SCIENTIFIC REPORTS

OPEN

# Quantum Annealing for Prime Factorization

Shuxian Jiang[1], Keith A. Britt[2], Alexander J. McCaskey[2], Travis S. Humble[2] & Sabre Kais[1,3]

Received: 25 July 2018
Accepted: 8 November 2018
Published online: 05 December 2018

**We have developed a framework to convert an arbitrary integer factorization problem to an executable Ising model by first writing it as an optimization function then transforming the k-bit coupling ($k \geq 3$) terms to quadratic terms using ancillary variables. Our resource-efficient method uses $\mathcal{O}(\log^2(N))$ binary variables (qubits) for finding the factors of an integer $N$. We present how to factorize 15, 143, 59989, and 376289 using 4, 12, 59, and 94 logical qubits, respectively. This method was tested using the D-Wave 2000Q for finding an embedding and determining the prime factors for a given composite number. The method is general and could be used to factor larger integers as the number of available qubits increases, or combined with other ad hoc methods to achieve better performances for specific numbers.**

Integer factorization reduces an integer $N$ to its factors $p$ and $q$ such that $pq = N$. While this fundamental problem in number theory is computationally hard in practice, integer factorization is not believed to belong to the class of NP-hard problems. However, all known classical factoring algorithms which are deterministic and don't have unproven hypotheses require time exponential in log$N$. For example, the fastest, known classical algorithm for integer factorization is the general number field sieve method[1], which scales exponentially in the number of operations required with respect to the integer $N$. Thus, the integer factorization problem has been used as a basic hardness assumption for many encryption methods including the widely deployed RSA cryptographic system. With broad applications in cryptographic data storage and communications[2], identifying new methods for integer factorization plays an important role in modern information security.

Quantum computing theory has the potential to reduce the number of operations required for solving the integer factorization problem. Within the circuit model of quantum computation, Shor's algorithm is perhaps the most well-known method for integer factorization, in which the number of operations to factorize an integer $N$ is polynomial in the size log$N$[3]. This exponential speedup over the general number field sieve is achieved by reducing factorization to the order-finding problem. Several experimental demonstrations using quantum computing hardware have validated the correctness of Shor's algorithm for small integer values, including early work by Vandersypen *et al.*[4] to factorize $N = 15$ using seven spin-1/2 nuclei in a molecule as qubits. Subsequent experiments by Lanyon *et al.*[5], Lu *et al.*[6], and Politi *et al.*[7] implemented compiled versions of Shor's algorithm using photonic systems for factoring 15. Martín-López *et al.*[8] factored 21 using qubit recycling, and Lucero *et al.*[9] used superconducting qubits to factor 15. Geller *et al.*[10] used a simplified version of Shor's algorithm for factoring products of the Fermat primes 3, 5, 17, 257, and 65537. Recent work from Grosshans *et al.* has shown how factoring safe semi-primes using the quantum order-finding algorithm can reduce the failure probability[11].

An equally powerful model of quantum computing is the adiabatic quantum computing (AQC) model[12,13], which can also solve the integer factorization problem. Peng *et al.* first developed integer factorization within AQC by reducing it to unconstrained optimization and solving this problem using adiabatic quantum dynamics. They further validated these ideas experimentally using a three-qubit NMR quantum processor for the case of $N = 21$[14], while Xu *et al.* subsequently factored 143 using similar NMR technology[15]. Schaller *et al.* developed a novel approach based on multiplication tables that can be cast as an optimization, which they have demonstrated for biprimes up to $N = 217$[16]. Dridi *et al.* furthered these ideas by using Gröbner bases to reduce the number of auxiliary variables required and simplify equations, thereby enabling a demonstration of factorization up to 223357[17].

In this contribution, we introduce a new procedure for solving the integer factorization problem using quantum annealing[18,19] which utilizes adiabatic quantum computation. Differ from the recent work[20] which sketched the hardware design of reversible multiplier to achieve factorization, we provide specific mathematical derivations to be tested on the existing hardware. We begin by describing a direct method for integer factorization that

[1]Department of Computer Science, Purdue University, West Lafayette, IN, 47906, USA. [2]Quantum Computing Institute, Oak Ridge National Laboratory, Oak Ridge, TN, 37831, USA. [3]Department of Chemistry, Physics and Birck Nanotechnology Center, Purdue University, West Lafayette, IN, 47906, USA. Correspondence and requests for materials should be addressed to T.S.H. (email: humblets@ornl.gov) or S.K. (email: kais@purdue.edu)

reduces the problem to unconstrained optimization. We review how this optimization problem can be reduced to a quadratic form Ising Hamiltonian and be solved using quantum annealing. We then describe a modified multiplication table method that reduces the overall resource requirements on the optimization problem and permits methods to account for constraints that may appear in quantum annealing hardware, such as connectivity and number of qubits. Our modification also reduces the range of coefficients in the underlying cost function without increasing the number of qubits required. This method avoided the time-consuming preprocessing steps[17] to achieve comparable results, and could be combined with other ad-hoc function simplification methods to further reduce the number of qubits or other aspects. We finally tested both methods with results from experimental demonstrations using quantum annealing hardware.

## Background

Quantum Annealing was introduced[18] to solve optimization problems using quantum fluctuations to transit to the ground state, compared to simulated annealing which uses thermal fluctuations to get to the global minimum. Quantum fluctuations such as quantum tunneling[21] provide ways of transitions between states. The transverse field controls the rate of the transition, as the role of temperature played in simulated annealing.

Farhi *et al.*[12] remodeled the procedure as Adiabatic Quantum Computing(AQC), which finds the energetic ground state of a problem Hamiltonian by adiabatically evolving the quantum state. If the system begins at the ground state, after the adiabatic evolution, the system will remain at the ground state. By combining the initial Hamiltonian $H_B$ and the problem Hamiltonian $H_P$ linearly, the system could be defined as a time-dependent Hamiltonian $H(t) = \left(1 - \frac{t}{T}\right)H_B + \frac{t}{T}H_P$, where the duration $T$ defines the time-scale for evolution and controls the rate at which the time-dependent Hamiltonian changes, the initial Hamiltonian $H_B$ is of the form $H_B = -\sum_{i=1}^{L} \sigma_x^{(i)}$ over $L$ qubits with the Pauli operator $\sigma_x^{(i)}$ defining the $x$-basis of the $i$-th qubit. The problem Hamiltonian $H_P$ is in the form of Ising model over $L$ qubits as

$$H_P = \sum_{i=1}^{L} h_i \sigma_i^z + \sum_{i<j}^{L} J_{ij} \sigma_i^z \sigma_j^z \tag{1}$$

where $\sigma_z^{(i)}$ defines the $z$-basis for the $i$-th qubit and the local fields $h_i$ and couplings $J_{ij}$ define the problem instance.

