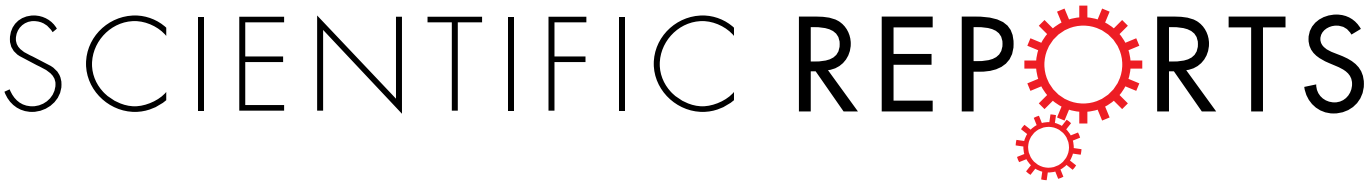


OPEN

Prime factorization using quantum annealing and computational algebraic geometry

Received: 18 October 2016
Accepted: 18 January 2017
Published: 21 February 2017

Raouf Dridi & Hedayat Alghassi

We investigate prime factorization from two perspectives: quantum annealing and computational algebraic geometry, specifically Gröbner bases. We present a novel autonomous algorithm which combines the two approaches and leads to the factorization of all bi-primes up to just over 200000, the largest number factored to date using a quantum processor. We also explain how Gröbner bases can be used to reduce the degree of Hamiltonians.

Prime factorization is at the heart of secure data transmission because it is widely believed to be NP-complete. In the prime factorization problem, for a large bi-prime M, the task is to find the two prime factors p and q such that $M = pq$. In RSA cryptosystem, the message to be transmitted is encrypted using a public key which is, essentially, a large bi-prime that can only be decrypted using its prime factors, which are kept in a private key. Prime factorization also connects to many branches of mathematics; two branches relevant to us are computational algebraic geometry[1] and quantum annealing[2–4].

To leverage the problem of finding primes p and q into the realm of computational algebraic geometry, it suffices to transform it into a system of algebraic equations $\mathcal{S}$. This is done using the binary representation $p = 1 + \sum_{i=1\ldots s_p} 2^i P_i$ and $q = 1 + \sum_{i=1\ldots s_q} 2^i Q_i$, which is plugged into $M = pq$ and expanded into a system of polynomial equations. The system $\mathcal{S}$ is given by this initial system of equations in addition to the auxiliary equations expressing the binary nature of the variables P_i and Q_i, carry-on, and connective variables. The two primes p and q are then given by the unique zero of $\mathcal{S}$. In theory, we can solve the system $\mathcal{S}$ using Gröbner bases; however, in practice, this alone does not work, since Gröbner basis computation (Buchberger's algorithm) is exponential in the number of variables.

The connection to quantum annealing can also be easily described. Indeed, finding p and q can be formulated into an unconstrained binary optimization problem ($\mathcal{P}$), where the cost function f is the sum of the squares of polynomials in $\mathcal{S}$. The unique zero of $\mathcal{S}$ now sits on the unique global minimum of ($\mathcal{P}$) (which has minimum energy equal to zero). There are, however, a few non-trivial requirements we need to deal with before solving the cost function using quantum annealing. These requirements concern the nature of cost functions that quantum annealers can handle. In particular, we would like the cost function of ($\mathcal{P}$) to be a positive quadratic polynomial. We also require that the coefficients of the cost function (coupling and external field parameters) be rather uniform and match the hardware-imposed dynamic range.

In the present paper, we suggest looking into the problem through both lenses, and demonstrate that indeed this approach gives better results. In our scheme, we will be using quantum annealing to solve ($\mathcal{P}$), but at the same time we will be using Gröbner bases to help us reduce the cost function f into a positive quadratic polynomial f^+ with desired values for the coefficients. We will be also using Gröbner bases at the important step of pre-processing f^+ before finally passing it to the quantum annealer. This pre-processing significantly reduces the size of the problem. The result of this combined approach is an algorithm with which we have been able to factorize all bi-primes up to 2×10^5 using the D-Wave 2X processor. The algorithm is autonomous in the sense that no a priori knowledge, or manual or ad hoc pre-processing, is involved. We refer the interested reader to Supplementary materials for a brief description of the D-Wave 2X processor, along with some statistics for several of the highest numbers that we embedded and solved. More detail about the processor architecture can be found in ref. 5. Another important reference is the work of S. Boixo *et al.* in ref. 6, which presents experimental evidence that the scalable D-Wave processor implements quantum annealing (with surprising robustness against noise and imperfections).

1QB Information Technologies (1QBit), Vancouver, British Columbia, V6C 2B5, Canada. Correspondence and requests for materials should be addressed to R.D. (email: raouf.dridi@1qbit.com) or H.A. (email: hedayat.alghassi@1qbit.com)

Additionally, evidence that, during a critical portion of quantum annealing, the qubits become entangled and entanglement persists even as the system reaches equilibrium is presented in ref. 7.

Relevant to us also is the work in ref. 8, which uses algebraic geometry to solve optimization problems (though not specifically factorization; see *Methods* for an adaptation to factorization). Therein, Gröbner bases are used to compute standard monomials and transform the given optimization problem into an eigenvalue computation. Gröbner basis computation is the main step in this approach, which makes it inefficient. In contrast to that work, we ultimately solve the optimization problem using a quantum annealing processor and pre-process and adjust the problem with algebraic tools, that is, we reduce the size of the cost function and adjust the range of its parameters. However, we share that work's point of view of using real algebraic geometry, and our work is the first to introduce algebraic geometry, and Gröbner bases in particular, to solve quantum annealing-related problems. We think that this is a fertile direction for both practical and theoretical endeavours.

