Variational Quantum Factoring

Eric R. Anschuetz,* Jonathan P. Olson,† Alán Aspuru-Guzik,‡ and Yudong Cao§

Zapata Computing Inc., 501 Massachusetts Avenue, Cambridge MA 02138

Abstract

Integer factorization has been one of the cornerstone applications of the field of quantum computing since the discovery of an efficient algorithm for factoring by Peter Shor. Unfortunately, factoring via Shor's algorithm is well-beyond the capabilities of today's noisy intermediate-scale quantum (NISQ) devices. In this work, we revisit the problem of factoring, developing an alternative to Shor's algorithm, which employs established techniques to map the factoring problem to the ground state of an Ising Hamiltonian. The proposed variational quantum factoring (VQF) algorithm employs simple rules for Boolean equations. We implement a preprocessing step for reducing the number of qubits needed for the Hamiltonian. We then find an approximate ground state of the Ising Hamiltonian by training variational circuits using the quantum approximate optimization algorithm (QAOA). We benchmark the VQF algorithm on various instances of factoring and present numerical results on its performance. We also discuss the potential scaling challenges and opportunities for further improvement of the algorithm.

* eric.anschuetz@zapatacomputing.com

† jonny@zapatacomputing.com

‡ aspuru@zapatacomputing.com

§ yudong@zapatacomputing.com

I. INTRODUCTION

Integer factorization is one of the first practically-relevant problems that can be solved exponentially faster on a quantum computer than any currently-known methods for classical computation by employing an algorithm introduced by Peter Shor, and now known as Shor's algorithm [1]. Since its initial appearance, numerous follow-up studies have been carried out to optimize the implementation of Shor's algorithm from both algorithmic and experimental perspectives [2–11]. For an input number of n bits, circuit constructions involving $2n+3$ [9] and $2n+2$ qubits [12, 13], and further restricting to only nearest-neighbor interactions [14] have been proposed. Concrete resource estimates in realizing Shor's algorithm for factoring practically-relevant RSA numbers have also been performed for specific architectures [15–18]. For example, on one particular architecture of a fault-tolerant quantum computer [16, 17] it is estimated that factoring a 2048-bit RSA number requires a circuit depth on the order of 10^9, requiring roughly 10 days on a quantum computer comprised of 10^5 logical qubits [16, c.f. Fig. 15]. Another resource estimate [19] considering a photonic architecture suggests that factoring a 1024-bit RSA number would take 2.3 years with 1.9 billion photonic modules. In contrast, present technologies are in the era of noisy intermediate-scale quantum (NISQ) devices [20], where quantum devices typically have on the order of 10^2-10^3 noisy qubits that can only reliably implement circuits of limited depth. This renders the practical impact of Shor's algorithm (as well as alternative algorithms for quantum factoring that use subroutines requiring fault-tolerance, such as [11, 21]) a reality at least as distant as the realization of fault-tolerant quantum computers.

Another approach to factoring on a quantum computer exploits the mapping from factoring to the ground state problem of an Ising Hamiltonian [22]. The basic idea underlying the mapping is to simply use the fact that factoring is the inverse operation of multiplication. Therefore, by working through the multiplication of two undetermined n-bit numbers and fixing the output to be the number to be factored, one can write a set of equations relating the bits of the factors and the carry bits. The Hamiltonian is constructed such that the ground state satisfies all of the generated equations and any bit assignments which violate any of the equations receives an energy penalty. Interesting observations [8, 23, 24] have been made about specific instances of factoring which allow one to simplify the equations tremendously. On the experimental side, most of the current efforts focus on analog approaches

such as quantum annealing [25] and simulated adiabatic evolution [24]. However, the same ground state problem of Ising Hamiltonians can be approximately solved on gate model NISQ devices using the quantum approximate optimization algorithm (QAOA) [26, 27].

Here we introduce an approach which we call *variational quantum factoring* (VQF). As other hybrid quantum-classical algorithms such as the variational quantum eigensolver (VQE) [28], or the variational quantum autoencoder (QAE) [29], classical preprocessing coupled with quantum state preparation and measurement can be used to optimize a cost function. In particular, we employ the QAOA algorithm originally introduced by Farhi [27] and classical preprocessing for factoring. The VQF scheme has two main components: first, we map the factoring problem to an Ising Hamiltonian, using an automated program to find reduction in the number of qubits whenever appropriate. Then, we train the QAOA ansatz for the Hamiltonian using a combination of local and global optimization. We explore five instances of the factoring problem (namely, the factoring of 35, 77, 1207, 56153, and 291311) to demonstrate the effectiveness of our scheme as well as its robustness with respect to noise. The remainder of the paper is organized as follows. Section II describes the mapping from a factoring problem to an Ising Hamiltonian, together with the simplification scheme that is used for reducing the number of qubits needed. Section III introduces QAOA and describes our method for training the ansatz. Section IV presents the numerical results. We conclude in Section V with further discussion on open problems and future works.