Computation within the AQC model evolves the $L$-qubit quantum state under the time-dependent Hamiltonian $H(t)$ according to the Schrödinger equation $i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = H(t)|\psi(t)\rangle$, where $|\psi(t)\rangle$ is the state of the system at time $t \in [0, T]$ and we will set $\hbar = 1$. Let $|\phi_i(t)\rangle$ be the $i$-th instantaneous eigenstate, such that $H(t)|\phi_i(t)\rangle = E_i(t)\ |\phi_i(t)\rangle$, and let the initial state of the system be the ground state at time $t = 0$, such that $|\psi(0)\rangle = |\phi_0(0)\rangle$. According to the adiabatic theorem[22], the system state will remain in the instantaneous ground state of the time-dependent Hamiltonian provided the evolution is sufficiently slow to prevent excitations to higher-lying states. Under these idealized adiabatic conditions, the system will evolve into the energetic ground state of the problem Hamiltonian as $|\psi(T)\rangle = |\phi_0(T)\rangle$. This prepared quantum state of $L$ qubits is then measured to generate a classical string of $L$ bits that represents the solution to the encoded factorization problem.

Several practical considerations limit the applicability of the AQC model for solving optimization problems. Foremost is the requirement that changes in the quantum state must be adiabatic, i.e, slow, relative to the internal timescales of the instantaneous Hamiltonian $H(t)$. Theoretical analyses of this requirement provide a best lower bound on the time $T$ as $O(\Delta^{-2})$, where $\Delta$ is the minimum energy gap within the instantaneous eigen spectrum of $H(t)$[13]. However, the minimum spectral gap is dependent on the specific instances of the initial and final Hamiltonians and the interpolation between them. A priori knowledge of spectral gap information would provide a significant insight into the underlying optimization problem, if not the solution directly, and therefore is an impractical expectation for a computational method. In addition, the pure-state model for AQC fails to account for finite-temperature effects observed in actual hardware as well as unexpected environmental coupling, unpredictable control noise, unwanted crosstalk, and other imperfections.

Practically quantum annealing relaxes the AQC with guarantee that the observed final state corresponds to the energetic ground state. However, the probability to observe the energetic ground state may be reduced due to physical noise and non-adiabatic dynamics. The resulting error rate $p_e(L)$ characterizes the quantum annealing dynamics, which is most accurately modeled as an open quantum system of $L$-qubits evolving the presence of an uncontrolled environment. The quantum annealing model is therefore more robust to the above practical considerations but it is necessarily a probabilistic computational model. Statistical sampling of a quantum annealing computation is always necessary to gather confidence in the observed result. Quantum annealing may also be interpreted as a meta-heuristic for managing noisy AQC computation, whereby the aggregate likelihood of success $p_s$ is determined by the number of samples $S$ as $p_s = 1-(p_e)^S$. The number of samples necessary to achieve a desired probability of success is therefore $S \geq \log(1-p_s)/\log(p_e)$. For a fixed annealing duration $T$ and probability of success $p_s$, we may expect the probability of error $p$ to increase as the size of the system increases, i.e., as $L$ increases. The rate at which the sample number $S$ increases with system size plays an important role in determining the computational complexity of using the quantum annealing model. For example, an error rate that increases exponentially with system size, i.e., $p_e \propto \exp(L)$, yields a sampling rate that increases linearly.

A related practical consideration is the resource efficiency with which quantum annealing can be implemented. Specifically, the number of qubits necessary to realize the problem Hamiltonian $H_P$ influences not only the number of samples required but also the feasibility of demonstrating the method on available hardware. The most general case of $L$-qubit Hamiltonian may include all-to-all connectivity, whereby each qubit must interact with every other qubits. However, most of the existing hardware does not permit such connectivity directly, and methods for realizing implicit connections have been developed[23,24]. In our implementation of integer factorization using the Ising Hamiltonian, it is necessary to compose a problem Hamiltonian in terms of pairwise

interactions, and we develop an efficient transformation of the factoring problem Hamiltonian into pair-wise coupling.

## Methods

We describe two methods for implementing integer factorization within the quantum annealing model. We found these two corresponding Hamiltonians $H_P$ to encode the factors of an input integer $N$, such that the energetic ground state corresponds to factorization of the input. The first is a direct method to compute the factors of $N=pq$ by constructing the associated optimization problem as an Ising Hamiltonian. The second method is based on the modified multiplication tables to translate the problem into the Ising Hamiltonian. We test our methods using experimental quantum computing hardware appropriate for quantum annealing. The D-Wave 2000Q processor natively implements an Ising model Hamiltonian and provides programmable control over the parameters $h_i$ and $J_{i,j}$ as well as the annealing duration $T$.

**Direct Method.** Our direct method factors $N=pq$, where $p$ and $q$ are prime numbers. We set $l_1 = \lfloor \log_2(p) \rfloor$ and $l_2 = \lfloor log_2(q) \rfloor$. Because $p$ and $q$ are prime numbers, we use the binary representations $p = (x_{l_1-1}x_{l_1-2}\ldots x_1 1)_2$ and $q = (x_{l_1+l_2-2}x_{l_1+l_2-3}\ldots x_{l_1}1)_2$, where $l_1 > l_2$ and $x_i \in \{0, 1\}$ for $i=1$ to $l_1+l_2-2$. We define the cost function $f(x_1,\ x_2,\ x_3,\ x_4,\ \ldots, x_{l_1+l_2-2}) = (N - pq)^2$, and explicit multiplication of the binary representations for $p$ and $q$ yields a sum of binary products. We reduce the resulting 3-local terms to 2-local terms using the following equivalence[24]: for $x, y, z \in \{0, 1\}$, $xy=z$ iff $xy-2xz-2yz+3z=0$, and $xy \neq z$ iff $xy-2xz-2yz+3z>0$. Therefore,

$$x_1x_2x_3 = x_4x_3 + 2(x_1x_2 - 2x_1x_4 - 2x_2x_4 + 3x_4) \text{ if } x_4 = x_1x_2$$

and

$$x_1x_2x_3 < x_4x_3 + 2(x_1x_2 - 2x_1x_4 - 2x_2x_4 + 3x_4) \text{ if } x_4 \neq x_1x_2$$

thus, the $x_1x_2x_3$ term may be transformed to quadratic form by replacing $x_1x_2$ with $x_4$ plus a constraint in the form of a penalty term:

$$\min(x_1x_2x_3) = \min(x_4x_3 + 2(x_1x_2 - 2x_1x_4 - 2x_2x_4 + 3x_4) \tag{2}$$

By introducing a new variable and adding the penalty term, we are able to transform 3-local terms to 2-local terms.