Mapping the factorization problem into a degree-4 unconstrained binary optimization problem is first discussed in ref. 9. There, the author proposes solving the problem using a continuous optimization method he calls curvature inversion descent. Another related work is the quantum annealing factorization algorithm proposed in ref. 10. We will discuss it in the next section and improve upon it in two ways. The first involves the addition of the pre-processing stage using Gröbner bases of the cost function. This dramatically reduces the number of variables therein. The second way concerns the reduction of the initial cost function, for which we propose a general Gröbner basis scheme that precisely answers the various requirements of the cost function. In *Results*, we present our algorithm (the column algorithm) which outperforms this improved algorithm (i.e., the cell algorithm). Using a reduction proposed in ref. 10 and ad-hoc simplifications, the paper[11] reports the factorization of bi-prime 143 on a liquid-crystal NMR quantum processor. It has been then observed by ref. 12 that the same 4-qubit Hamiltonian can be used to factor biprimes 3599, 11663, and 56153. More recently, in ref. 13, the authors factored the bi-prime 551 using a 500 MHz NMR spectrometer.

This review will not be complete without mentioning Shor's algorithm[14] and Kitaev's phase estimation[15], which, respectively, solve the factorization problem and the abelian hidden subgroup problem in polynomial time, both for the gate model paradigm. The largest number factored using a physical realization of Shor's algorithm is 15[16]; see ref. 17 also for a discussion about oversimplification in the previous realizations. Finally, in ref. 18, it has been proved that contextuality (Kochen-Specker theorem) is needed for any speed-up in a measurement-based quantum computation factorization algorithm.

Results

The binary multiplication of the two primes p and q can be expanded in two ways: cell-based and column-based procedures (see *Methods*). Each procedure leads to a different unconstrained binary optimization problem. The cell-based procedure creates the unconstrained binary quadratic programming problem

$$(\mathcal{P}_1)\begin{cases} min_{\mathbb{Z}_2}\sum_{ij} H_{ij}^2, \\ \text{with } H_{ij}: = Q_i P_j + S_{i,j} + Z_{i,j} - S_{i+1,j-1} - 2Z_{i,j+1}, \end{cases} \tag{1}$$

and the column-based procedure results in the problem

$$(\mathcal{P}_2)\begin{cases} min_{\mathbb{Z}_2} \sum_{1\le i\le (s_p+s_q+1)} H_i^2, \\ \text{with } H_i: = \sum_{j=0}^{s_q} Q_j P_{i-j} + \sum_{j=1}^{i} Z_{j,i} - m_i - \sum_{j=1}^{s_q+1+i-m_i} 2^{j-i} Z_{i,i+j}. \end{cases} \tag{2}$$

The two problems $(\mathcal{P}_1)$ and $(\mathcal{P}_2)$ are equivalent. Their cost functions are not in quadratic form, and thus must be reduced before being solved using a quantum annealer. The reduction procedure is not a trivial task. In this paper we define, for both scenarios: (1) a reduced quadratic positive cost function and (2) a pre-processing procedure. Thus, we present two different quantum annealing-based prime factorization algorithms. The first algorithm's decomposition method (i.e., the cell procedure) has been addressed in ref. 10, without pre-processing and without the use of Gröbner bases in the reduction step. Here, we discuss it from the Gröbner bases framework and add the important step of pre-processing. The second algorithm, however, is novel in transformation of its quartic terms to quadratic, outperforming the first algorithm due to its having fewer variables.

We write $\mathbb{R}[x_1, \ldots, x_n]$ for the ring of polynomials in $x_1, \ldots, x_n$ with real coefficients and $\mathcal{V}(f)$ for the affine variety defined by the polynomial $f \in \mathbb{R}[x_1, \ldots, x_n]$, that is, the set of zeros of the equation $f = 0$. Since we are interested only in the binary zeros (i.e., $x_i \in \mathbb{Z}_2$), we need to add the binarization polynomials $x_i(x_i - 1)$, where $i = 1, \ldots, n$, to f and obtain the system $\mathcal{S} = \{f, x_i(x_i - 1), i = 1, \ldots, n\}$. The system $\mathcal{S}$ generates an ideal $\mathcal{I}$ by taking all linear combinations over $\mathbb{R}[x_1, \ldots, x_n]$ of all polynomials in $\mathcal{S}$; we have $\mathcal{V}(\mathcal{S}) = \mathcal{V}(\mathcal{I})$. The ideal $\mathcal{I}$ reveals the hidden polynomials which are the consequence of the generating polynomials in $\mathcal{S}$. To be precise, the set of all hidden polynomials is given by the so-called radical ideal $\sqrt{\mathcal{I}}$, which is defined by $\sqrt{\mathcal{I}} = \{g \in \mathbb{R}[x_1, \ldots, x_n] \,|\, \exists\, r \in \mathbb{N}: g^r \in \mathcal{I}\}$. In practice, the ideal $\sqrt{\mathcal{I}}$ is infinite, so we represent such an ideal using a Gröbner basis $\mathcal{B}$ which one might take to be a triangularization of the ideal $\sqrt{\mathcal{I}}$. In fact, the computation of Gröbner bases generalizes Gaussian elimination in linear systems. We also have $\mathcal{V}(\mathcal{S}) = \mathcal{V}(\mathcal{I}) = \mathcal{V}(\sqrt{\mathcal{I}}) = \mathcal{V}(\mathcal{B})$ and $\mathcal{I}(\mathcal{V}(\mathcal{I})) = \sqrt{\mathcal{I}}$. A brief review of Gröbner bases is given in *Methods*.

The cell algorithm. Suppose we would like to define the variety $\mathcal{V}(\mathcal{I})$ by the set of global minima of an unconstrained optimization problem $min_{\mathbb{Z}_2^n}(f^+)$, where f^+ is a quadratic polynomial. For instance, we would like f^+ to behave like f^2. Ideally, we want f^+ to remain in $\mathbb{R}[x_1, \ldots, x_n]$ (i.e., not in a larger ring), which implies that no slack variables will be added. We also want f^+ to satisfy the following requirements:

(i). f^+ vanishes on $V(\mathcal{I})$ or, equivalently, $f^+ \in \sqrt{\mathcal{I}}$.
(ii). $f^+ > 0$ outside $V(\mathcal{I})$, that is, $f^+ > 0$ over $\mathbb{Z}_2^n - V(\mathcal{I})$.
(iii). Coefficients of the polynomial f^+ are adjusted with respect to the dynamic range allowed by the quantum processor.