II. ENCODING FACTORING INTO AN ISING HAMILTONIAN

It is known from previous work that factoring can be cast as the minimization of a cost function [22], which can then be encoded into the ground state of an Ising Hamiltonian [23, 30, 31]. To see this, consider the factoring of $m = p \cdot q$, each having binary representations,

$$\begin{aligned} m &= \sum_{k=0}^{n_m} 2^i m_k, \\ p &= \sum_{k=0}^{n_p} 2^i p_k, \\ q &= \sum_{k=0}^{n_q} 2^i q_k, \end{aligned} \tag{1}$$

where $m_k \in \{0,1\}$ is the kth bit of m, n_m is the number of bits of m, and similarly for p and q. When n_p and n_q are unknown (as they are *a priori* when only given a number m to factor), one may assume without loss of generality that $p \geq q$, $n_p = n_m$, and $n_q = \lceil \frac{n_m}{2} \rceil$ [22]. By carrying out binary multiplication, the bits representing m, p, and q must satisfy the set of equations,

$$0 = \sum_{j=0}^{i} q_j p_{i-j} + \sum_{j=0}^{i} z_{j,i} - m_i - \sum_{j=1}^{n_c} 2^j z_{i,i+j} \tag{2}$$

for all $0 \leq i \leq n_c$, where $z_{i,j} \in \{0,1\}$ represents the carry bit from bit position i into bit position j [22, 30, 31] and $n_c = n_p + n_q - 1$ is the number of equations that are necessary to specify Eq. 2. If we associate a clause C_i over $\mathbb{Z}$ with each equation such that,

$$C_i = \sum_{j=0}^{i} q_j p_{i-j} + \sum_{j=0}^{i} z_{j,i} - m_i - \sum_{j=1}^{n_c} 2^j z_{i,i+j}, \tag{3}$$

then factoring can be represented as finding the assignment of binary variables $\{p_i\}$ and $\{q_i\}$ which solves,

$$0 = \sum_{i=0}^{n_c} C_i^2. \tag{4}$$

This corresponds to the minimization of the classical energy function,

$$E = \sum_{i=0}^{n_c} C_i^2, \tag{5}$$

which has a natural quantum representation as a *factoring Hamiltonian*,

$$H = \sum_{i=0}^{n_c} \hat{C}_i^2, \tag{6}$$

where each bit assignment for p_i, q_i, and $z_{j,i}$ in the energy function is now quantized by taking the transformation,

$$b_k \to \frac{1}{2}\left(1 - \sigma_{b,k}^z\right), \tag{7}$$

where $b \in \{p, q, z\}$ and k is its associated index. We further note that H can be represented in quadratic form by substituting each product $q_j p_{i-j}$ with a new binary variable $w_{i,j}$ and adding additional constraints to the Hamiltonian [30]. Therefore, there is a natural encoding of a factoring instance into the ground state of an Ising Hamiltonian. If the number being factored is a biprime, analogous clauses to Eq. (3) that must not be satisfied can be derived by expanding a $(2+\ell)$-term product $m = p \cdot q \cdot \prod_{i=1}^{\ell} c^{(i)}$ for ℓ extraneous factors $c^{(i)}$. These

clauses can then be added as an energy penalty to Eq. (5) to raise the energy of some non-satisfying assignments and aid later steps in finding the ground state of the factoring Hamiltonian.

One method for reducing the resource requirements for finding the ground state of the Ising Hamiltonian is to directly solve for a subset of the binary variables in the optimization problem that are easy to solve for classically [23, 31]. This reduction iterates through all clauses C_i as given by Eq. (3) a constant number of times. In the following discussion, let $x, y \in \mathbb{F}_2$ be unknown variables, and $a, b, a_i \in \mathbb{Z}^+$ unknown constants (as they are either functions of m or are known through the classical preprocessing of C_j for $j < i$). Along with some trivial relations, we apply the classical preprocessing rules:

$$\begin{aligned} 0 = xy - 1 &\implies x = y = 1, \\ 0 = x + y - 1 &\implies xy = 0, \\ 0 = a - bx &\implies x = 1, \\ 0 = \sum_i a_i &\implies a_i = 0, \\ 0 = \sum_{i=1}^{n} a_i - n &\implies a_i = 1. \end{aligned} \tag{8}$$

This classical preprocessing iterates through each of $O(n_m)$ terms in each of $O(n_m)$ clauses, with a classical computer runtime of $O(n_m^2)$. In practice, for most instances we have observed this greatly reduces the number of qubits needed for the quantum solver portion of the algorithm, as is shown in Figure 1.

