω as a function of N. (a) Scatter plot of $\langle\omega\rangle$ defined according to (120); (b) double averaged, binned $\langle\langle\omega\rangle\rangle$ defined according to (121).

goals are the same, and the central results are similar, there are substantial differences in how the research programs are executed, and there are new findings in our work. Among the new findings is the existence of a non-exponential regime for small n (see Sec. V), analytical results for the non-exponential and exponential regimes (see Sec. VI) and the existence of a provable bound for the maximal possible period ω of a given semiprime N (see Appendix B).

The main difference between [15] and our work concerns the choice of ω in the simulations. While in our work we simulate the period-finding part of Shor's algorithm for actual semiprimes N, and actual, associated ω values, FH use an effective $\omega = 2 + N/2$. Thus, our calculations are more realistic than those reported in [15] and check and complement the calculations in [15] under more realistic conditions. A first comment in this connection concerns the choice of FH's effective ω value. It was chosen as a good representative of ω values in Fig. 5 of [15]. However, the ω values in this figure extend up to $\omega = N$, which is more than twice larger than the maximal possible ω, which is smaller than $N/2$ (see Appendix B for the proof). Therefore, rather than located in the middle of Fig. 5 of [15], FH's effective ω actually lies beyond the allowed range of ω. However, this is not expected to make any difference in the conclusions of [15], since, as shown in Fig. 5 of [15], according to the simulations reported in [15], $P(n, b)$ exhibits flat plateaus in ω.

In this connection it may be interesting to present more information on the distribution of allowed ω values. In Fig. 9 (a) we show the properly averaged ω values,

$$\langle\omega\rangle = \frac{1}{\varphi_E(N)} \sum_{j=1}^{a(N)} \nu(\omega_j)\omega_j \tag{120}$$

as a function of N in the form of a scatter plot. The symbols in (120) have the same meaning as explained in connection with (65), i.e. $\varphi_E(N)$ is Euler's totient function, $a(N)$ is the number of ω values for given N, and $\nu(\omega)$ is the multiplicity of ω. We see that $\langle\omega\rangle$ is a sensitive function of N with a large spread over the entire allowed $\langle\omega\rangle$ range, i.e. $2 \leq \langle\omega\rangle < N/2$. To make more sense of the raw $\langle\omega\rangle$ data, Fig. 9 (b) shows a binned average of the $\langle\omega\rangle$ data in Fig. 9 (a) defined as

$$\langle\langle\omega\rangle\rangle(N^{(i)}) = \frac{1}{\chi(N^{(i)}+250) - \chi(N^{(i)}-250)} \sum_{\lambda=\chi(N^{(i)}-250)+1}^{\chi(N^{(i)}+250)} \langle\omega\rangle_\lambda,$$

$$N^{(i)} = 500\left(i - \frac{1}{2}\right), i = 1, \ldots, 20, \tag{121}$$

where $\chi(N)$ is the semiprime counting function and $\langle\omega\rangle_\lambda$ is the average ω [see (120)] associated with the λth semiprime. Figure 9 (b) shows that the twice averaged $\langle\langle\omega\rangle\rangle$ are linear in N with

$$\langle\langle\omega\rangle\rangle \approx N/5. \tag{122}$$

Therefore, according to Fig. 9 (b), a representative ω value for a given N is an allowed ω value in the vicinity of $N/5$.

In contrast to our choice of a single l state representing a Fourier peak, FH choose two l states to represent a Fourier peak, one to the left and one to the right of the position of the peak's maximum. This choice is more symmetrical than ours, but, because of the uniform response of all states under a Fourier peak (see Fig. 2 and the discussion in Sec. IV), one representative is sufficient.

FH quote $\gamma_{FH} = 2$ as a safe estimate, which is about a factor 2 larger than our, more optimistic, $\gamma = 1.1$. On the basis of the data in Fig. 6 of [15] we computed the actual γ_{FH} corresponding to the six panels of FH's Fig. 6, and obtained $\gamma_{FH} = 0.5$ $(b = 0)$, 1.85

($b = 1$), 1.83 ($b = 2$), 1.79 ($b = 3$), 1.78 ($b = 4$), 1.77 ($b = 5$), 1.73 ($b = 6$), and 1.57 ($b = 7$). Discarding the γ_{FH} value for $b = 0$ (it is not generic, since it involves only H and M gates and no rotation gate), and the γ_{FH} values for $b = 6$ and $b = 7$ (given the numerical range of the data, the exponential regime displayed in Fig. 6 of [15] is very short, resulting in uncertainty in the decay constant of an exponential fit), the γ_{FH} values are well characterized by $\gamma_{FH} \approx 1.8$, slightly more optimistic than the quoted $\gamma_{FH} = 2$. What is interesting for us is that $\gamma_{FH} = 1.8$ is already closer to our value of $\gamma = 1.1$.