Let $\mathcal{B}$ be a Gröbner basis for $\mathcal{I}$. We can then go ahead and define

$$f^+ = \sum_{t\in\mathcal{B}|deg(t)\leq 2} a_t t, \tag{3}$$

where the real coefficients a_i are subject to the requirements above; note that we already have $f^+ \in \sqrt{\mathcal{I}}$ and thus the first requirement (i) is satisfied.

Let us apply this procedure to the optimization problem $(\mathcal{P}_1)$ above. There, $f = H_{ij}$ and the ring of polynomials is $\mathbb{R}\left[P_j, Q_i, S_{i,j}, S_{i+1,j-1}, Z_{i,j}, Z_{i,j+1}\right]$. We obtain the following Gröbner basis (see *Methods* about algorithm used):

$$\begin{cases}
t_1 &:= Q_iP_j + S_{i,j} + Z_{i,j} - S_{i+1,j-1} - 2Z_{i,j+1},\\
t_2 &:= \left(-Z_{i,j+1} + Z_{i,j}\right)S_{i+1,j-1} + \left(Z_{i,j+1} - 1\right)Z_{i,j},\\
t_3 &:= \left(-Z_{i,j+1} + Z_{i,j}\right)S_{i,j} + Z_{i,j+1} - Z_{i,j+1}Z_{i,j},\\
t_4 &:= \left(S_{i+1,j-1} + Z_{i,j+1} - 1\right)S_{i,j} - S_{i+1,j-1}Z_{i,j+1},\\
t_5 &:= \left(-S_{i+1,j-1} - 2Z_{i,j+1} + Z_{i,j} + S_{i,j}\right)Q_i - S_{i,j} - Z_{i,j} + S_{i+1,j-1} + 2Z_{i,j+1},\\
t_6 &:= \left(-S_{i+1,j-1} - 2Z_{i,j+1} + Z_{i,j} + S_{i,j}\right)P_j - S_{i,j} - Z_{i,j} + S_{i+1,j-1} + 2Z_{i,j+1},\\
t_7 &:= \left(-Z_{i,j+1} + Z_{i,j+1}Z_{i,j}\right)Q_i + Z_{i,j+1} - Z_{i,j+1}Z_{i,j},\\
t_8 &:= -S_{i+1,j-1}Z_{i,j+1} + S_{i+1,j-1}Q_iZ_{i,j+1},\\
t_9 &:= \left(-Z_{i,j+1} + Z_{i,j+1}Z_{i,j}\right)P_j + Z_{i,j+1} - Z_{i,j+1}Z_{i,j},\\
t_{10} &:= -S_{i+1,j-1}Z_{i,j+1} + S_{i+1,j-1}P_jZ_{i,j+1}.
\end{cases} \tag{4}$$

We have used the lexicographic order $plex(P_j, Q_i, S_{i,j}, S_{i+1,j-1}, Z_{i,j}, Z_{i,j+1})$; see *Methods* for definitions. Note that $t_1 = H_{ij}$. We define

$$H_{ij}^+ = \sum_{t\in\mathcal{B}|deg(t)\leq 2} a_t t, \quad \text{that is,} \quad H_{ij}^+ = \sum_{1\leq k\leq 6} a_k t_k, \tag{5}$$

where the real coefficients a_k are to be found. We need to constrain the coefficients a_k with the other requirements. The second requirement (ii), which translates into a set of inequalities on the unknown coefficients a_k, can be obtained through a brute force evaluation of H_{ij}^+ over the 2^6 points of $\mathbb{Z}_2^6$. The outcome of this evaluation is a set of inequalities expressing the second requirement (ii) (see Supplementary materials).

The last requirement (iii) can be expressed in different ways. We can, for instance, require that the absolute values of the coefficients of H_{ij}^+, with respect to the variables P_j, Q_i, $S_{i,j}$, $S_{i+1,j-1}$, $Z_{i,j}$, and $Z_{i,j+1}$, be within $[1-\varepsilon, 1+\varepsilon]$. This, together with the set of inequalities from the second requirement, define a continuous optimization problem and can be easily solved. Another option is to minimize the distance between the coefficients to one specific coefficient. The different choices of the objective function and the solution of the corresponding continuous optimization problem are presented in Supplementary materials.

Having determined the quadratic polynomial $H_{ij}^+ \in R$ satisfyies the important requirements (i, ii, and iii) above, we can now phrase our problem $(\mathcal{P}_1)$ as the equivalent quadratic unconstrained binary optimization problem $min_{\mathbb{Z}_2} \sum_{ij} H_{ij}^+$. Notice that this reduction is performed only once for all cases; it need not to be redone for different bi-primes M. Before passing the problem to the quantum annealer, we use Gröbner bases again, this time to reduce the size of the problem. In fact, what we pass to the quantum annealer is $\mathcal{H} = \sum NF_{\mathcal{B}}\left(H_{ij}^+\right)$, where NF is the normal form and $\mathcal{B}$ is now the Gröbner basis cutoff, which we discuss in the next section. The largest bi-prime number that we embedded and solved successfully using the cell algorithm is ~35 000. Table 1 presents a small sample of many bi-prime numbers M that we tested using the cell algorithm, the number of variables using both the customized reduction *CustR* (i.e., reduction explained above before pre-processing with Gröbner bases) and the window-based *GB* reduction (i.e., reduction *CustR* followed with pre-processing with Gröbner bases), the overall reduction percentage *R%*, and the embedding and solving status inside the D-Wave 2X processor *Embed*.