III. VARIATIONAL QUANTUM FACTORING ALGORITHM

The main component of our scheme is an approximate quantum ground state solver for the Hamiltonian in Eq. (6) as a means to approximately factor numbers on near-term gate model quantum computers. We use the *quantum approximate optimization algorithm* (QAOA), which is a hybrid classical/quantum algorithm for near-term quantum computers that approximately solves classical optimization problems [27]. The goal of the algorithm is to satisfy (i.e., find the simultaneous zeros of) clauses C_i over n variables, which is cast as the minimization of a classical cost Hamiltonian H_c which we set to be identical to the

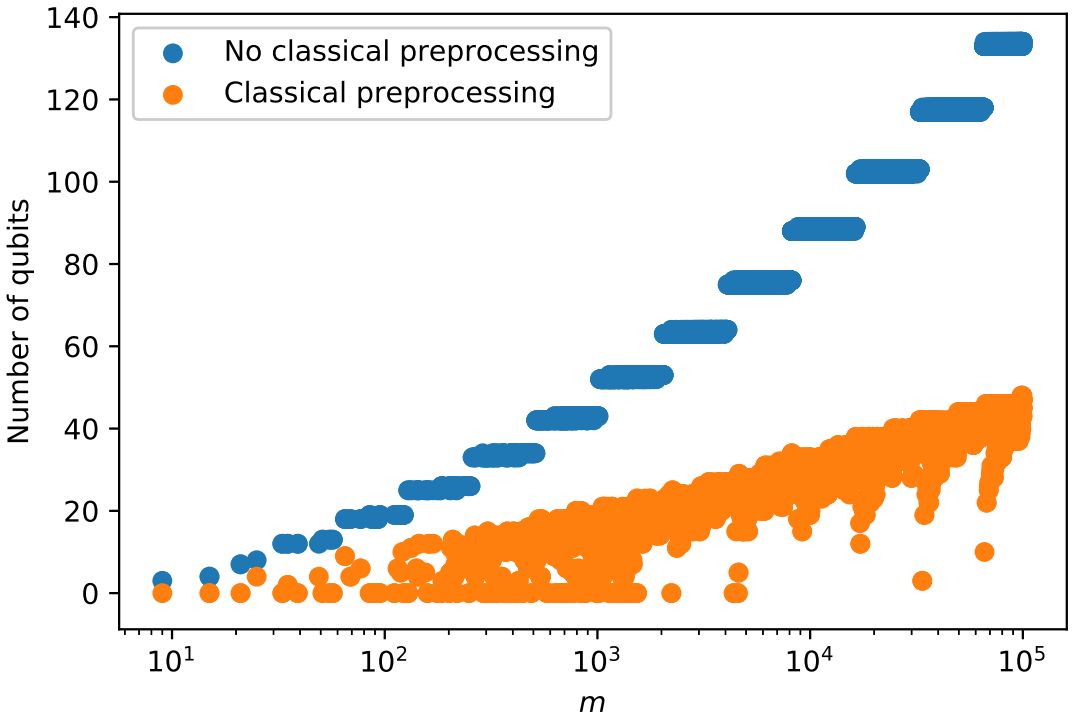


FIG. 1. This figure empirically demonstrates the reduction in qubit number using a classical preprocessing step. After the classical preprocessing algorithm (orange), the number of qubits necessary for our algorithm empirically scales approximately as $O(n_m)$. In contrast, with no simplification (blue), the number of qubits scales as $O(n_m \log(n_m))$, asymptotically [22].

Hamiltonian in Eq. (6), i.e. $H = H_c$. This is done by preparing an ansatz state,

$$|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle = \prod_{i=1}^{s} \left(\exp\left(-i\beta_i H_a\right) \exp\left(-i\gamma_i H_c\right)\right) |+\rangle^{\otimes n} \tag{9}$$

parametrized by angles $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ over n qubits, and where s is the number of layers of the QAOA algorithm. Here, H_a is the *admixing Hamiltonian*,

$$H_a = \sum_{i=1}^{n} \sigma_i^x. \tag{10}$$

For a fixed s, QAOA uses a classical optimizer to minimize the cost function,

$$M(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \langle \boldsymbol{\beta}, \boldsymbol{\gamma}| H_c |\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle . \tag{11}$$

For $s \to \infty$, $M(\boldsymbol{\beta}, \boldsymbol{\gamma})$ is minimized when the fidelity between $|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle$ and the true ground state tends to 1.

Input number m	Number of qubits	Number of carry bits
$35 = 7 \times 5$	2	0
$77 = 11 \times 7$	6	3
$1207 = 71 \times 17$	8	5
$56153 = 241 \times 233$	4	0
$291311 = 557 \times 523$	6	0

TABLE I. Biprime numbers used in this study. The middle and right columns refer to the total number of qubits necessary and among them the number of carry bits produced in the Ising Hamiltonian after simplifying the Boolean equations with rules described in (8). The observed difference between instances with carry bits versus without carry bits is shown in Figure 5.