Finally, what difference does it make for the performance of a quantum computer if $\gamma = 2$ or $\gamma = 1.1$? The answer depends on the performance level of the quantum computer. Since a factor 2 difference in γ is the difference between performance and the square of the performance, a factor 2 difference in gamma has basically no effect if the quantum computer operates with close to 100% performance, but has a *large* effect, if the quantum computer operates, e.g., on the 10% level.

Because of the critical need for quantum error correction and fault-tolerant operation [28], FH also present an error-tolerant, approximate construction of rotation gates, consisting of more fundamental elementary gates. In fact, each single-qubit rotation gate, as written in the quantum algorithm, may result in thousands of gates when decomposed. Unlike FH, we did not discuss the actual realization of gates, since, in this paper, we focus on the algorithmic aspects of Shor's algorithm, in particular on the scaling of the performance with n and b. In any case, as shown by FH, the actual experimental realization of fault-tolerant gates may require large numbers of additional, ancillary gates and qubits, motivating and emphasizing the critical need to reduce required quantum resources as much as possible by optimizing the quantum algorithms.

Given that error correction and fault-tolerant operation may introduce many additional auxiliary gates and qubits, what happens to our scaling laws in this case? Since our scaling laws depend on two parameters, b and n, the answer has two parts. (i) Error correction will not affect the b scaling, since the possibility of reducing the full quantum Fourier transform to a narrow-band quantum Fourier transform with bandwidth b is an intrinsic property of the mathematical structure of the Fourier transform itself that has nothing to do with quantum error correction. In fact, under noisy conditions, it may not even be a good idea to increase the bandwidth of the quantum Fourier transform, because the algorithmic accuracy of the transform gained might be more than offset by the errors introduced by the additional gates that are now exposed to noise and decoherence. (ii) It is clear that each computational qubit in Shor's algorithm has to be protected with quantum circuits that consist of additional qubits. However, since the scaling laws derived in this paper refer to the number n of computational qubits, our scaling laws remain unchanged.

Summarizing the discussion in this section, we see our work as complementary to the pioneering work of FH, adding new insights, and confirming the major conclusions of FH, using an independent approach based on period-finding simulations of actual semiprimes N, supported by analytical results.

VIII. DISCUSSION

An absolute limit of classical computing is reached when the physical requirements exceed the resources of the universe. According to this definition we can safely say that a classical computer, no matter its precise architecture, using the best currently available factoring algorithms, will never be able to factor a semiprime with 5000 decimal digits or more. We see this in the following way. The best currently known algorithm for factoring large, "hard" semiprimes (more than ~ 130 decimal digits; no small factors) is the general number field sieve (GNFS) [1]. It was recently used by Kleinjung *et al.* [8] to factor the RSA challenge number RSA-768 (232 decimal digits). This factorization took the equivalence of 2000 years on a 2.2 GHz Opteron workstation [8]. The performance of the GNFS scales approximately as [1]

$$P(N) \sim \exp\left\{1.9[\ln(N)]^{1/3}[\ln\ln(N)]^{2/3}\right\}, \tag{123}$$

where N is the semiprime to be factored. If we take the Kleinjung *et al.* factorization as the current, best benchmark, and estimate an Opteron processor to consist of roughly 10^{25} particles, then we can factor a 232-decimal-digit semiprime with $2000 \times 12 \times 10^{25} \approx 2 \times 10^{29}$ particles in the time span of a month. According to (123), then, in order to factor a 5000-decimal-digit number in the span of a month we need