The column algorithm (factoring up to 200000). The total number of variables in the cost function of the previous method is $2s_ps_q$, before any pre-processing. Here we present the column-based algorithm where the

M	$p \times q$	*CustR*	*GB*	*R%*	*Embed*
31861	211×151	111	95	14	✓
34889	251×139	111	95	14	✓
46961	311×151	125	109	13	×
150419	431×349	143	125	12	×

Table 1. Reduction and embedding statistics using Cell Algorithm for a sample of bi-primes.

number of variables (before pre-processing) is bounded by $1 + s_p s_q + (s_p + s_q)\, log_2(s_p)$. Recall that here we are phrasing the factorization problem $M = pq$ as

$$(\mathcal{P}_2)\text{: } min_{P_1,\ldots,P_{sp},Q_1,\ldots,Q_{sp},Z_{12},Z_{23},Z_{24},\ldots\in\mathbb{Z}_2} \sum_i H_i^2, \tag{6}$$

where H_i, for $1 \le i \le s_p$, is

$$H_i = \sum_{j=0}^{sq} Q_j P_{i-j} + \sum_{j=1}^{i} Z_{j,i} - m_i - \sum_{j=1}^{L_i} 2^{j-i} Z_{i,i+j}\ (Q_0 = P_0 = m_0 = 1,\ L_i = s_q + 1 + i - m_i). \tag{7}$$

The cost function is of degree 4 and, in order to use quantum annealing, it must be replaced with a positive quadratic polynomial with the same global minimum. The idea is to replace the quadratic terms $Q_j P_{i-j}$ inside the different H_i with new binary variables $W_{i-j,j}$, and add the penalty $(Q_j P_{i-j} - W_{i-j,j})^+$ to the cost function (now written in terms of the variables $W_{i-j,j}$). To find $(Q_j P_{i-j} - W_{i-j,j})^+$, we run Gröbner bases computation on the system

$$\begin{cases} Q_j P_{i-j} - W_{i-j,j}, \\ Q_j^2 - Q_j, \\ P_{i-j}^2 - P_{i-j}, \\ W_{i-j,j^2} - W_{i-j,j}. \end{cases} \tag{8}$$

Following the same steps as in the previous section, we get

$$(Q_j P_{i-j} - W_{i-j,j})^+ = a(Q_j W_{i-j,j} - W_{i-j,j}) + b(P_{i-j} W_{i-j,j} - W_{i-j,j}) + c(P_{i-j} Q_j - W_{i-j,j}), \tag{9}$$

with $a, b, c \in \mathbb{R}$ such that $-a-b-c>0$, $-b-c>0$, $-a-c>0$, $c>0$ (e.g., $c=1$, $a=b=-2$). The new cost function is now

$$\mathcal{H} = \sum_i H_i(W)^2 + \sum_{ij} (Q_j P_{i-j} - W_{i-j,j})^+. \tag{10}$$

We can obtain a better Hamiltonian by pre-processing the problem before applying the W transformation. Indeed, let us first fix a positive integer cutoff $\le (s_p + s_q + 1)$ and let $\mathcal{B} \subset \mathbb{R}[P_1, \ldots, P_{s_p}, Q_1, \ldots, Q_{s_q}, Z_{12}, Z_{23}, Z_{24} \ldots]$ be a Gröbner basis of the set of polynomials

$$\{H_i\}_{i=1\ldots\text{cutoff}} \bigcup \{P_i(P_i - 1),\ Q_i(Q_i - 1),\ Z_{ij}(Z_{ij} - 1)\}_{i,j}. \tag{11}$$

In practice, the cutoff is determined by the size of the maximum subsystem of polynomials H_i on which one can run a Gröbner basis computation; it is defined by the hardware. We also define a cutoff on the other tail of $\{H_i\}$, that is, we consider $\{H_i\}_{i=2\text{ndcutoff}\ldots(s_p+s_q+1)}$. Notice that here we are working on the original H_i rather than the new $H_i(W)$. This is because we would like to perform the replacement $Q_j P_{i-j} \to W_{i-j,j}$ after the pre-processing (some of the quadratic terms might be simplified by this pre-processing). Precisely, what we pass to the quantum annealer is the quadratic positive polynomial

$$\mathcal{H} = \sum \left(\text{NF}_{W_{i-j,j} - \text{LT}\left(\text{NF}_{\mathcal{B}_c}\left(Q_j P_{i-j}\right)\right)} \left(\text{NF}_{\mathcal{B}_c}(H_i)\right) \right)^2 + \sum_{ij} \left(W_{i-j,j} - \text{LT}\left(\text{NF}_{\mathcal{B}_c}\left(Q_j P_{i-j}\right)\right) \right)^+. \tag{12}$$

Here LT stands for the leading term with respect to the graded reverse lexicographic order. The second summation is over all i and j such that $\text{LT}(\text{NF}_{\mathcal{B}}(Q_j P_{i-j}))$ is still quadratic. The outer normal form in the first summation refers to the replacement $\text{LT}(\text{NF}_{\mathcal{B}}(Q_j P_{i-j})) \to W_{i-j,j}$, which is again performed only if $\text{LT}(\text{NF}_{\mathcal{B}}(Q_j P_{i-j}))$ is still quadratic.

The columns of Table 2 present: a small sample of many bi-prime numbers that we tested and their prime factors, the number of variables using each of a naïve polynomial-to-quadratic transformation tool *P2Q* written mostly based on the algorithm discussed in ref. 19 (Other degree reduction procedures are discussed in refs 20–23). Our novel polynomial-to-quadratic transformation *CustR*, and our window-based reduction *GB* after applying pre-processing. The overall reduction percentage *R%* and the embedding and solving status in the

M	$p \times q$	*P2Q*	*CustR*	*GB*	*R*	*Embed*
150419	431×349	116	86	73	37	✓
151117	433×349	117	88	72	38	✓
174541	347×503	117	86	72	38	✓
200099	499×401	115	89	75	35	✓
223357	557×401	125	96	80	36	×

Table 2. Reduction and embedding statistics using Column Algorithm for a sample of bi-primes.