For integer factorization, we require $\binom{l_1}{2} + \binom{l_2}{2} = \frac{l_1(l_1-1)}{2} + \frac{l_2(l_2-1)}{2}$ auxiliary variables to form a quadratic cost function, and when $l_1=l_2=l$ this number is $l\times(l-1)$. Counting the variables to denote the factors themselves, the quadratic function requires $L=2\times(l-1)+l\times(l-1)=(l+2)\times(l-1)$ binary variables in total. Since $l = \mathcal{O}(\log(N))$, $L = \mathcal{O}(\log^2(N))$. We could also let $p = (1x_{l_1-2}\ldots x_1 1)_2$, $q = (1x_{l_1+l_2-4}\ldots x_{l_1-1}1)_2$ when lengths of $p$ and $q$ are prefixed.

We illustrate this direct method of factorization for the case of $N=15$. Because $\log_2(p) \leq 2 < \log_2(q) < 4$, we define $p=(x_1 1)_2$ and $q=(x_2x_3 1)_2$. The objective function $f(x_1, x_2, x_3)=(N-pq)^2$ may then be reduced to the 3-local form:

$$f(x) = 128x_1x_2x_3 - 56x_1x_2 - 48x_1x_3 + 16x_2x_3 - 52x_1 - 52x_2 - 96x_3 + 196.$$

We reduce the 3-local terms to 2-local terms using the method described above to obtain

$$\begin{aligned} f'(x) \ = \ & 200x_1x_2 - 48x_1x_3 - 512x_1x_4 + 16x_2x_3 - 512x_2x_4 + 128x_3x_4 \\ & -52x_1 - 52x_2 - 96x_3 + 768x_4 + 196, \end{aligned}$$

where

$$\min_{x_1x_2=x_4} f(x_1,\ x_2,\ x_3,\ x_4) = \min f'(x_1,\ x_2,\ x_3,\ x_4)$$

This result is a quadratic unconstrained binary optimization (QUBO) problem over $L=4$ variables that may be transformed into an equivalent Ising Hamiltonian as defined in Eq. 1 by identifying the binary variable $x_i$ with the $i$-th spin state $s_i=2x_i-1$. For $N=15$, the local fields $h^T$ and couplings $J$ of the Ising problem Hamiltonian are then determined to be

$$h^T = (58, 50, 12, -80) \tag{3}$$

and

$$J = \begin{pmatrix} 25 & -6 & -64 \\ & 2 & -64 \\ & & 16 \end{pmatrix} \tag{4}$$

It is notable that the $L\times L$ coupling matrix J is generally dense on the upper triangle, indicating that $L(L-1)/2$ couplings are necessary. Similarly Ising parameters may be generated for other integers $N$ by the appropriate quadritization into a QUBO problem and then reduced to Ising form.

| | $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|---|---|---|---|
| $p$ | | | | | 1 | $p_2$ | $p_1$ | 1 |
| $q$ | | | | | 1 | $q_2$ | $q_1$ | 1 |
| | | | | | 1 | $p_2$ | $p_1$ | 1 |
| | | | | $q_1$ | $p_2q_1$ | $p_1q_1$ | $q_1$ | |
| | | | $q_2$ | $p_2q_2$ | $p_1q_2$ | $q_2$ | | |
| | | 1 | $p_2$ | $p_1$ | 1 | | | |
| carries | | $c_4$ | $c_3$ | $c_2$ | $c_1$ | | | |
| $p \times q = 143$ | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

**Table 1.** Multiplication table for $13 \times 11$ or $11 \times 13 = 143$ in binary.

**Modified Multiplication Table Method.** The second method is based on modified multiplication table. It reduces the range of Ising parameter values used as coefficients for the local fields and couplings. At the meantime, it considers to use a smaller number of carry variables without complicated preprocessing.

The modified multiplication table method uses local minimizations over the products of individual binary substring bits representing the integers $p$ and $q$. It divides the multiplication table into several blocks, and considers each block individually. We could also choose the size of each block to get the desired range of parameters and the number of variables, or make a balance between them. A detailed analysis of the range of coefficients is shown in the last section of the supplemental material. Note that the modified multiplication table method does not eliminate the need for quadratization of 4-body and 3-body product terms and auxiliary variables are required. However, this approach does reduce the number of these higher-order terms compared to the direct method, which makes it possible to embed larger problem sizes on currently available quantum hardware.

We describe this method by an illustrative example of $N = 143$, for which $p = 13$ and $q = 11$. Past approaches for integer factorization constructed a system of equations from each column or part of each column in the multiplication table[17,25,26], which accounts for a carry bit or several carry bits for each part. In our approach, we divide the multiplication table into blocks so that it is only needed to use carries between blocks. This greatly reduced the number of carries, thus the total number of variables reduced too.