$$2 \times 10^{29} \times P(10^{5000})/P(10^{232}) \approx 10^{89} \tag{124}$$

particles. This exceeds the number of particles in the universe ($\approx 10^{80}$) by several orders of magnitude. Clearly, the factorization of a 5000-decimal-digit semiprime is physically impossible to perform within a reasonable time ($\sim$ 1 month) on a classical computer. Even if we allow substantial progress in computer development, for instance replacing the current MOSFET transistors [29] used in computer chips by single-electron transistors [30] and increasing the clock-speed of a processor from 2.2 GHz to the optical regime of $\sim 10^{15}$ Hz, we gain only insignificantly. Therefore, in the absence of a breakthrough in the design of classical factoring algorithms, if we want to make any progress in factoring large numbers, we need a different computing paradigm. This is provided by switching from classical computing to quantum computing, i.e., running Shor's algorithm on a quantum computer. Instead of scaling (sub) exponentially, according to (123), Shor's algorithm scales $\sim O[(\ln N)^2(\ln\ln N)(\ln\ln\ln N)]$ [11] and thus provides an exponential speed-up that allows us, in principle, to tackle semiprimes vastly in excess of $N = 10^{5000}$. Obviously, for the practical implementation of powerful quantum computers, any optimization of quantum algorithms is welcome. Addressing this point, our paper shows that replacing the full quantum Fourier transform in Shor's algorithm with a narrow-band version incurs only a negligible performance penalty. We also show how the performance of such a streamlined version of Shor's algorithm scales with the number of qubits n.

In order to objectively characterize the performance of a quantum computer with n qubits, equipped with a banded quantum Fourier transform of bandwidth b, we defined the performance measure $P(n, b, \omega)$ in Sec. IV [see (22)]. This measure was carefully chosen to accurately reflect the performance of the quantum computer in terms of the probability of a successful factorization, yet not excessively expensive to compute numerically and, most importantly, a convenient starting point for analytical computations. As shown in Secs. V and VI, our performance measure fulfills both goals. Although any given peak in the quantum Fourier transform contains several l states with significant overlap with the Fourier peak, and useful for factorization in classical post-processing [10, 18], our performance measure defined in (22) is based only on a single l state, i.e. the state $|l_j\rangle$ closest to the central maximum of the Fourier peak number j [see (20)]. This, no doubt, is convenient for analytical calculations, as successfully demonstrated in Sec. VI, and for the following reason it is also justified. Numerically investigating the response of the Fourier peaks to a reduction of the bandwidth b, we found that the width of the Fourier peaks stays the same (about one state) while the height of the Fourier peaks is reduced. Thus, all l states under a Fourier peak respond in unison to a change in b (see Fig. 2), and since the width of the Fourier peaks stays the same, the number of significant states in a peak is conserved, too. This means that a single state under the peak, for instance, the state with maximal overlap, accurately represents the response of any other state under the peak, in particular the states useful for factorization. Thus, summarizing our choice of performance measure, we may say that, of course, choosing all those states under a Fourier peak that are useful for factorization, would be best. However, this is computationally prohibitively expensive and not useful for analytical calculations. A proxy is necessary. Because of the uniform response of all states in a Fourier peak, this proxy is provided, e.g., by the state closest to the central peak, $|l_j\rangle$, and leads directly to our performance measure $P(n, b)$ defined in (22).

The exponential fit function in (66) is shifted by 8 units in n. A possible explanation is the following. $n = 8$ corresponds to $N = 15$, the smallest odd semiprime. However, for $N = 15$ all possible orders ω are powers of 2. Therefore, according to the discussion in Sec. IV, Shor's algorithm performs perfectly in this case for all b. This means that $P(n = 8, b, \omega) = 1$ for all b, which is true independently of b only if ξ_b is multiplied with $n - 8$ in the exponent of (66).

The largest RSA challenge number [31] is RSA-2048. It has 2048 binary digits, which corresponds to 617 decimal digits. Factoring this number on a quantum computer requires a minimum of 4096 qubits. As an illustrative example, let us assume that we factor this number on a quantum computer with $b = 8$. Since no numerical simulation data are available in this very-large n regime, we have to rely on our results (66) and (67) to estimate the performance of the quantum computer. Which of the two formulas to use depends on which regime, exponential or non-exponential, we are in. For $b = 8$, and according to (75), the transition point n_t for $b = 8$ occurs at $n_t = 20$. Therefore, since $n \gg n_t$ in this case, we are sure that we are not in the non-exponential regime. However, how certain can we be that the exponential law (66) is valid all the way up to $n = 4096$, when we checked it numerically only up to $n \approx 30$ (see Sec. V)?