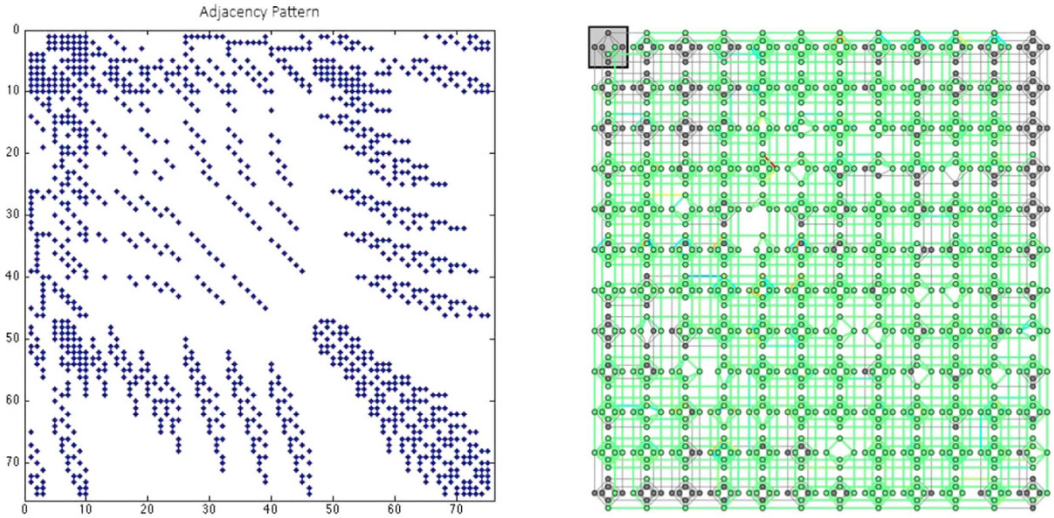


Figure 1. The column algorithm: the adjacency matrix pattern (left) and embedding into the the D-Wave 2X quantum processor (right) of the quadratic binary polynomial for $M = 200099$.

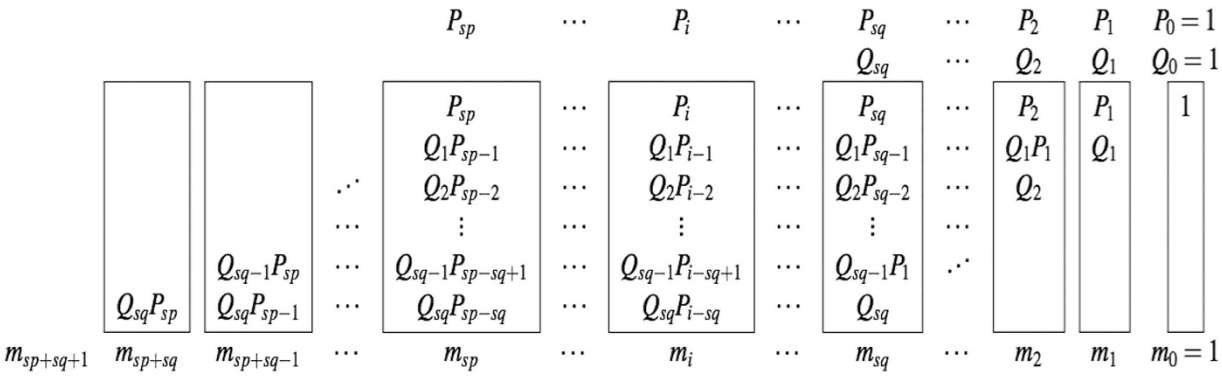


Figure 2. The column algorithm: the adjacency matrix pattern (left) and embedding into the the D-Wave 2X quantum processor (right) of the quadratic binary polynomial for $M = 200099$.

D-Wave 2X processor *Embed* are also shown. Figure 1 shows the adjacency matrix of the corresponding positive quadratic polynomial graph H and its embedded pattern inside the Chimera graph of the D-Wave 2X processor for one of the bi-primes. Details pertaining to the use of the hardware can be found in Supplementary materials.

Discussion

In this work, factorization is connected to quantum annealing through binarization of the long multiplication. The algorithm is autonomous in the sense that no a priori knowledge, or manual or ad hoc pre-processing, is involved. We have attained the largest bi-prime factored to date using a quantum processor, though more-subtle connections might exist. A future direction that this research can take is to connect factorization (as an instance of the abelian hidden subgroup problem), through Galois correspondence, to covering spaces and thus to covering graphs and potentially to quantum annealing. We believe that more-rewarding progress can be made through the investigation of such a connection.

