Overreliance on Classical Computing in Quantum Factorization

Andrew Brockmann

University of Wisconsin, Madison
Computer Sciences Department
abrockmann@wisc.edu

July 13, 2018

Abstract

A 2012 quantum experiment factored 143 after performing some simplifications classically. Further research demonstrated that that experiment arguably performed the quantum factorizations of other numbers too, such as 56153. This paper characterizes the numbers factored by the 2012 experiment, demonstrates that there are infinitely many of these numbers if the Bateman-Horn conjecture is correct, and provides $N_{2000} \approx 7.86 \cdot 10^{2000}$ as an explicit example. Finally, we show that, in asymptotic terms, most of the work in these factorizations was done classically. These quantum factorizations therefore do not seem to indicate progress toward factoring large RSA moduli.

1 Introduction

A strong quantum computer using Shor's algorithm would be able to efficiently factor large numbers, thereby breaking the RSA cryptosystem. RSA moduli tend to be on the order of thousands of bits, with 1024, 2048, and 4096 being common bit lengths. The progress of quantum computers toward breaking RSA in practice can be measured by keeping track of the largest number experimentally factored using Shor's algorithm. Progress on this front has been slow—to date, the largest number factored using Shor's algorithm is 21, and even this feat was accomplished using some prior knowledge of the solution [11].

In 2001, Burges presented a reformulation of factoring as an unconstrained optimization problem [3]. Building on further work by Schaller and Schützhold [10] and Farhi et. al. [5], Xu et. al. [12] used Burges' idea to perform an actual quantum factorization. Specifically, they started with a system of equations encoding the factoring problem for $N = 143$, performed some simplifications classically, and then used a quantum computer to solve the resulting system.

In 2014, two years later, Dattani and Bryans [4] revealed a new insight into this factoring method. While different values of N lead to different systems of equations and hence require different simplifications, the resulting simplified systems of equations may nevertheless be the same for different N. In particular, the factoring equations for 56153 simplify to the same system as do the factoring equations for 143. Since the solution of the simplified system constitutes the quantum portion of the factoring algorithm proposed by Xu et. al., their quantum factorization of 143 arguably also factored 56153 and other numbers.

The organization of this paper is as follows. We begin with a closer look at the existing literature surrounding these quantum factorizations. Several previous factorizations (such as those of 143 and 56153) used a hybrid algorithm with a classical and a quantum phase; we state the steps of this algorithm explicitly and prove its correctness when used to factor products of two distinct primes.

Next, we examine which—and how many—values of N were "factored" by Xu et. al. Conditional on the Bateman-Horn conjecture, we prove that infinitely many N were factored. One such value, $N_{2000} \approx 7.86 \cdot 10^{2000}$, is given explicitly in the appendix. This is arguably a new record for quantum factorization.

Lastly, we study this factoring method in more detail. We prove unconditionally that this algorithm can factor infinitely many N with an 18 qubit quantum computer. However, we also show that the quantum phase of the algorithm can be performed in polynomial time classically, meaning that the classical phase cannot be done efficiently unless factoring is in P.

2 The Factoring Equations

In "Factoring as Optimization", Burges [3] reformulates the factoring problem in terms of the "factoring equations" obtained by writing factors p and q as bit sequences and performing longhand multiplication. For example, Xu et. al. begin with the following multiplication table for $N = 143$:

	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			
	$z_{6,7}$	$z_{5,6}$	$z_{4,5}$	$z_{3,4}$	$z_{2,3}$	$z_{1,2}$		
	$z_{5,7}$	$z_{4,6}$	$z_{3,5}$	$z_{2,4}$				
$p \times q = 143$	1	0	0	0	1	1	1	1

Here, each p_i is the ith bit of p (i.e. the coefficient of 2^i in the binary expansion of p) and likewise for the q_i. Similarly, each $z_{i,j}$ is a binary value representing the carry bit from column i to column j. Note that there are several simplifications implicit in this table. First, the least significant bits of both p and q have been set to 1, since any factor of an odd product must be odd. Second, it is assumed that the most significant bit of each of p and q occurs at index 3. This assumption is unjustified, but guessing where the most significant bits of the factors occur only incurs $O(\log N)$ overhead in general. An alternative is to use $\lfloor \log N \rfloor$ variables for each of p and q. (Although N has bit length $\lfloor \log N \rfloor + 1$, the factors p and q must have shorter bit lengths than N in any nontrivial factorization.) This does, however, lead to a more complicated system.

From the multiplication table for any particular N, one can then derive the factoring equations by summing over the individual columns of the table. For $N = 143$, this yields:

$$\begin{aligned}
p_1 + q_1 &= 1 + 2z_{1,2} \\
p_2 + p_1q_1 + q_2 + z_{1,2} &= 1 + 2z_{2,3} + 4z_{2,4} \\
1 + p_2q_1 + p_1q_2 + 1 + z_{2,3} &= 1 + 2z_{3,4} + 4z_{3,5} \\
q_1 + p_2q_2 + p_1 + z_{3,4} + z_{2,4} &= 0 + 2z_{4,5} + 4z_{4,6} \\
q_2 + p_2 + z_{4,5} + z_{3,5} &= 0 + 2z_{5,6} + 4z_{5,7} \\
1 + z_{5,6} + z_{4,6} &= 0 + 2z_{6,7} \\
z_{6,7} + z_{5,7} &= 1
\end{aligned}$$