To optimize the QAOA parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, we employed an iterative brute-force $n_c^2 \times n_c^2$ grid search layer-by-layer, with the output fed into the BFGS global optimization algorithm. The grid size is motivated by a gradient bound given in [27].

IV. SIMULATIONS

In a Python environment utilizing QuTiP [32], we simulated a number of instances of biprime factoring using the algorithm described above. Table I lists all of the instances used. With the technique described in Section III, the success probability of finding the correct factors of $m = 56153$ and $m = 291311$ as a function of the number of layers s is plotted in Figure 2. While one may be tempted to think that it is necessary for the highest amplitude output state of the algorithm to coincide with the correct factor, so long as the correct amplitude scales as $O(1/\text{poly}(n_m))$, one can efficiently sample the output states and check if the corresponding factors divide m. Noting that so long as the success probability From empirical observation, it does not seem that the number of layers or training time is a significant barrier to scaling the algorithm [33].

An obvious concern for the scalability of the algorithm is the number of qubits that are needed to perform the factoring and the associated noise that would be incurred from increased circuit depth. To explore this possibility empirically, we considered a random Pauli-error model with varying probabilities of error after each evolution of the Hamiltonian, which is plotted in Figure 3. Note that in the noisy case the success probability drops below

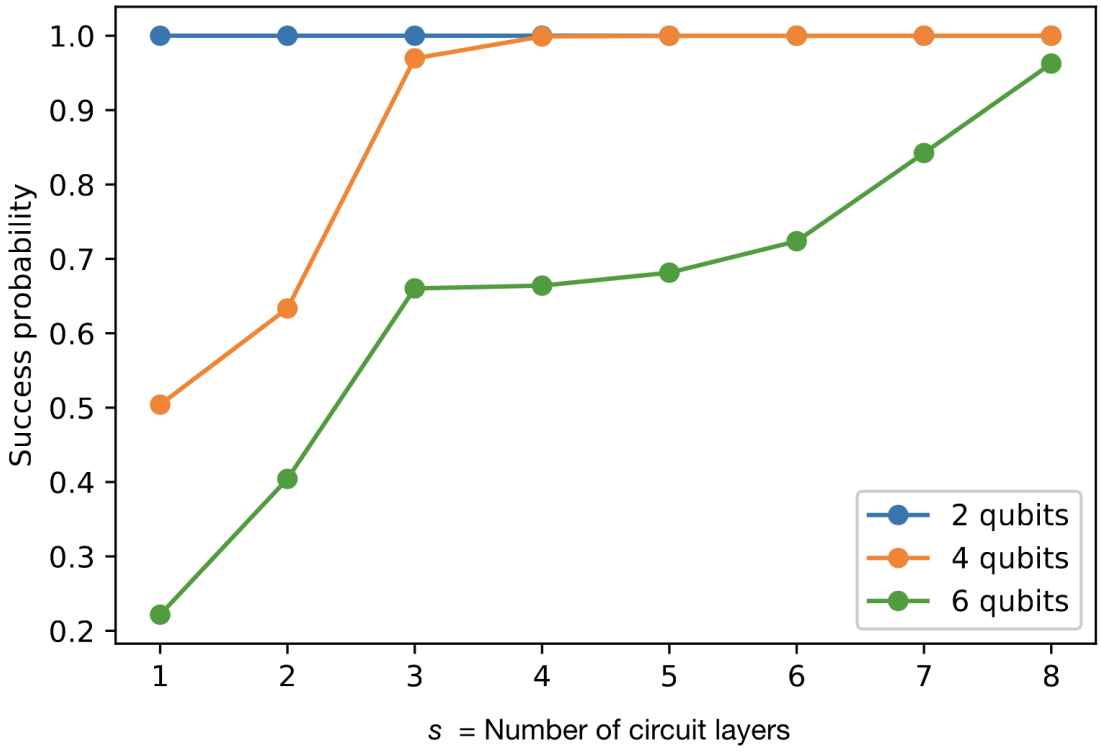


FIG. 2. The probability of sampling the correct state after optimizing the QAOA ansatz for a 2-qubit problem (factoring 35), a 4-qubit problem (factoring 56153), and a 6-qubit problem (factoring 291311). We note that the classically processed factoring equations for these numbers have no carry bits; we empirically noticed worse performance in the presence of carry bits amongst instances requiring the same number of qubits (see Figure 5).

50%, motivating further exploration of error mitigation techniques [34, 35].