As shown in Table 1 for $N = 143$, we introduce two sets of carry bits. We denote them using $c_i \in \{0, 1\}$, and the two-bit numbers $(c_2c_1)_2 = c_2 \times 2 + c_1$ and $(c_4c_3)_2 = c_4 \times 2 + c_3$ represent the carry bits for each of the divided columns in the table. Note that the columns are composed along two-bit domains, so that addition within each block is over two-bit numbers. But the sums are over four-bit numbers. The resulting block system of equations derived from Table 1 is

$$\begin{aligned}
(p_2 + p_1q_1 + q_2) \times 2 + (p_1 + q_1) &= c_2 \times 2^3 + c_1 \times 2^2 + (11)_2 \\
&= c_2 \times 8 + c_1 \times 4 + 3 \\
(q_1 + p_2q_2 + p_1 + c_2) \times 2 + (1 + p_2q_1 + p_1q_2 + 1 + c_1) &= c_4 \times 2^3 + c_3 \times 2^2 + (01)_2 \\
&= c_4 \times 8 + c_3 \times 4 + 1 \\
(1 + c_4) \times 2 + (q_2 + p_2 + c_3) &= (100)_2 \\
&= 4
\end{aligned} \tag{5}$$

Because this modified multiplication table method calculates carries only for each block, it avoids requiring carry bits for each column in the multiplication table. This reduces the overall complexity of the computation by reducing the number of carry bits as well as the number of couplings between bits. In the limit of a single column per block, the conventional multiplication table is recovered, while in the limit of a single equation the direct method is recovered. Instead of making the sum of each column equal to every each bit of the number to be factored as in a conventional multiplication table, we make each block of the multiplication table equal to the corresponding block of the number to be factored. As shown in the appendix material, the equations for these blocks may be reduced to the non-negative cost function

$$\begin{aligned}
f(p,\ q,\ c) \ = \ & (2p_2 + 2p_1q_1 + 2q_2 - 8c_2 - 4c_1 + p_1 + q_1 - 3)^2 \\
& + (2q_1 + 2p_2q_2 + 2p_1 + 2c_2 - 8c_4 - 4c_3 + p_2q_1 + p_1q_2 \\
& + c_1 + 1)^2 + (q_2 + p_2 + c_3 + 2c_4 - 2)^2.
\end{aligned}$$

This form may be expanded and further simplified using the property $x_i^2 = x_i$ for $x_i = 0, 1$, while the remaining cubic and higher-order terms like $c_1p_1q_1$ and $p_1p_2q_1q_2$ can be reduced to quadratic form by introducing auxiliary variables. In particular, we note that the quadratization of the negative term is similar to the position term, e.g.,

$$\begin{cases}
-x_1x_2x_3 = -x_4x_3 + 2(x_1x_2 - 2x_1x_4 - 2x_2x_4 + 3x_4) & \text{if} \quad x_4 = x_1x_2 \\
-x_1x_2x_3 < -x_4x_3 + 2(x_1x_2 - 2x_1x_4 - 2x_2x_4 + 3x_4) & \text{if} \quad x_4 \neq x_1x_2
\end{cases}$$

as detailed in the appendix material, the conversion to QUBO form leads to the parameters for the Ising Hamiltonian. For $N = 143$, this yields the local fields

| | $2^{15}$ | $2^{14}$ | $2^{13}$ | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^{9}$ | $2^{8}$ | $2^{7}$ | $2^{6}$ | $2^{5}$ | $2^{4}$ | $2^{3}$ | $2^{2}$ | $2^{1}$ | $2^{0}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| p | | | | | | | | | 1 | $p_6$ | $p_5$ | $p_4$ | $p_3$ | $p_2$ | $p_1$ | 1 |
| q | | | | | | | | | 1 | $q_6$ | $q_5$ | $q_4$ | $q_3$ | $q_2$ | $q_1$ | 1 |
| | | | | | | | | | 1 | $p_6$ | $p_5$ | $p_4$ | $p_3$ | $p_2$ | $p_1$ | 1 |
| | | | | | | | | $q_1$ | $p_6q_1$ | $p_5q_1$ | $p_4q_1$ | $p_3q_1$ | $p_2q_1$ | $p_1q_1$ | $q_1$ | |
| | | | | | | | $q_2$ | $p_6q_2$ | $p_5q_2$ | $p_4q_2$ | $p_3q_2$ | $p_2q_2$ | $p_1q_2$ | $q_2$ | | |
| | | | | | | $q_3$ | $p_6q_3$ | $p_5q_3$ | $p_4q_3$ | $p_3q_3$ | $p_2q_3$ | $p_1q_3$ | $q_3$ | | | |
| | | | | | $q_4$ | $p_6q_4$ | $p_5q_4$ | $p_4q_4$ | $p_3q_4$ | $p_2q_4$ | $p_1q_4$ | $q_4$ | | | | |
| | | | | $q_5$ | $p_6q_5$ | $p_5q_5$ | $p_4q_5$ | $p_3q_5$ | $p_2q_5$ | $p_1q_5$ | $q_5$ | | | | | |
| | | | $q_6$ | $p_6q_6$ | $p_5q_6$ | $p_4q_6$ | $p_3q_6$ | $p_2q_6$ | $p_1q_6$ | $q_6$ | | | | | | |
| | | 1 | $p_6$ | $p_5$ | $p_4$ | $p_3$ | $p_2$ | $p_1$ | 1 | | | | | | | |
| | $c_{11}$ | $c_{10}$ | $c_9$ | $c_8$ | $c_7$ | $c_6$ | $c_5$ | $c_4$ | $c_3$ | | $c_2$ | $c_1$ | | | | |
| | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