We answer this question in the following way. The moment expansion (86) is certainly valid out to n values for which our low-order Taylor expansion of $\exp(-i\varphi)$ is valid, i.e., for $\varphi < 1$. Since $\varphi < \varphi_{max}$, the safest estimate for the validity of (66) is $n \lesssim 2^{b+1}/(2\pi)$, which is obtained from (54) for $n \gg b$. For $b = 8$ this implies $n < 81$. This is already deeply in the n regime where current numerical simulations cannot follow. However, we can do better than that. The moment expansion (86), together with our numerical observation that the 4th order terms are given by $(\hat{\sigma}^2)^2/2$ shows that the relevant expansion parameter of (86) is not φ, but $\hat{\sigma}^2$, which is much smaller than φ_{max}^2. Therefore, we can safely assume exponential decay out to n values for which $\hat{\sigma}^2 < 1$. According to (93), then, this yields the estimate $n < 12 \times 2^{2b}/\pi^2$, which amounts to $n < 79682$ for $b = 8$, much larger than $n = 4096$ required for the factorization of RSA-2048. We conclude that, for $b = 8$, we may safely use the exponential law (66) to estimate the performance of the quantum computer. Therefore, using $n = 4096$ and $b = 8$ in (66), we obtain $P(n, b) = 0.954$, i.e. a quantum computer with a bandwidth of only $b = 8$ can factor the RSA challenge number RSA-2048 with a performance of better than 95%. If we increase $b = 8$ by only one unit to $b = 9$, the performance increases to 98%.

Concluding this section, we briefly discuss the paper by Barenco *et al.* [32], which also investigates the effect of the banded quantum Fourier transform on the performance of the period-finding part of Shor's algorithm. In fact, their performance measure Q, based on the probability of obtaining an $|l\rangle$-state closest to $2^n/\omega$, is, up to normalization, identical with our performance measure. However, the main focus of [32] is the effect of decoherence on Q and, similar to the work of Fowler and Hollenberg [15], Barenco *et al.* do not use factoring of actual semiprimes N in their numerical simulations. Finally, the analytical performance estimates in [32] require $b > \log_2(n) + 2$, which, for $b = 8$, implies $n < 64$. Therefore, for small $b \lesssim 8$, the analytical formulas of [32] are not applicable to the performance of a quantum computer in the technically and commercially interesting small-b, large-n regime with $n \gtrsim 4000$.

IX. SUMMARY AND CONCLUSIONS

Given that quantum computers are difficult to build, any advance in the optimization of quantum algorithms is welcome. Accordingly, in this paper, we investigated the performance of Shor's algorithm equipped with a banded quantum Fourier transform. Our predictions are based on the following five substantial advances.

1. Properly ω-averaged numerical simulations of factoring actual semiprimes N for qubit numbers ranging from $n = 9$ to $n = 33$, yielding the numerical performance estimates (66) in the large-n regime and (67) in the small-n regime.

2. Analytical and numerical justification of the separation of the k and j sums in the

definition of the performance measure as the foundation of analytical computations of the performance measure in the large-n and small-n regimes. It is shown that both separations are exponentially accurate, with exponential improvement of accuracy for increasing bandwidth b of the quantum Fourier transform.

3. Analytical computation of the performance measure in the exponential, high-n regime, which predicts the 2^{-2b} scaling exactly and the prefactor in ξ_b within 10% of the numerical result (66).

4. Analytical computation of the performance measure in the small-n regime, which predicts the functional form of the performance measure accurately and provides a reasonable estimate of a single, overall scaling factor.

5. Analytical formula (75) for the cross-over points n_t that mark the transition from the non-exponential regime to the exponential regime of quantum computer performance. For given bandwidth b and number of qubits n, this allows a quick, accurate, and convenient decision of whether the resulting finite-bandwidth quantum computer is working in the exponential or non-exponential regime.

In addition, in Appendix A, we prove the existence and uniqueness of an order-2 seed for any semiprime N, which, in Appendix B, is used to prove that the maximal possible order ω of a seed is less than $N/2$ (see Figs. 9 and 10). The maximally allowed ω is smaller than the effective, representative ω chosen in [15]. However, due to the insensitivity of the results in [15] with respect to the chosen ω (see Fig. 5 of [15]), this fact is not expected to change the results predicted in [15]. Lastly, we investigate the statistical properties of an inverse factor of ω in Appendix C.

In our opinion, and based on the numerical and analytical results presented in this paper, we conclude that the period-finding part of Shor's algorithm equipped with a banded quantum Fourier transform of bandwidth b is now essentially understood. However, period-finding is not the most demanding part of Shor's algorithm to implement. This distinction is reserved for the f-mapping part of Shor's algorithm (the modular exponentiation part), which feeds register II with $f(s)$ values (see Sec. II) and, compared with the period-finding part of Shor's algorithm, requires vastly more quantum resources to implement [25, 33–35]. Therefore, attention now has to be directed toward optimizing the f-mapping part of Shor's algorithm.

Appendix A: Existence and Uniqueness of an element of Order 2

In support of the result that the probability of encountering a seed with a small order is small, we provide here a proof that there is one and only one seed x of order 2 for any semi-prime $N = pq$, where $p \neq q$ are primes larger than 2. A seed is any positive integer, larger than 1, that is relatively prime to N. Let us collect all possible seeds x_j, $j = 1, \ldots, L-1$, including the unit 1, into a set $G_N = \{1, x_1, x_2, \ldots, x_{L-1}\}$. This way, G_N forms a multiplicative group modulo N [36] containing L elements.