V. DISCUSSION

The ability to efficiently solve integer factorization has significant implications for public-key cryptography. In particular, encryption schemes based on abelian groups such as RSA and elliptic curves can be compromised if efficient factorization were feasible. However, an implementation of Shor's algorithm for factoring cryptographically relevant integers would require thousands of *error-corrected* qubits. This is far too lengthy for noisy intermediate-scale quantum devices that are available in the near-term, rendering the potential of quantum

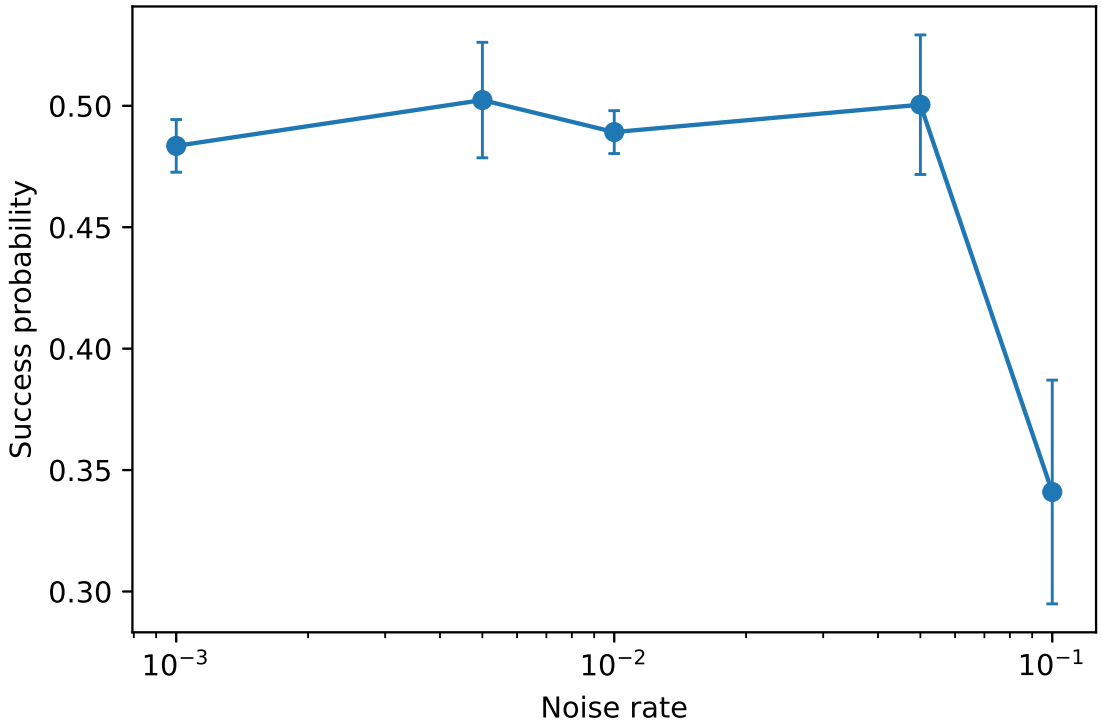


FIG. 3. The dependence on factoring $m = 291311$ with depth $s = 8$ for different Pauli-error noise rates. For sufficiently small noise rates, the success probability is approximately halved when compared to the noiseless case, even at very high depth (see Figure 2).

computers to compromise modern cryptosystems with Shor's algorithm a distant reality. Hybrid approximate quantum/classical methods that utilize classical pre- and post-processing techniques, like the proposed VQF approach, may be more amenable to factoring on a quantum computer in the next decade.

Although we show that it is in principle possible to factor using QAOA, as with most heuristic algorithms, it remains to be seen whether it is capable of scaling asymptotically under realistic constraints posed by imperfect optimization methods and noise on quantum devices. We are currently in the process of examining more detailed analytical and empirical arguments to better determine the potential scalability of the protocol under realistic NISQ conditions. We look forward to working with our collaborators on experimental implementation on current NISQ devices.

The VQF approach can also be employed in an error-corrected setting. Given its heuris-

tic approach it presents a tradeoff of the number of coherent gates versus the number of repetitions, similar to the previous VQE and QAE approaches. In this sense, VQF could be competitive with Shor's algorithm even in the regime of fault-tolerant quantum computation. Further work by our team and the community is needed to benchmark and understand these use cases.

Variational quantum algorithms such as the VQF approach presented here present many stimulating challenges for the community. QAOA, the optimization algorithm employed in our approach, has been studied by several groups in order to understand its effectiveness in several situations [26, 27]. The power and limitations of QAOA are inherited by VQF, and therefore many more numerical experimentations and analytical studies are needed to understand the applicability of VQF in the near future.

VI. ACKNOWLEDGEMENTS

We would like to acknowledge the Zapata Computing scientific team, including Peter Johnson, Jhonathan Romero, Borja Peropadre, and Hannah Sim for their insightful and inspiring comments.