**Table 2.** Multiplication table for $251 \times 239 = 59989$ in binary.

| $2^{18}$ | $2^{17}$ | $2^{16}$ | $2^{15}$ | $2^{14}$ | $2^{13}$ | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^{9}$ | $2^{8}$ | $2^{7}$ | $2^{6}$ | $2^{5}$ | $2^{4}$ | $2^{3}$ | $2^{2}$ | $2^{1}$ | $2^{0}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | 1 | $p_8$ | $p_7$ | $p_6$ | $p_5$ | $p_4$ | $p_3$ | $p_2$ | $p_1$ | 1 |
| | | | | | | | | | 1 | $q_8$ | $q_7$ | $q_6$ | $q_5$ | $q_4$ | $q_3$ | $q_2$ | $q_1$ | 1 |
| | | | | | | | | | 1 | $p_8$ | $p_7$ | $p_6$ | $p_5$ | $p_4$ | $p_3$ | $p_2$ | $p_1$ | 1 |
| | | | | | | | | $q_1$ | $p_8q_1$ | $p_7q_1$ | $p_6q_1$ | $p_5q_1$ | $p_4q_1$ | $p_3q_1$ | $p_2q_1$ | $p_1q_1$ | $q_1$ | |
| | | | | | | | $q_2$ | $p_8q_2$ | $p_7q_2$ | $p_6q_2$ | $p_5q_2$ | $p_4q_2$ | $p_3q_2$ | $p_2q_2$ | $p_1q_2$ | $q_2$ | | |
| | | | | | | $q_3$ | $p_8q_3$ | $p_7q_3$ | $p_6q_3$ | $p_5q_3$ | $p_4q_3$ | $p_3q_3$ | $p_2q_3$ | $p_1q_3$ | $q_3$ | | | |
| | | | | | $q_4$ | $p_8q_4$ | $p_7q_4$ | $p_6q_4$ | $p_5q_4$ | $p_4q_4$ | $p_3q_4$ | $p_2q_4$ | $p_1q_4$ | $q_4$ | | | | |
| | | | | $q_5$ | $p_8q_5$ | $p_7q_5$ | $p_6q_5$ | $p_5q_5$ | $p_4q_5$ | $p_3q_5$ | $p_2q_5$ | $p_1q_5$ | $q_5$ | | | | | |
| | | | $q_6$ | $p_8q_6$ | $p_7q_6$ | $p_6q_6$ | $p_5q_6$ | $p_4q_6$ | $p_3q_6$ | $p_2q_6$ | $p_1q_6$ | $q_6$ | | | | | | |
| | | $q_7$ | $p_8q_7$ | $p_7q_7$ | $p_6q_7$ | $p_5q_7$ | $p_4q_7$ | $p_3q_7$ | $p_2q_7$ | $p_1q_7$ | $q_7$ | | | | | | | |
| | $q_8$ | $p_8q_8$ | $p_7q_8$ | $p_6q_8$ | $p_5q_8$ | $p_4q_8$ | $p_3q_8$ | $p_2q_8$ | $p_1q_8$ | $q_8$ | | | | | | | | |
| 1 | $p_8$ | $p_7$ | $p_6$ | $p_5$ | $p_4$ | $p_3$ | $p_2$ | $p_1$ | 1 | | | | | | | | | |
| | $c_{14}$ | | | $c_{10}$ | $c_9$ | $c_8$ | $c_7$ | $c_6$ | $c_5$ | $c_4$ | $c_3$ | $c_2$ | $c_1$ | | | | | |
| | | $c_{13}$ | $c_{12}$ | $c_{11}$ | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

**Table 3.** Multiplication table for $659 \times 571 = 376289$ in binary.

$$h^T = (130.5, 107.5, 130.5, 107.5, -41, -82, 3, 6, -137, -81, -107, -81) \tag{6}$$

and the upper triangular coupling matrix

$$J = \begin{pmatrix} 2 & 79 & 47.5 & -2 & -4 & -8 & -16 & -148 & -84 & 0 & 0 \\ & 47.5 & 71 & -8 & -16 & 1 & 2 & 6 & 6 & -124 & -84 \\ & & 2 & -2 & -4 & -8 & -16 & -148 & 0 & 0 & -84 \\ & & & -8 & -16 & 1 & 2 & 6 & -84 & -124 & 6 \\ & & & & 34 & -4 & -8 & -8 & 1 & 2 & 1 \\ & & & & & -8 & -16 & -16 & 2 & 4 & 2 \\ & & & & & & 34 & 0 & -4 & -8 & -4 \\ & & & & & & & 0 & -8 & -16 & -8 \\ & & & & & & & & 0 & 1 & 0 \\ & & & & & & & & & 0 & 0 \\ & & & & & & & & & & 0 \end{pmatrix} \tag{7}$$

Our approach requires a decision to partition the columns of the multiplication table into blocks, and this choice must balance the number of unknown variables (carries) against the range of coefficients in the problem Hamiltonian. We illustrate this choice for the factorization of biprimes 59989 and 376289 in Tables 2 and 3. Our approach is to set the bit-length of the carry variable for each block based on the largest possible number of that block(the right neighboring columns). For example, the maximum carry for the right-most block in the multiplication table of $N = 59989$ is 3 which requires two bits to represent. Thus, the bit-length of the carry variable for this block is 2, i.e., $(c_2c_1)_2$. Similarly, for $N = 376289$, the bit-length of the carry variables for the right-most block is 3, while the bit-length of the carry variable for the third block is 4. Because this bit-length is larger than the size of the fourth block, which has a bit-length of 3, the most significant bit of the carry is included in the neighboring block, i.e., the fifth block in this example.

We estimate the number of variables needed to construct the Ising Hamiltonian that encodes the factorization problem for $N = 59989$ and 376289. This requires quadratization of the resulting systems of factoring equations followed by reduction to the Ising form, exactly as discussed explicitly above. For the case of $N = 59989$, we have $l_1 = l_2 = 6$, and therefore 12 variables are required to represent the factors themselves plus 11 variables to denote the carries, while 36 auxiliary variables are required for quadratization of the factoring equations. The total number variables is 59. For $N = 376289$, we have $l_1 = l_2 = 8$ with 14 carries and 64 auxiliary variables. As noted above, sometimes vary bits in the multiplication table will overlap, as is the case for column $2^{14}$ of 376289 shown in Table 3. In such circumstances, we just simply add these carries in the table and then use the same method as before to find the corresponding Ising Hamiltonian. Thus, this problem Hamiltonian requires 94 qubits.

Generally, for factoring a biprime number, we use approximately $\log(N)$ binary variables to encode the integer factors and about $\log(N)$ binary variables to denote the carries where $N$ is the number to be factored. An additional $\log^2(N)/4$ auxiliary binary variables are required for quadratization. Therefore, a total of approximately $\log^2(N)/4$ binary variables are required to represent the problem Hamiltonian and, consequently, a similar number of qubits must be available within hardware. As a point of reference, applying this method to the current factoring record for RSA-768 would require approximately 147456 qubits.

## Experiments

We tested both the direct and modified multiplication table methods using quantum annealing hardware from D-Wave Systems. The D-Wave System hardware consists of a programmable platform of integrated superconducting flux qubits designed to operate within the quantum annealing model. In particular, the hardware system accepts as input a problem Hamiltonian $H_P$ presented in Ising form and the parameters $h$ and $J$. The hardware also enables interpolation from the starting Hamiltonian $H_B$ and the final Hamiltonian at a rate controlled by the annealing duration $T$. Measurements of the resulting quantum state are performed in $\sigma_z$ basis for each qubits.