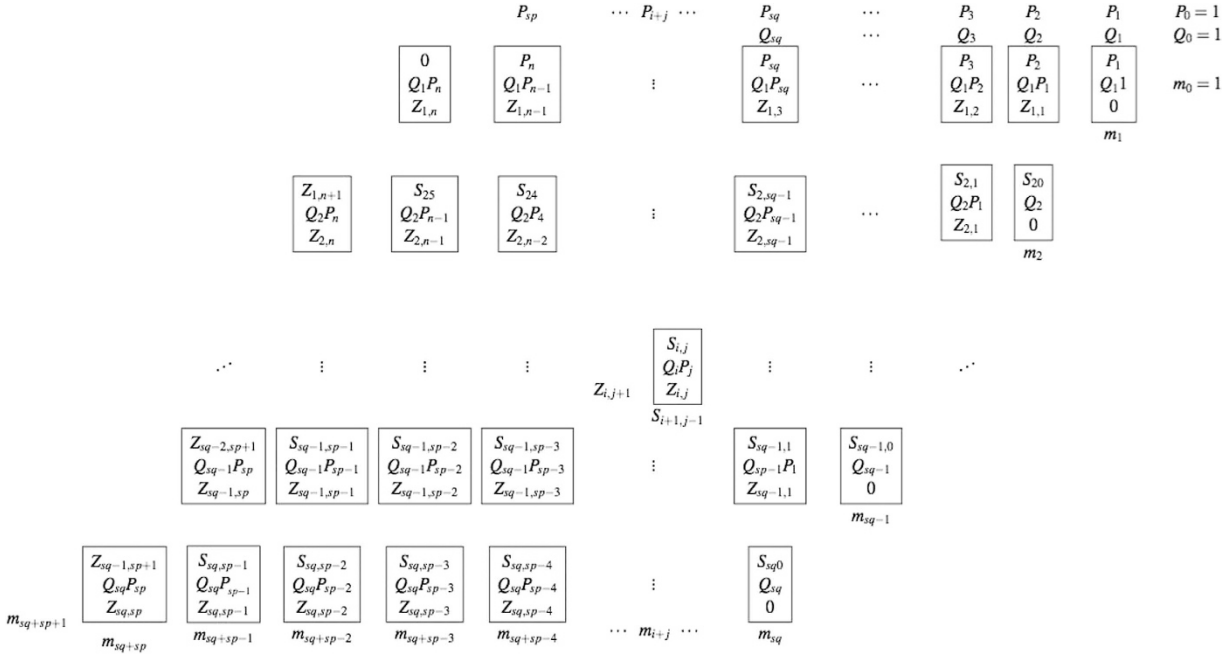


Figure 3. The column algorithm: the adjacency matrix pattern (left) and embedding into the the D-Wave 2X quantum processor (right) of the quadratic binary polynomial for $M = 200099$.

Methods

Column factoring procedure. Here we discuss the two single-bit multiplication methods of the two primes p and q. The first method generates a Hamiltonian for each of the columns of the long multiplication expansion, while the second method generates a Hamiltonian for each of the multiplying cells in the long multiplication expansion. The column factoring procedure initially introduced in ref. 9, has been generalized. The generalized column factoring procedure of $p = 2^{sp}P_{sp} + 2^{sp-1}P_{sp-1} + \cdots + 2P_1 + 1$ and $q = 2^{sq}Q_{sq} + 2^{sq-1}Q_{sq-1}\cdots + 2Q_1 + 1$ is depicted Figure 2.

The equation for an arbitrary column (i) can be written as the sum of the column's multiplication terms (above) plus all previously generated carry-on terms from lower significant columns ($j < i$). This sum is equal to the column's bi-prime term m_i plus the carry-ons generated from higher significant columns. The polynomial equation for the i-th column is

$$\sum_{j=0}^{sq} Q_j P_{i-j} + \sum_{j=1}^{i} Z_{j,i} = m_i + \sum_{j=1}^{L_i} 2^{j-i} Z_{i,i+j} \ (Q_0 = P_0 = m_0 = 1). \tag{13}$$

The above equation is used as the main column procedure's equation H_i. The Hamiltonian generation and reduction is discussed in detail in *Results*.

Cell factoring procedure. In the cell multiplication procedure the ultimate goal is to break each of the column equations discussed above into multiple smaller equations so that each equation contains only one quadratic term. This not only simplifies the generation of quadratic Hamiltonians, but also generates Hamiltonians with more-uniform quadratic coefficients in comparison to the column procedure. We generalized the procedure initially introduced in ref. 10. Figure 3 depicts our generalization:

Each cell contains one of the total $(s_p - 1)(s_q - 1)$ quadratic terms in the form of Q_iP_j. To chain a cell to its upper cell, one extra sum variable $S_{i,j}$ is added. Also, each carry-on variable $Z_{i,j}$ in a cell is the carry-on of the cell directly to its right, so each cell contains four variables. The sum of three terms Q_iP_j, $S_{i,j}$, and $Z_{i,j}$ is at most 3; thus, it generates an additional sum variable $S_{i+1,j-1}$ and one carry-on variable $Z_{i,j+1}$. Therefore, the equation for an arbitrary cell indexed (i, j), shown in the centre of the above table, is

$$S_{i,j} + Q_iP_j + Z_{i,j} = S_{i+1,j-1} + 2Z_{i,j+1}. \tag{14}$$

As we can see, only six binary variables are involved in each cell equation and the equation contains one quadratic term, so it can be transformed into a positive Hamiltonian without adding slack variables. The Hamiltonian generation and reduction procedure is discussed in detail in *Results*.

Gröbner bases. Good references for the following definitions are chapters 1 and 2 of ref. 1 and chapter 1 of ref. 24.

Normal forms. A normal form is the remainder of Euclidean divisions in the ring of polynomials $\mathbb{R}[x_1, \ldots, x_n]$. Precisely, the normal form of a polynomial $f \in \mathbb{R}[x_1, \ldots, x_n]$, with respect to the set of polynomials $\mathcal{B} \subset \mathbb{R}[x_1, \ldots, x_n]$ (usually a Gröbner basis), is the polynomial $\mathrm{NF}(f) \in \mathbb{R}[x_1, \ldots, x_n]$, which is the image of f modulo $\mathcal{B}$. It is the remainder of the Euclidean of f by all $g \in \mathcal{B}$.

Term orders. A term order on $\mathbb{R}[x_1, \ldots, x_n]$ is a total order $\prec$ on the set of all monomials $x^a = x_1^{a_1} \ldots x_n^{a_n}$, which has the following properties: (1) if $x^a \prec x^b$, then $x^{a+c} \prec x^{b+c}$ for all positive integers a, b, and c; (2) $1 \prec x^a$ for all strictly positive integers a. An example of this is the pure lexicographic order $plex(x_1, \ldots, x_n)$. Monomials are compared first by their degree in x_1, with ties broken by degree in x_2, etc. This order is usually used in eliminating variables. Another example, is the graded reverse lexicographic order $tdeg(x_1, \ldots, x_n)$. Monomials are compared first by their total degree, with ties broken by reverse lexicographic order, that is, by the smallest degree in x_n, x_{n-1}, etc.

Gröbner bases. Given a term order $\prec$ on $\mathbb{R}[x_1, \ldots, x_n]$, then by the leading term (initial term) LT of f we mean the largest monomial in f with respect to $\prec$. A (reduced) Gröbner basis to the ideal $\mathcal{I}$ with respect to the ordering $\prec$ is a subset $\mathcal{B}$ of $\mathcal{I}$ such that: (1) the initial terms of elements of $\mathcal{B}$ generate the ideal $\mathrm{LT}(\mathcal{I})$ of all initial terms of $\mathcal{I}$; (2) for each $g \in \mathcal{B}$, the coefficient of the initial term of g is 1; (3) the set $\mathrm{LT}(g)$ minimally generates $\mathrm{LT}(\mathcal{I})$; and (4) no trailing term of any $g \in \mathcal{B}$ lies in $\mathrm{LT}(\mathcal{I})$. Currently, Gröbner bases are computed using sophisticated versions of the original Buchberger algorithm, for example, the F4 and F5 algorithms by J. C. Faugère[25,26].