---

[1] P. W. Shor, SIAM Review **41**, 303 (1999), arXiv:9508027.

[2] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, Science **351**, 1068 (2016).

[3] C. Y. Lu, D. E. Browne, T. Yang, and J. W. Pan, Physical Review Letters **99**, 1 (2007).

[4] E. Martín-López, A. Laing, T. Lawson, R. Alvarez, X. Q. Zhou, and J. L. O'brien, Nature Photonics **6**, 773 (2012).

[5] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. V. James, A. Gilchrist, and A. G. White, Physical Review Letters **99**, 5 (2007).

[6] A. Politi, J. C. F. Matthews, and J. L. O. Brien, Science **325**, 1221 (2010).

[7] E. Lucero, R. Barends, Y. Chen, J. Kelly, M. Mariantoni, A. Megrant, P. O'Malley, D. Sank, A. Vainsencher, J. Wenner, T. White, Y. Yin, A. N. Cleland, and J. M. Martinis, Nature Physics **8**, 719 (2012).

[8] M. R. Geller and Z. Zhou, Scientific Reports **3**, 1 (2013).

[9] S. Beauregard, Quantum Information & Computation **3**, 175 (2003), arXiv:0205095v3 [quant-ph].

[10] M. Ekerå, IACR Cryptology ePrint Archive (2016).

[11] M. Ekerå and J. Håstad, arXiv:1702.00249 [cs.CR] (2017).

[12] T. Häner, M. Roetteler, and K. M. Svore, Quantum Info. Comput. **17** (2017).

[13] Y. Takahashi and N. Kunihiro, Quantum Info. Comput. **6**, 184 (2006).

[14] A. G. Fowler, S. J. Devitt, and L. C. L. Hollenberg, Quant. Info. Comput. **4**, 237 (2004).

[15] A. G. Fowler and L. C. L. Hollenberg, Phys. Rev. A **70** (2004), arXiv:0306018v4 [quant-ph].

[16] N. C. Jones, R. Van Meter, A. G. Fowler, P. L. McMahon, J. Kim, T. D. Ladd, and Y. Yamamoto, Physical Review X **2**, 1 (2012).

[17] R. V. Meter, T. D. Ladd, A. G. Fowler, and Y. Yamamoto, Int. J. Quantum Inf. **8**, 295 (2010), arXiv:0906.2686v2 [quant-ph].

[18] D. D. Thaker, T. S. Metodi, A. W. Cross, I. L. Chuang, and F. T. Chong, in *Proceedings - International Symposium on Computer Architecture*, Vol. 2006 (2006) pp. 378–389.

[19] S. J. Devitt, A. M. Stephens, W. J. Munro, and K. Nemoto, Nat. Comm. **4**, 1 (2013).

[20] J. Preskill, Quantum **2** (2018).

[21] D. J. Bernstein, J.-F. Biasse, and M. Mosca, Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Proceedings (2017).

[22] C. J. C. Burges, *Factoring as Optimization*, Tech. Rep. (2002).

[23] N. S. Dattani and N. Bryans, (2014), arXiv:1411.6758.

[24] N. Xu, J. Zhu, D. Lu, X. Zhou, X. Peng, and J. Du, Physical Review Letters **108**, 1 (2012).

[25] S. Jiang, K. A. Britt, A. J. McCaskey, T. S. Humble, and S. Kais, (2018), arXiv:1804.02733 [quant-ph].

[26] E. Farhi and A. W. Harrow, MIT/CTP-4771 (2016), arXiv:1602.07674 [quant-ph].

[27] E. Farhi, J. Goldstone, and S. Gutmann, (2014), arXiv:1411.4028.

[28] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, Nature communications **5**, 4213 (2014).

[29] J. Romero, J. Olson, and A. Aspuru-Guzik, Quantum Science and Technology **2**, 045001 (2016).

[30] R. Dridi and H. Alghassi, Scientific Reports **7**, 43048 (2017), arXiv:1604.05796.

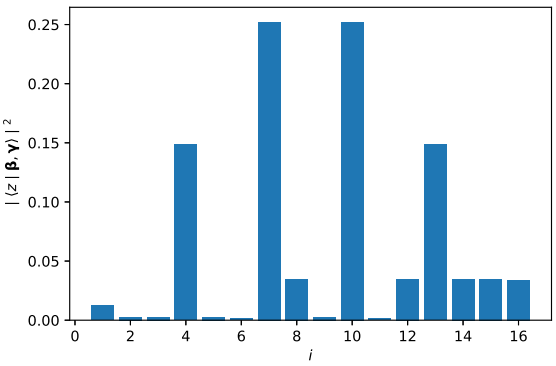
[31] N. Xu, J. Zhu, D. Lu, X. Zhou, X. Peng, and J. Du, Physical Review Letters **108**, 130501 (2012), arXiv:1111.3726.