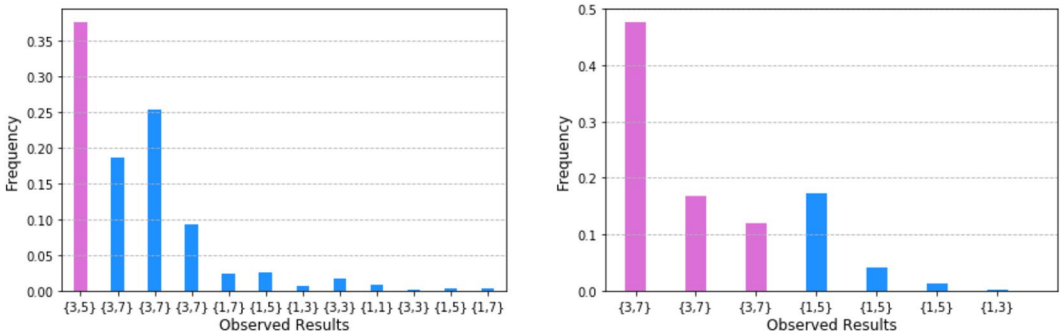
While the latest 2000Q system contains up to 2048 qubits arranged in a connectivity pattern expressed as a 16-by-16 Chimera graph, this sparse connectivity pattern requires additional resources to ensure the required interactions between the logical variables defining the problem Hamiltonian. This is accomplished by embedding the problem Hamiltonian into the hardware graph while maintaining the logical form of the cost function[27–29]. From the coupling matrix $J$, we define a graph $G$ that represents the variables as vertices and non-zero coupling as edges. A minor embedding of $G(V, E)$ into a hardware graph $G'(V', E')$ is defined by a mapping $\phi$: $G \mapsto G'$ such that each vertex $v \in G$ is mapped to a connected subtree $T_v$ of $G'$ and if $(u, v) \in E$ then there exist $i_u, i_v \in G'$ such that $i_u \in T_u$, $i_v \in T_v$ and $(i_u, i_v) \in E'$. If such a mapping $\phi$ exists between $G$ and $G'$, we say $G$ is a *minor* of $G'$ and we use $G \leq_m G'$ to denote such relationship.

The logical parameters for local fields and couplers should also be considered. In the parameter setting problem[30], we assign each node and each edge in the minor embedding graph such that: (1) for each node in the tree $T_i$ expanded by the same vertex $i$, its value $h'_{i_k}$ satisfies $\sum h'_{i_k} = h_i$, (2) for each edge in the tree $T_i$ expanded by the same vertex $i$, the value $J_{i_k, i_{k'}}$ needs to be large enough to make sure all physical qubits that correspond to the same logical qubit to be of the same value and (3) for each edge in the minor embedding graph which is in the original graph, we could use the same $J_{ij}$ value.

Our programmed implementation of these methods were written in C/C++ using the XACC programming framework[31]. XACC enables integration of the D-Wave solver application programming interface (SAPI) using a directive-based programming model. Pre-processing of the input $N$ generated the Ising parameters for a logical Hamiltonian that was then embedded into the hardware graph structure. For minor embedding, we use the sapi_findembedding method included in the D-Wave 2000Q control software, SAPI version 3.0. This embedding methods is based on a randomized algorithm from Cai, Macready and Roy[32], while access to these methods were managed using the XACC dwsapi-embedding plugin[31]. The corresponding biases and couplings for the embedded problem were generated using the logical Ising parameters. The output of the embedding was a program implementation of the physical Ising model that was submitted for execution on the D-Wave processor. Additional parameters for the execution included the number of samples $S$ and the annealing duration $T$. The default annealing schedule for the 2000Q was used for all executions. The output from each of the $S$ executions was a measured binary string designating $\pm 1$ values for each spin variable. The number of samples was $S = 10{,}000$. Each returned string was then classified according to the corresponding energy for the physical Ising model and subsequently decoded into the factors $p$ and $q$. A histogram of all solutions returned for a specific annealing time was recorded.

Figure 1(a,b) shows the frequency of each decoded solutions to the factorization problem for $N = 15$ and 21 using the direct method. These observed solutions are decoded using the inverse of the embedding with majority vote used to resolve any ambiguity in results. The plot presents the decoded results in order of lowest energy to highest energy (left to right). For these two examples, the lowest energy solution corresponds to the correct factors. In addition, several other computed solutions decoded into the correct factors as the associated errors were resolved by the decoding method. For example, several solutions are labeled as (3,7) because the observed bit strings corresponded to high-energy states before decoding. Only the first (leftmost) corresponds to the lowest energy state. The others were higher energy solutions, thus can't be counted as the correct solution.

Using the modified multiplication table method for factoring 143, we embed the problem Hamiltonian to D-Wave machine using the following method. Suppose $n$ qubits are needed in the Hamiltonian, we divide $n$ into $\left\lfloor \frac{n}{4} \right\rfloor$ groups. For each group, we use 4 copies of the nodes with each $h'_{i_k} = \frac{1}{4} h_i$. We assign each edge in the tree $T_i$ the negative number with largest absolute value to make it a penalty term. This method guarantees the nodes correspond to the same original qubit have the same value. We assign each edge corresponding to the original edge in the problem graph the same $J_{ij}$ value. The embedded graph to D-Wave machine is in Fig. 2.




(a) Factoring 15 with annealing time 200 (b) Factoring 21 with annealing time 2000

**Figure 1.** Experimental results on D-Wave machine: rates of getting different solutions. For example, the (3, 5) in the x-axis denotes the factorization of 15 is 3 multiplied by 5, the number in y-axis denotes the rate to get this factorization.