Factorization as an eigenvalue problem. In this section, for completeness, we describe how the factorization problem can be solved using eigenvalues and eigenvectors. This is an adaptation of the method presented in ref. 8 to factorization, which is itself an adaption to real polynomial optimization of the method of solving polynomial equations using eigenvalues in ref. 1.

Let $\mathcal{H}$ be in $\mathbb{R}[x_1, \ldots, x_n]$ as in (12), where we have used the notation x_i instead of the P_s, Q_s, Z_s, and W_s. Define

$$\mathcal{H}_\alpha := \mathcal{H} + \sum_i \alpha_i x_i (x_i - 1), \tag{15}$$

which is in the larger ring $\mathbb{R}[x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_n]$. We also define the set of polynomials

$$\mathcal{C} = \left\{ \partial H_\alpha / \partial_{x_1}, \ldots, \partial H_\alpha / \partial_{x_n}, \partial H_\alpha / \partial_{\alpha_1}, \ldots, \partial H_\alpha / \partial_{\alpha_n} \right\}. \tag{16}$$

The variety $\mathcal{V}(\mathcal{C})$ is the set of all binary critical points of $\mathcal{H}$. Its coordinates ring is the residue algebra $A := \mathbb{R}[x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_n]/\mathcal{C}$. We need to compute a basis for A. This is done by first computing a Gröbner basis for $\mathcal{C}$ and then extracting the standard monomials (i.e., the monomials in $\mathbb{R}[x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_n]$ that are not divisible by the leading term of any element in the Gröbner basis). In the simple example below, we do not need to compute any Gröbner basis since $\mathcal{C}$ is a Gröbner basis with respect to $plex(\alpha, x)$. We define the linear map

$$\begin{aligned} m_{\mathcal{H}_\alpha} : \; & A \to A \\ & g \mapsto \mathcal{H}_\alpha g \end{aligned} \tag{17}$$

Since the number of critical points is finite, the algebra A is always finite-dimensional by the Finiteness Theorem (page 39 of ref. 1). Now, the key points are:

- The value of $\mathcal{H}_\alpha$ (i.e., values of $\mathcal{H}$), on the set of critical points $\mathcal{V}(\mathcal{C})$, are given by the eigenvalues of the matrix $m_{\mathcal{H}_\alpha}$.
- Eigenvalues of m_{x_i} and m_{α_i} give the coordinates of the points of $\mathcal{V}(\mathcal{C})$.
- If v is an eigenvector for $m_{\mathcal{H}_\alpha}$, then it is also an eigenvector for m_{x_i} and m_{α_i} for $1 \le i \le n$.

We illustrate this in an example. Consider $M = pq = 5 \times 3$ and let

$$\mathcal{H} = 2 + 7x_4 + 2x_3 + 2x_4x_3 - 2x_3x_2 - x_1 - 4x_4x_1 - 2x_3x_1 + x_2x_1 \tag{18}$$

be the corresponding Hamiltonian as in (12), where $x_1 = p_2$, $x_2 = q_1$, $x_3 = w_{2,1}$, and $x_4 = z_{2,3}$. A basis for the residue algebra A is given by the set of the 16 monomials

$$\{1, x_4, x_3, x_4x_3, x_2, x_4x_2, x_3x_2, x_3x_2x_4, x_1, x_4x_1, x_3x_1, x_1x_3x_4, x_2x_1, x_4x_1x_2, x_1x_3x_2, x_1x_3x_2x_4\}. \tag{19}$$

The matrix $m_{\mathcal{H}_\alpha}$ is

$$m_{\mathcal{H}_\alpha} := \begin{bmatrix}
2 & 7 & 2 & 2 & 0 & 0 & -2 & 0 & -1 & -4 & -2 & 0 & 1 & 0 & 0 & 0 \\
0 & 9 & 0 & 4 & 0 & 0 & 0 & -2 & 0 & -5 & 0 & -2 & 0 & 1 & 0 & 0 \\
0 & 0 & 4 & 9 & 0 & 0 & -2 & 0 & 0 & 0 & -3 & -4 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 13 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & -7 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 2 & 7 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & -4 & -2 & 0 \\
0 & 0 & 0 & 0 & 0 & 9 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & -4 & 0 & -2 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & -4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 0 & 2 & 1 & 0 & -2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 2 & 0 & 1 & 0 & -2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 5 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & -2 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5
\end{bmatrix}$$

We expect the matrix's smallest eigenvalue to be zero and, indeed, we get the following eigenvalues for $m_{\mathcal{H}_\alpha}$:

$$\{0, 1, 2, 4, 5, 6, 9, 11, 13\}. \tag{20}$$

This is also the set of values which $\mathcal{H}_\alpha$ takes on $\mathcal{V}(\mathcal{C})$. The eigenvector v which corresponds to the eigenvalue 0 is the column vector

$$v := (1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0)^T. \tag{21}$$

This eigenvector is used to find the coordinates of $\hat{x} \in \mathcal{V}(\mathcal{C})$ that cancel (minimize) $\mathcal{H}_\alpha$. The coordinates of the global minimum $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_n)$ are defined by $m_{x_i} v = \hat{x}_i v$, and this gives $x_1 = x_2 = x_3 = 1$, $x_4 = 0$, and $\alpha_1 = 2\alpha_2 = \alpha_3 = 2$, $\alpha_4 = 5$.