[32] J. Johansson, P. Nation, and F. Nori, Comp. Phys. Comm. **184**, 1234 (2013), arXiv:1211.6518 [quant-ph].

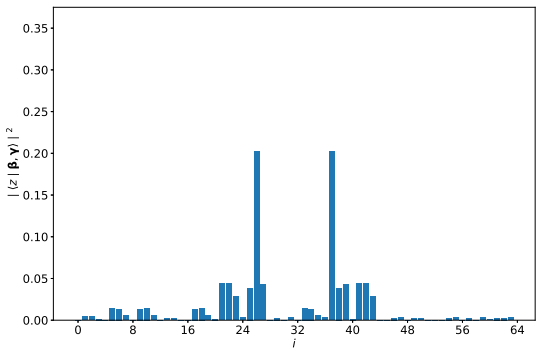
[33] To access the data generated for all instances considered in this study, including those which produced Figures 2-5, please refer to our Github repository at `https://github.com/zapatacomputing/VQFData`.

[34] K. Temme, S. Bravyi, and J. M. Gambetta, Physical Review Letters **119**, 180509 (2017).

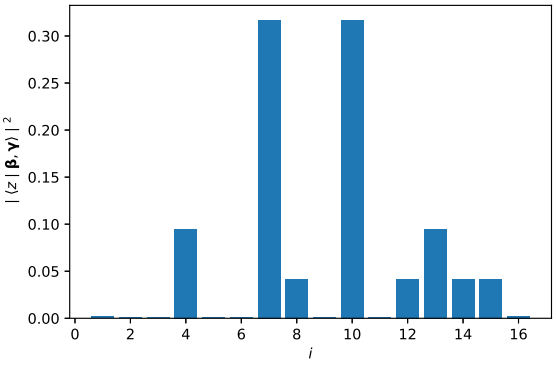
[35] S. Endo, S. C. Benjamin, and Y. Li, Physical Review X **8**, 031027 (2018).