The computation of L is straightforward. There are at most $N-1$ numbers that are relatively prime to $N = pq$. (By definition, the unit element 1 is relatively prime to N [27], but N is not.) However, $p-1$ of these numbers contain a factor q and $q-1$ of these numbers contain a factor p, and these numbers are all different. Therefore, there are $L = (N-1)-(p-1)-(q-1) = N-p-q+1$ group elements. Since N, p, and q are odd, L is even. At this point we cite a well-known theorem of elementary algebra that states that each group with an even number of elements has at least one element that is different from the unit element and is of order two [27]. Applied to our group G_N this means that there exists at least one seed $x \neq 1$ with $x^2 = 1$ modulo N, i.e. a seed of order 2.

At this point it is important to observe that if there is a seed x with $x^2 \mod N = 1$, then there is a mirror seed $z = N-x$, which is also of order 2, since $z^2 \mod N = (N^2-2Nx+x^2) \mod N = x^2 \mod N = 1$. Therefore, without restriction of generality, we will restrict ourselves to the range of seeds smaller than $N/2$ and prove that there is only one $x < N/2$ with $x^2 \mod N = 1$, where $N = pq$.

We already proved that there is at least one x with

$$x^2 \mod N = 1. \tag{A1}$$

Without restriction of generality, we can choose this x to be smaller than $N/2$, since, if it is larger than $N/2$, its mirror will be smaller than $N/2$. Assume that there exists another seed of order 2, $y < N/2$, with $y > x$ (no restriction of generality) and

$$y^2 \mod N = 1. \tag{A2}$$

Since $x^2 \mod N = 1$ and $y^2 \mod N = 1$, we have

$$(y^2 - x^2) \mod N = (y-x)(y+x) \mod N = 0. \tag{A3}$$

This equation holds if either (i) at least one of the factors is divisible by N or (ii) $(y-x)$ contains p and $(y+x)$ contains q, or vice versa. However, case (i) is impossible: Since both x and y are smaller than $N/2$, $(y+x) < N$ is, therefore, never divisible by N. For the same reason $(y-x)$ is divisible by N only if $(y-x) = 0$, which is excluded, since, according to assumption, $y \neq x$. This leaves case (ii).

Since $x^2 \mod N = 1$, we have $(x-1)(x+1) \mod N = 0$. Since $(x-1) < N$ and $(x+1) < N$, for any $N > 2$, neither factor is divisible by N and the product is divisible by N only if $(x-1)$ is a multiple of p and $(x+1)$ is a multiple of q. There is no restriction of generality here, since which factor of the product is divisible by which factor of N (p or q) is merely a matter of properly labeling the factors of N. So, let us write:

$$x - 1 = \lambda p, \tag{A4}$$

$$x + 1 = \mu q, \tag{A5}$$

where λ and μ are positive integers. We observe immediately that λ cannot contain a factor q, since otherwise $(x-1)$ would be divisible by N. In the same way we reason that μ cannot contain a factor p. We record this observation as

$$\lambda \mod q \neq 0, \tag{A6}$$

$$\mu \mod p \neq 0. \tag{A7}$$

We also have $y^2 \mod N = 1$, i.e. $(y-1)(y+1) \mod N = 0$, which now implies two possibilities, since in (A4) and (A5) we already chose the naming convention for the two factors p and q of N. The two cases are:

$$\text{(A)} \quad (y-1) \text{ is a multiple of } p,\ (y+1) \text{ is a multiple of } q \tag{A8}$$

$$\text{(B)} \quad (y-1) \text{ is a multiple of } q,\ (y+1) \text{ is a multiple of } p. \tag{A9}$$

Let us look at case (A) first. Let us write:

$$(y-1) = \alpha p, \tag{A10}$$

$$(y+1) = \beta q. \tag{A11}$$

In analogy with the reasoning that led us to (A6) and (A7) we have

$$\alpha \mod q \neq 0, \tag{A12}$$

$$\beta \mod p \neq 0. \tag{A13}$$

Then, because of $x, y < N/2$, (A3), and the discussion following (A3), we need to prove that either $(y - x)$ contains a factor p and $(y + x)$ a factor q or vice versa. We write:

$$y + x = (y - 1) + (x + 1) = \alpha p + \mu q. \tag{A14}$$

But since α is not divisible by q [see (A12)] and μ is not divisible by p [see (A7)], $(y + x)$ is neither divisible by p nor by q. Therefore, case (A) leads to a contradiction, which implies that according to case (A) a second order-2 seed $y \neq x$ does not exist.