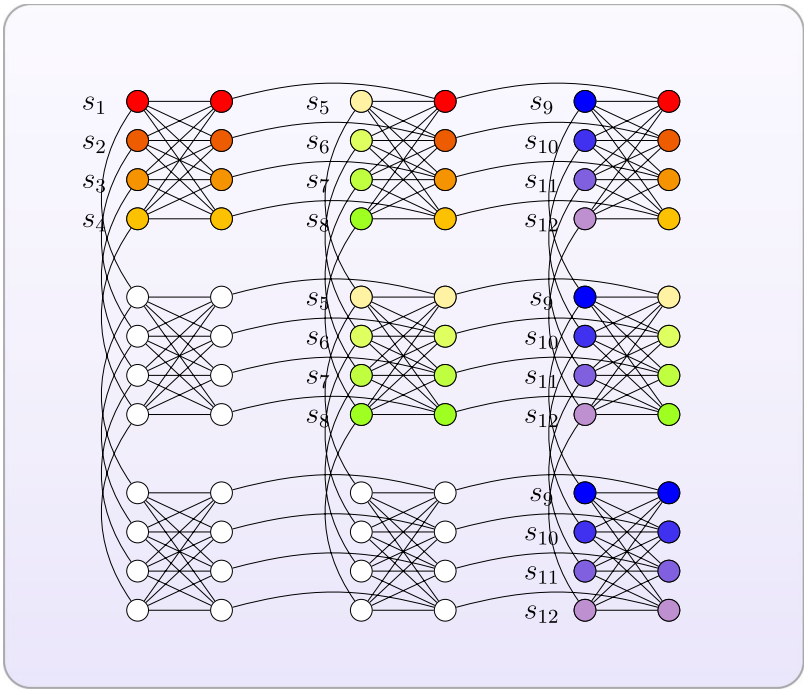



**Figure 2.** Embedding the factoring instance $N = 143$ to Chimera graph. The nodes with the same color denote the same original qubit, with their connected lines corresponding to strong couplings. The left footnotes refer to which spin the node was embedded.

The results graph are shown in Fig. 3. The final state of the part of the system which represents the problem solution will be $|1\ -1\ -1\ 1\rangle$ or $|-1\ 1\ 1\ -1\rangle$ with relatively high probability, which corresponds to solutions $p = (1p_2p_1 1) = (1101)_2 = 13$, $q = (1q_2q_1 1) = (1011)_2 = 11$ or $p = 11$, $q = 13$.

For factoring larger numbers like 376289 using D-Wave, we embed the Hamiltonians into the Chimera hardware using the predefined function *find_embedding* and *embed_problem* available in the vendor's software developer packages, because the problem graph can't be embedded directly like the case 143 shown above which has 12 qubits, due to the size limitation of the current Chimera hardware. For now, the largest number experimented for finding an embedding in D-Wave 2000Q is 249919 which equals to $509 \times 491$. It uses 74 qubits in the final Ising Hamiltonian, and embeds to 1803 physical qubits in Chimera graph.

## Conclusions

We have presented two general methods for factoring integers using quantum annealing for optimizing a cost function that is reduced to an Ising Hamiltonian. Both methods requires $\mathcal{O}(\log^2(N))$ qubits in total, where $N$ is the number to be factored. The novelty of our demonstration of quantum annealing for prime factorization is based on the reduction in quantum resources required to execute factoring and the experimental verification of

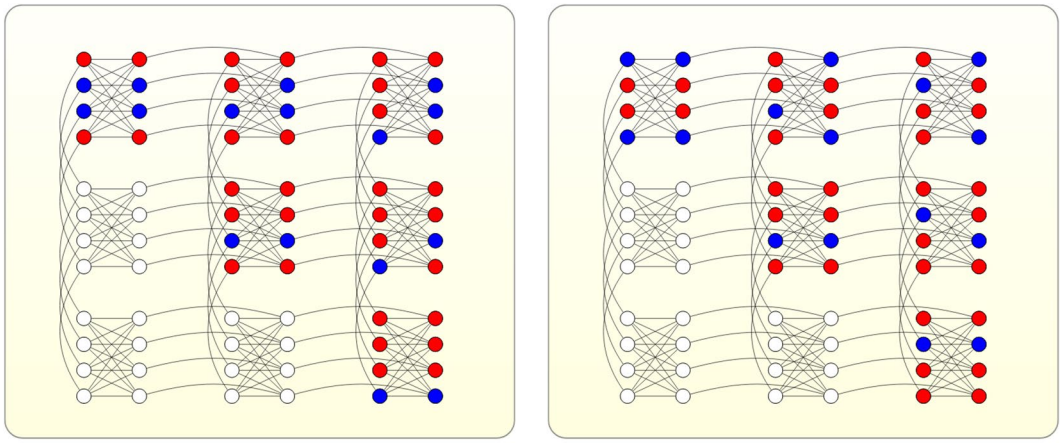



**Figure 3.** Experimental results on D-Wave machine: final ground state of factoring 143. Nodes colored red denote +1, nodes colored blue denote −1. (**a**) This graph shows $s_1 = 1$, $s_2 = -1$, $s_3 = -1$, $s_4 = 1$ which means $p = 1101$, $q = 1011$. (**b**) This graph shows $s_1 = -1$, $s_2 = 1$, $s_3 = 1$, $s_4 = -1$ which means $p = 1011$, $q = 1101$.

the algorithmic accuracy using currently available hardware. As a proof-of-concept, we have demonstrated these methods by factoring integers using the D-Wave 2000Q quantum annealing hardware, but these methods may be used on any other quantum annealing system with a similar number of qubits, qubit degree of connectivity, and hardware parameter precision. Assuming that quantum annealing hardware systems will continue to grow both in the number of qubits and bits of precision capabilities, our methods offer a promising path toward factor much larger numbers in the future. It is also good to combine our method with other ad hoc methods to achieve significantly better performances for specific numbers.

Finally, we note that while our demonstrations of factoring have made use of currently available quantum annealers, there is an outstanding question regarding the asymptotic complexity for this approach. It is well known that algorithmic complexity within the AQC model depends on the minimum spectral gap between the ground and first-excited states of the underlying time-dependent Hamiltonian. Attempts to classify the complexity of the spectral gap with respect to system size have not yet succeed and, indeed, Cubitt, Perez-Garcia, and Wolf have proven that the problem of claiming a Hamiltonian has a gap is undecidable in general[33]. Nonetheless, there is hope that our resource-efficient algorithms may find use in pre-processing potential factors for noisy factorization algorithms, e.g., as suggested by Patterson *et al*. within the context of RSA[34].

## Data Availability

The data that support the plots within this paper and other findings of this study are available from the corresponding author upon reasonable request.