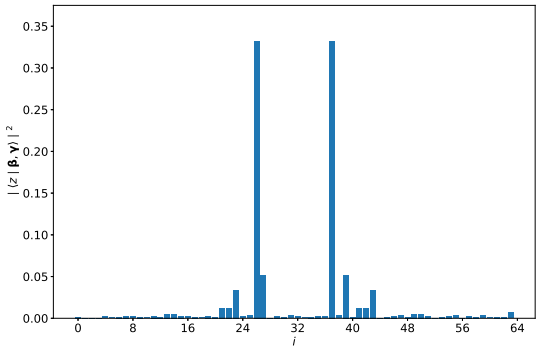
(a) $s = 1$, $m = 56153$



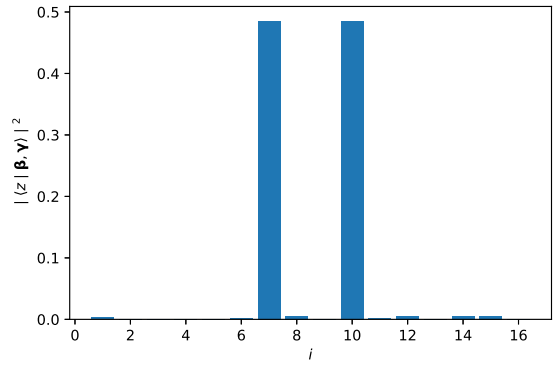
(b) $s = 2$, $m = 291311$



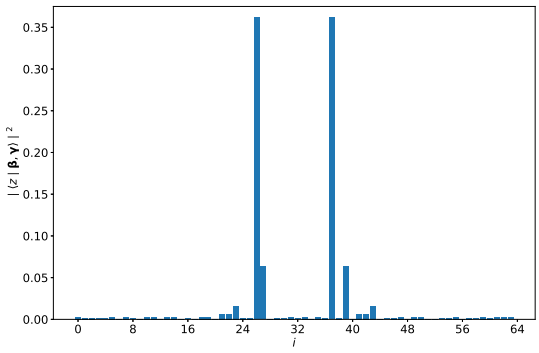
(c) $s = 2$, $m = 56153$



(d) $s = 4$, $m = 291311$

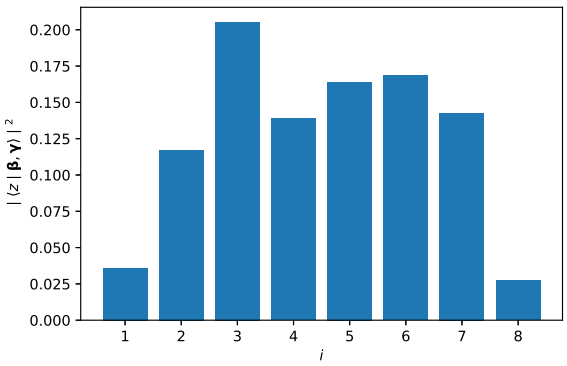


(e) $s = 3$, $m = 56153$

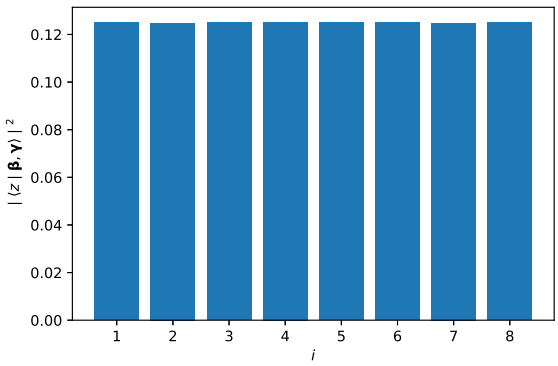


(f) $s = 6$, $m = 291311$

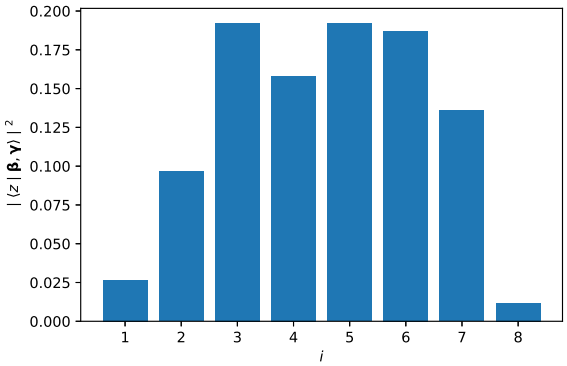
FIG. 4. Distributions corresponding to the output of the presented factoring algorithm for various circuit depths. i labels computational basis states in lexicographic order. The two modes of each diagram correspond to the computational basis states yielding the correct p and q; there are two modes due to the $p \leftrightarrow q$ symmetry of the problem.



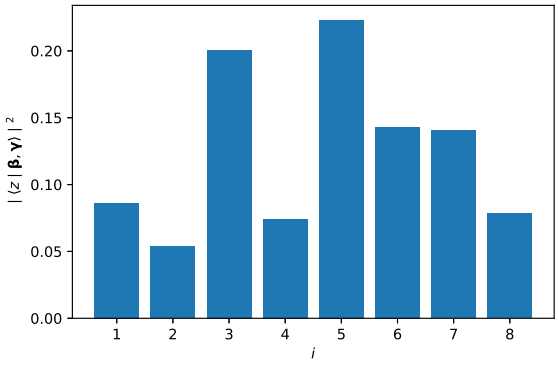
(a) $s = 1$, $m = 77$



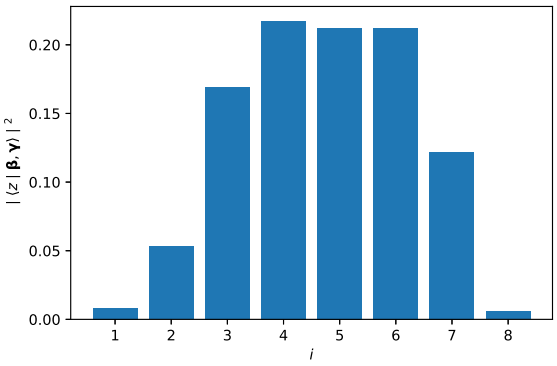
(b) $s = 1$, $m = 1207$



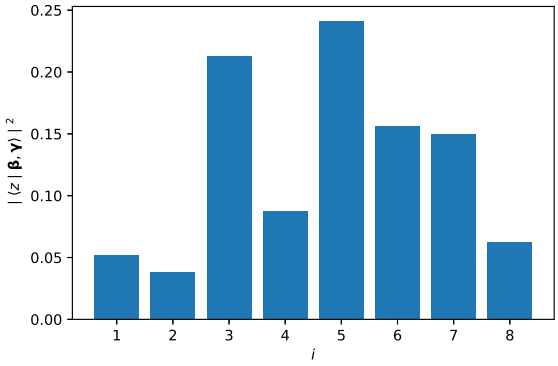
(c) $s = 4$, $m = 77$



(d) $s = 4$, $m = 1207$



(e) $s = 8$, $m = 77$



(f) $s = 8$, $m = 1207$

FIG. 5. Distributions corresponding to the output of the presented factoring algorithm for various circuit depths. The modes of the high depth distributions are the correct solutions. We notice worse performance in depth when the factoring Hamiltonian has carry bits, even for the same number of qubits. We trace over all carry bits when calculating the final distribution, as an incorrect assignment of carry bits can still lead to the correct factors.