Let us now look at case (B). Let us write:

$$(y - 1) = \gamma q, \tag{A15}$$

$$(y + 1) = \nu p, \tag{A16}$$

where, again, in analogy with the reasoning that led us to (A6) and (A7), we have

$$\gamma \mod p \neq 0, \tag{A17}$$

$$\nu \mod q \neq 0. \tag{A18}$$

Then:

$$y - x = (y - 1) - (x - 1) = \gamma q - \lambda p, \tag{A19}$$

which, because of (A17) and (A18) is neither divisible by p nor by q. Therefore, case (B), too, leads to a contradiction.

As a result, we obtain that the existence of an additional order-2 seed $y \neq x$, $y < N/2$ is impossible. Therefore, x is the unique order-2 seed with $x < N/2$. This means that for any given semi-prime $N = pq$, there are exactly two order-2 seeds, $x < N/2$ and its mirror $N - x > N/2$.

Appendix B: Maximal Order

In connection with Shor's algorithm, for a given semi-prime N, we consider seeds x with an *even* order $\omega = 2\Omega$, where $\Omega \geq 1$ is a positive integer. The purpose of this section is to show that the largest possible even ω is smaller than $N/2$.

A seed x, $1 \leq x < N$ is a positive integer, relatively prime to $N = pq$, where $p \neq q$ are prime numbers larger than 2. As discussed in Appendix A, the set of seeds x forms a group G_N with

$$|G_N| = N - p - q - 1 = (p-1)(q-1) \tag{B1}$$

elements. We note that, according to (B1), $|G_N|$ is divisible by 4, a fact which will become relevant below. If x is relatively prime to N, so is $N - x$. Therefore, if x is a seed, so is $N - x$, which implies (i) a symmetry of seeds with respect to $N/2$ and (ii) that there is an even number of seeds. We use (i) to define a set $\hat{G}_N$, consisting of elements $\hat{x} = (x, N - x)$, where x and $N - x$ are identified. The set $\hat{G}_N$ forms a group. This is so, since $\hat{G}_N$ contains the unit element $\hat{1} = (1, N - 1)$, the product $\hat{x}\hat{y}$ of two elements of $\hat{G}_N$ is again in $\hat{G}_N$, and with each $\hat{x}$, we also find its inverse $(\hat{x})^{-1}$ in $\hat{G}_N$. Because of (i) the group $\hat{G}_N$ has

$$|\hat{G}_N| = |G_N|/2 \tag{B2}$$

elements.

Let us form the set G_N^* that contains the squares of x modulo N. Since G_N^* contains the unit element 1, and since with each x^2 and y^2 in G_N^*, the product

$$(x^2)(y^2) \mod N = (xy)^2 \mod N \tag{B3}$$

is also in G_N^*, and since with each x^2 we also find its inverse

$$(x^2)^{-1} \mod N = (x^{-1})^2 \mod N \tag{B4}$$

in G_N^*, the set G_N^* is a group. In the same way we form the set $\hat{G}_N^*$ from the squares of $\hat{x}$ in $\hat{G}_N$. Because of the definition of $\hat{G}_N$, identifying x and $N - x$, and because of

$$(N - x)^2 \mod N = x^2 \mod N, \tag{B5}$$

which shows that the squares of x and $N - x$ are identical, the groups G_N^* and $\hat{G}_N^*$ have the same number of elements. In addition, as is easily verified, the groups G_N^* and $\hat{G}_N^*$ are isomorphic, which implies that the order of an element in $\hat{G}_N^*$ is the same as the order of an element in G_N^*. Let us denote the number of elements in these two groups by

$$|G_N^*| = |\hat{G}_N^*| = M. \tag{B6}$$

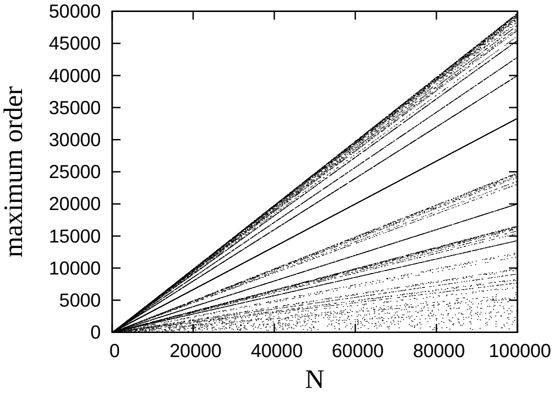

FIG. 10: Maximal possible orders ω (maximum order) computed and displayed for each N in a complete list of semiprimes in the interval $0 < N < 10^5$. Apparently, the maximal possible order never exceeds $N/2$, a fact proved in the text.