## References

1. Lenstra, A. K., Lenstra, H. W. Jr., Manasse, M. S. & Pollard, J. M. The number field sieve. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, 564–572 (ACM, 1990).
2. Wagstaff, S. S. *The Joy of Factoring*, vol. 68 (American Mathematical Soc., 2013).
3. Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review* **41**, 303–332 (1999).
4. Vandersypen, L. M. *et al*. Experimental realization of shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature* **414**, 883–887 (2001).
5. Lanyon, B. *et al*. Experimental demonstration of a compiled version of shor's algorithm with quantum entanglement. *Phys. Rev. Lett.* **99**, 250505 (2007).
6. Lu, C.-Y., Browne, D. E., Yang, T. & Pan, J.-W. Demonstration of a compiled version of shor's quantum factoring algorithm using photonic qubits. *Phys. Rev. Lett.* **99**, 250504 (2007).
7. Politi, A., Matthews, J. C. & O'brien, J. L. Shor's quantum factoring algorithm on a photonic chip. *Science* **325**, 1221–1221 (2009).
8. Martín-López, E. *et al*. Experimental realization of shor's quantum factoring algorithm using qubit recycling. *Nat. Photonics* **6**, 773–776 (2012).
9. Lucero, E. *et al*. Computing prime factors with a josephson phase qubit quantum processor. *Nat. Phys.* **8**, 719–723 (2012).
10. Geller, M. R. & Zhou, Z. Factoring 51 and 85 with 8 qubits. *Sci. reports* **3** (2013).
11. Grosshans, F., Lawson, T., Morain, F. & Smith, B. Factoring safe semiprimes with a single quantum query. *arXiv preprint arXiv:1511.04385* (2015).
12. Farhi, E. *et al*. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* **292**, 472–475 (2001).
13. Albash, T. & Lidar, D. A. Adiabatic quantum computation. *Rev. Mod. Phys.* **90**, 015002, https://doi.org/10.1103/RevModPhys.90.015002 (2018).
14. Peng, X. *et al*. Quantum adiabatic algorithm for factorization and its experimental implementation. *Phys. Rev. Lett.* **101**, 220405 (2008).
15. Xu, N. *et al*. Quantum factorization of 143 on a dipolar-coupling nuclear magnetic resonance system. *Phys. review letters* **108**, 130501 (2012).
16. Schaller, G. & Schützhold, R. The role of symmetries in adiabatic quantum algorithms. *arXiv preprint arXiv:0708.1882* (2007).

17. Dridi, R. & Alghassi, H. Prime factorization using quantum annealing and computational algebraic geometry. *Sci. Reports* **7** (2017).
18. Kadowaki, T. & Nishimori, H. Quantum annealing in the transverse ising model. *Phys. Rev. E* **58**, 5355 (1998).
19. Das, A. & Chakrabarti, B. K. Colloquium: Quantum annealing and analog quantum computation. *Rev. Mod. Phys.* **80**, 1061 (2008).
20. Maezawa, M., Imafuku, K., Hidaka, M., Koike, H. & Kawabata, S. Design of quantum annealing machine for prime factoring. In *Superconductive Electronics Conference (ISEC), 2017 16th International*, 1–3 (IEEE, 2017).
21. Ray, P., Chakrabarti, B. K. & Chakrabarti, A. Sherrington-kirkpatrick model in a transverse field: Absence of replica symmetry breaking due to quantum fluctuations. *Phys. Rev. B* **39**, 11828 (1989).
22. Messiah, A. *Quantum Mechanics: Volume 2* (North-Holland Publishing Company, 1962).
23. Biamonte, J. D. Nonperturbative k-body to two-body commuting conversion hamiltonians and embedding problem instances into ising spins. *Phys. Rev. A* **77**, 052331 (2008).
24. Boros, E. & Hammer, P. L. Pseudo-boolean optimization. *Discret. applied mathematics* **123**, 155–225 (2002).
25. Dattani, N. S. & Bryans, N. Quantum factorization of 56153 with only 4 qubits. *arXiv preprint arXiv:1411.6758* (2014).
26. Pal, S., Moitra, S., Anjusha, V., Kumar, A. & Mahesh, T. Hybrid scheme for factorization: Factoring 551 using a 3-qubit nmr quantum adiabatic processor. *arXiv preprint arXiv:1611.00998* (2016).
27. Choi, V. Minor-embedding in adiabatic quantum computation: I. the parameter setting problem. *Quantum Inf. Process.* **7**, 193–209 (2008).
28. Klymko, C., Sullivan, B. D. & Humble, T. S. Adiabatic quantum programming: minor embedding with hard faults. *Quantum Inf. Process.* **13**, 709–729, https://doi.org/10.1007/s11128-013-0683-9 (2014).
29. Humble, T. S. *et al.* An integrated programming and development environment for adiabatic quantum optimization. *Comput. Sci. Discov.* **7**, 015006 (2014).
30. Choi, V. Minor-embedding in adiabatic quantum computation: Ii. minor-universal graph design. *Quantum Inf. Process.* **10**, 343–353 (2011).
31. McCaskey, A. J. *et al.* Extreme-scale programming model for quantum acceleration within high performance computing. *arXiv preprint arXiv:1710.01794* (2017).
32. Cai, J., Macready, W. G. & Roy, A. A practical heuristic for finding graph minors. *arXiv preprint arXiv:1406.2741* (2014).
33. Cubitt, T. S., Perez-Garcia, D. & Wolf, M. M. Undecidability of the spectral gap. *Nature* **528**, 207 (2015).
34. Paterson, K. G., Polychroniadou, A. & Sibborn, D. L. A coding-theoretic approach to recovering noisy rsa keys. In Wang, X. & Sako, K. (eds) *Advances in Cryptology – ASIACRYPT 2012*, 386–403 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012).

## Acknowledgements

Access to the D-Wave 2000Q was provided by the Quantum Computing Institute at Oak Ridge National Laboratory and Google Quantum Artificial Intelligence Lab, USRA-Purdue. This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

## Author Contributions

S.K. and T.S.H. designed the research, S.J. performed the calculations, S.J., A.M. and K.A.B. run simulations on the D-Wave machine. All authors analyzed the data and wrote the paper.

## Additional Information

**Supplementary information** accompanies this paper at https://doi.org/10.1038/s41598-018-36058-z.

**Competing Interests:** The authors declare no competing interests.

**Publisher's note:** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.