References

1. Cox, D. A., Little, J. B. & O'Shea, D. *Using algebraic geometry*. Graduate texts in mathematics (Springer, New York, 1998).
2. Kadowaki, T. & Nishimori, H. Quantum annealing in the transverse ising model. *Phys. Rev. E* **58,** 5355–5363 (1998).
3. Farhi, E. *et al.* A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science* **292,** 472–475 (2001).
4. Das, A. & Chakrabarti, B. K. *Colloquium*: Quantum annealing and analog quantum computation. *Rev. Mod. Phys.* **80,** 1061–1081 (2008).
5. Johnson, M. W. *et al.* Quantum annealing with manufactured spins. *Nature* **473,** 194–198 (2011).
6. Boixo, S., Albash, T., Spedalieri, F. M., Chancellor, N. & Lidar, D. A. Experimental signature of programmable quantum annealing. *Nat Commun* **4** (2013).
7. Lanting, T. *et al.* Entanglement in a quantum annealing processor. *Phys. Rev. X* **4,** 021041 (2014).
8. Parrilo, P. A. & Sturmfels, B. Minimizing polynomial functions. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* (2001).
9. Burges, C. Factoring as optimization. Tech. Rep. MSR-TR-2002-83, Microsoft Research (2002).
10. Schaller, G. & Schutzhold, R. The role of symmetries in adiabatic quantum algorithms. *Quantum Information & Computation* **10,** 109–140 (2010).
11. Xu, N. *et al.* Quantum factorization of 143 on a dipolar-coupling nuclear magnetic resonance system. *Phys. Rev. Lett.* **108,** 130501 (2012).
12. Dattani, N. S. & Bryans, N. Quantum factorization of 56153 with only 4 qubits. *arXiv:1411.6758* (2014).
13. Pal, S., Moitra, S., Anjusha, V. S., Kumar, A. & Mahesh, T. S. Hybrid scheme for factorization: Factoring 551 using a 3-qubit NMR quantum adiabatic processor. *arXiv:1611.00998* [*quant-ph*] (2016).
14. Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **26,** 1484–1509 (1997).
15. Kitaev, A. Quantum measurements and the abelian stabilizer problem. *arXiv:9511026* (1995).
16. Monz, T. *et al.* Realization of a scalable shor algorithm. *Science* **351,** 1068–1070 (2016).
17. Smolin, J. A., Smith, G. & Vargo, A. Oversimplifying quantum factoring. *Nature* **499,** 163–165 (2013).
18. Raussendorf, R. Contextuality in measurement-based quantum computation. *Phys. Rev. A* **88,** 022322 (2013).
19. Boros, E. & Aritanan, G. On quadratization of pseudo-boolean functions. *arXiv:1404.6538* (2014).
20. Anthony, M., Boros, E., Crama, Y. & Gruber, A. Quadratization of symmetric pseudo-boolean functions. *Discrete Applied Mathematics* **203,** 1–12 (2016).
21. Babbush, R., O'Gorman, B. & Aspuru-Guzik, A. Resource efficient gadgets for compiling adiabatic quantum optimization problems. *Annalen der Physik* **525,** 877–888 (2013).
22. Babbush, R., Denchev, V. S., Ding, N., Isakov, S. & Neven, H. Construction of non-convex polynomial loss functions for training a binary classifier with quantum annealing. *CoRR* **abs/1406.4203** (2014).
23. Tanburn, R., Okada, E. & Dattani, N. S. Reducing multi-qubit interactions in adiabatic quantum computation without adding auxiliary qubits. part 1: The "deduc-reduc" method and its application to quantum factorization of numbers. *arXiv:1508.04816* (2015).
24. Sturmfels, B. *Gröbner bases and convex polytopes*, vol. 8 of *University Lecture Series* (American Mathematical Society, Providence, RI, 1996).
25. Faugére, J.-C. A new efficient algorithm for computing Gröbner bases (f4). *Journal of Pure and Applied Algebra* **139,** 61–88 (1999).
26. Faugère, J. C. A new efficient algorithm for computing Gröbner bases without reduction to zero (f5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, ISSAC '02, 75–83 (ACM, New York, NY, USA, 2002).

Acknowledgements

We appreciate discussions with Pooya Ronagh and thank Marko Bucyk for proofreading the manuscript.

Author Contributions

R.D. and H.A. designed the algorithms. R.D. and H.A. conceived the experiments and analysed the results. All authors wrote and reviewed the manuscript.

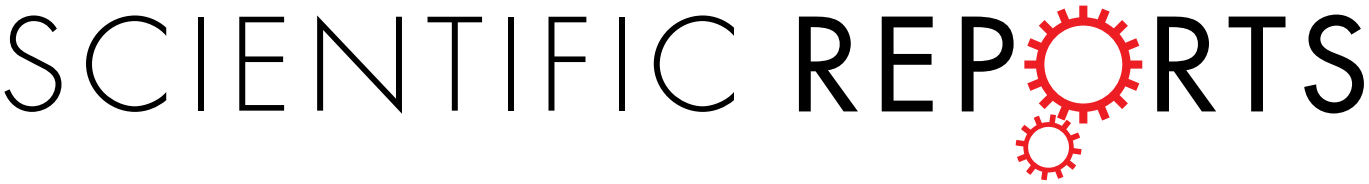
Additional Information

Supplementary information accompanies this paper at http://www.nature.com/srep

Competing financial interests: The authors declare no competing financial interests.

How to cite this article: Dridi, R. and Alghassi, H. Prime factorization using quantum annealing and computational algebraic geometry. *Sci. Rep.* **7**, 43048; doi: 10.1038/srep43048 (2017).

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



OPEN

Erratum: Prime factorization using quantum annealing and computational algebraic geometry

Raouf Dridi & Hedayat Alghassi

Scientific Reports 7:43048; doi: 10.1038/srep43048; published online 21 February 2017; updated on 22 March 2017

In this Article, the legend of Figure 2 is incorrect:

"The column algorithm: the adjacency matrix pattern (left) and embedding into the the D-Wave 2X quantum processor (right) of the quadratic binary polynomial for M = 200099".

Should read:

"Column factoring procedure".

In addition, the legend of Figure 3 is incorrect:

"The column algorithm: the adjacency matrix pattern (left) and embedding into the the D-Wave 2X quantum processor (right) of the quadratic binary polynomial for M = 200099".

Should read:

"Cell factoring procedure".