Then, because of (B2), and because $\hat{G}_N^*$ is a subgroup of $\hat{G}_N$, we have that

$$M = |\hat{G}_N^*| \text{ divides } |\hat{G}_N| = |G_N|/2. \tag{B7}$$

One possibility is $M = |G_N|/2$. However, since the group $\hat{G}_N^*$ of squares is a subgroup of $\hat{G}_N$, $M = |G_N|/2$ is possible only if there are as many squares $\hat{x}^2$ in $\hat{G}_N^*$ as there are elements $\hat{x}$ in $\hat{G}_N$. However, because of the existence of a non-trivial order-2 element $\hat{a}$ (see Appendix A), this is impossible, since both $\hat{1}^2 = \hat{1}$ and $\hat{a}^2 = \hat{1}$, which immediately implies $M < |G_N|/2$. Therefore, the largest possible M that divides $|G_N|/2$ (an even number) is $|G_N|/4$, which implies

$$M \leq |G_N|/4. \tag{B8}$$

According to Euler's totient theorem [27], we have for any $\hat{x}^2$ in $\hat{G}_N^*$:

$$(\hat{x}^2)^M = \hat{1}, \tag{B9}$$

which implies that the order of any element $\hat{x}^2$ in $\hat{G}_N^*$ is at most $M = |G_N|/4$. Because of the isomorphism between $\hat{G}_N^*$ and G_N^*, this implies that the order of any x^2 in G_N^* is at most $|G_N|/4$. This, finally, implies that the order of any element x in G_N is at most $|G_N|/2$, i.e.

$$\omega \leq |G_N|/2 < N/2. \tag{B10}$$

We note that since an essential element of the proof is to consider the group of squares of x, the proof indeed applies only to *even* ω. An illustration of (B10) is provided in Fig. 10,

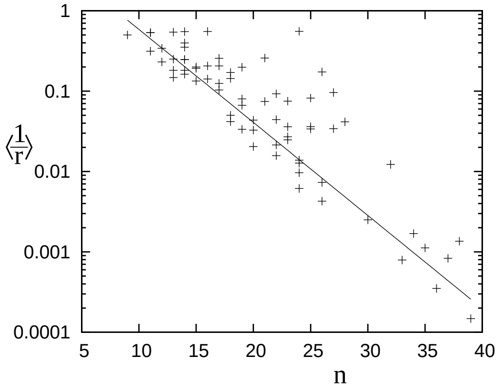

FIG. 11: The fraction $\langle\frac{1}{r}\rangle$ as a function of n for several semiprimes. The fit line (solid line) is the function $\langle\frac{1}{r}\rangle = 2^{-(n-8)/2.6}$.

which shows the maximum even orders of all semiprimes N ranging up to $N = 100000$. The figure illlustrates (i) that the maximal order is indeed smaller than $N/2$ and (ii) that the maximal order of a given semiprime N is not always close to $N/2$ but still has to divide the group order. Therefore, in addition to the line $\sim N/2$, we also see the lines corresponding to $\sim N/4$, $\sim N/6$, etc..

Appendix C: $1/r$ average

For the analytical formula (69), we need the average $\langle\frac{1}{r}\rangle$ of $1/r$ as a function of n, where r is defined in (62). We computed it in the following way. First, we computed all possible orders, ω_j, of a given semiprime N with their associated multiplicities, $\nu(\omega_j)$. Then, we extracted the odd part of the obtained orders, r, as defined in (62). Denoting the odd part of a specific order ω_j by r_j, in analogy with (65) and (120), we obtain

$$\langle\frac{1}{r}\rangle = \frac{1}{\varphi_E(N)} \sum_{j=1}^{a(N)} \nu(\omega_j)\frac{1}{r_j}, \tag{C1}$$

where the symbols in (C1) share the same definition as shown in (65) and (120), i.e. $\varphi_E(N)$ is Euler's totient function and $a(N)$ is the number of orders for given N. Figure 11 shows the computed $\langle\frac{1}{r}\rangle$ according to (C1) as a function of n, the number of qubits needed for a reliable determination of the order as described in connection with (64). By graphically extracting the n-dependence of $\langle \frac{1}{r} \rangle$ using the fit line in Fig. 11, we find

$$\langle \frac{1}{r} \rangle = 2^{-(n-8)/2.6}. \tag{C2}$$

[1] C. Pomerance, Notices Am. Math. Soc. **43**, 1473 (1996).

[2] R. Rivest, A. Shamir, and L. Adleman, Comm. ACM **21**, 120 (1978).

[3] D. Boneh, Notices Am. Math. Soc. **46**, 203 (1999).

[4] S. Robinson, SIAM News **36**(5) (2003).

[5] C. Pomerance, in *Computational Methods in Number Theory, Part I*, Math. Centre Tract, Vol. 154, edited by H. W. Lenstra, Jr. and R. Tijdeman, (Mathematisch Centrum, Amsterdam, 1982), pp. 89-139.

[6] R. D. Silverman, Math. Comput. **48**, 329 (1987).

[7] J. P. Buhler, H. W. Lenstra, Jr., C. Pomerance, in *The Development of the Number Field Sieve*, Lecture Notes in Mathematics Vol. 1554, edited by A. K. Lenstra and H. W. Lenstra, Jr. (Springer, New York, 1993), pp. 50-94.

[8] T. Kleinjung, K. Aoki, J. Franke, A. K. Lenstra, E. Thomé, J. W. Bos, P. Gaudry, A. Kruppa, P. L. Montgomery, D. A. Osvik, H. te Riele, A. Timofeev, and P. Zimmermann, in *CRYPTO'10 Proceedings of the 30th Annual Conference on Advances in Cryptology* (Springer, Berlin, 2010), pp. 333-350.

[9] E. Barker and A. Roginsky, NIST Special Publication 800-131A (2011).

[10] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).

[11] P. W. Shor, in *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science*, edited by S. Goldwasser (IEEE Press, Los Alamitos, CA, 1994), pp. 124-134.

[12] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Nature **414**, 883 (2001).

[13] E. Martín-López, A. Laing, T. Lawson, R. Alvarez, X.-Q. Zhou, and J. L. O'Brien, arXiv:1111.4147.

[14] D. Coppersmith, arXiv:quant-ph/0201067.

[15] A. G. Fowler and L. C. L. Hollenberg, Phys. Rev. A **70**, 032329 (2004).

[16] Y. S. Nam and R. Blümel, Phys. Rev. A **86**, 044303 (2012).

[17] R. Blümel, *Foundations of Quantum Mechanics—From Photons to Quantum Computers* (Jones and Bartlett, Sudbury, 2010).

[18] N. D. Mermin, *Quantum Computer Science* (Cambridge University Press, Cambridge, 2007).

[19] C.-Y. Lu, D. E. Browne, T. Yang, and J.-W. Pan, Phys. Rev. Lett. **99**, 250504 (2007).

[20] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. V. James, A. Gilchrist, and A. G. White, Phys. Rev. Lett. **99**, 250505 (2007).

[21] A. Politi, J. C. F. Matthews, J. L. O'Brien, Science **325**, 1221 (2009).

[22] R. B. Griffiths and C.-S. Niu, Phys. Rev. Lett. **76**, 3228 (1996).

[23] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in Fortran 77*, second edition (Cambridge University Press, Cambridge, 1992).

[24] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, 2nd Edition (Addison-Wesley, Reading, Massachusetts, 1994).

[25] A. Ekert and R. Jozsa, Rev. Mod. Phys. **68**, 733 (1996).

[26] P. W. Shor, arXiv:quant-ph/9508027v2.

[27] N. Jacobson, *Basic Algebra I* (Dover Publications, Mineola, New York, 2009).

[28] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, Phys. Rev. A **83**, 020302.

[29] R. G. Lerner and G. L. Trigg, *Encyclopedia of Physics*, second edition (VCH, New York, 1991).

[30] M. A. Kastner, Ann. Phys. (Leipzig) **9**, 885 (2000).

[31] http://www.rsa.com/rsalabs/node.asp?id=2093

[32] A. Barenco, A. Ekert, K.-A. Suominen, and P. Törmä, Phys. Rev. A **54**, 139 (1996).

[33] I. García-Mata, K. M. Frahm, and D. L. Shepelyansky, Phys. Rev. A **75**, 052311 (2007).

[34] V. Vedral, A. Barenco, and A. Ekert, Phys. Rev. A **54**, 147 (1996).

[35] R. Van Meter and K. M. Itoh, Phys. Rev. A **71**, 052320 (2005).

[36] M. Hazewinkel, N. Gubareni, and V. V. Kirichenko, *Algebras, Rings and Modules*, Volume 1 (Kluwer, Dordrecht, 2010).