Burges defines a *factoring potential* which is minimized only for integral solutions to the factoring equations [3]. He begins by shifting terms in the equations so that each has right hand side 0; the first equation, for example, becomes

$$p_1 + q_1 - 1 - 2z_{1,2} = 0.$$

The left hand side of each rewritten equation is then squared and added to the factoring potential. For instance, the potential term corresponding to the first equation is

$$(p_1 + q_1 - 1 - 2z_{1,2})^2,$$

which achieves its minimum value of 0 if and only if the equation $p_1 + q_1 - 1 - 2z_{1,2} = 0$ is satisfied. Then, for each variable w appearing in the factoring equations, the term $w^2(1-w)^2$ is added to the potential. Each such term attains its minimum value of 0 if and only if the corresponding variable w is binary. In essence, the factoring potential incurs a positive penalty for each unsatisfied factoring equation and for each nonbinary variable. All terms of the potential achieve their individual minima of 0—and hence, the potential achieves its global minimum of 0—precisely for binary solutions to the factoring equations, and these solutions are in one-to-one correspondence with the pairs $(p, q) \in \mathbb{N}^2$ satisfying $N = pq$.

Finally, Burges allows for the individual terms of the factoring potential to be weighted by some positive coefficients α_i, so that some constraint violations may be more costly than others [3]. A valid factorization of N into two factors can then be found by minimizing the factoring potential.

3 Classical/Quantum Hybrid Algorithm

Later, Schaller and Schützhold [10] showed how to adapt the Burges method into an adiabatic quantum algorithm. One qubit is used for each variable in the factoring equations, and the four-body interactions (corresponding to quartic terms in the factoring potential) can be eliminated to yield a Hamiltonian with at most three-body interactions between qubits.

Xu et. al. [12] sought to perform an actual quantum factorization using the adiabatic method of Schaller and Schützhold. However, the factoring equations for $N = 143$ (given above) have 14 variables, meaning that at least 14 qubits would be needed. As this was beyond the capabilities of existing quantum computers, they performed some simplifications classically on the factoring equations to obtain a system with only 4 variables. For example, in the equation $p_1 + q_1 = 1 + 2z_{1,2}$, we must have $z_{1,2} = 0$, or else the equation is unsatisfiable over binary values. The first equation therefore simplifies to $p_1 + q_1 = 1$, meaning that one of p_1 and q_1 is 0 and the other is 1. Regardless of which is which, their product, $p_1 q_1$, must then be 0, thereby simplifying the second equation. Performing similar simplifications to the whole system,

most variable values are determined and we are left with just three equations:

$$\begin{aligned} p_1 + q_1 &= 1 \\ p_2 + q_2 &= 1 \\ p_2 q_1 + p_1 q_2 &= 1 \end{aligned}$$

Xu et. al. constructed the appropriate Hamiltonian for the simplified system and finished the 4 qubit factorization of 143.

Dattani and Bryans [4] noticed that the factoring equations for other values of N also redcuce to the same 4 variable system obtained by Xu et. al. after some classical simplification. For example, after constructing the multiplication table for $N = 56153$, they reduced the resulting factoring equations to

$$\begin{aligned} p_3 + q_3 &= 1 \\ p_4 + q_4 &= 1 \\ p_4 q_3 + p_3 q_4 &= 1 \end{aligned}$$

This is just the reduced system for $N = 143$ but with indices 1 and 2 swapped out for 3 and 4. Therefore, Xu et. al. arguably also factored 56153, albeit unwittingly. More precisely, we have a classical/quantum hybrid algorithm, and Xu et. al. performed the quantum portion of the factorization for 143, 56153, and other numbers.

Dattani and Bryans [4] went further by partially characterizing the simplified factoring equations' dependence on the factors p and q. In particular, if p and q differ only on the two bits indexed by a and b, then the factoring equations for $N = pq$ will simplify to

$$\begin{aligned} p_a + q_a &= 1 \\ p_b + q_b &= 1 \\ p_b q_a + p_a q_b &= r, \end{aligned}$$

where r is a binary value dependent on N. Furthermore, Dattani and Bryans [4] reduce the factoring equations for $N = 291311$ to:

$$\begin{aligned} p_1 + q_1 &= 1 \\ p_2 + q_2 &= 1 \\ p_5 + q_5 &= 1 \\ p_1 q_2 + q_1 p_2 &= 1 \\ p_2 q_5 + q_2 p_5 &= 0 \\ p_5 q_1 + q_5 p_1 &= 1 \end{aligned}$$

Notably, in the factorization of $291311 = 557 \times 523$, the two factors differ precisely on bits 1, 2, and 5. These examples suggest that the factoring equations in general can be reduced to a common form:

Definition 1: A *semiprime* is a product of two distinct prime numbers. The *reduced factoring system* for a semiprime $N = pq$ is the system

$$\begin{aligned} p_{i_1} + q_{i_1} &= 1 \\ p_{i_2} + q_{i_2} &= 1 \\ &\vdots \quad \vdots \quad \vdots \\ p_{i_k} + q_{i_k} &= 1 \\ p_{i_1} q_{i_2} + p_{i_2} q_{i_1} &= r_1 \\ p_{i_2} q_{i_3} + p_{i_3} q_{i_2} &= r_2 \\ &\vdots \quad \vdots \quad \vdots \\ p_{i_k} q_{i_1} + p_{i_1} q_{i_k} &= r_k, \end{aligned}$$

where $i_1, \ldots, i_k$ are the bit indices on which p and q differ and the r_i are binary values dependent on N.

Note that this form is consistent with the reduced systems we have seen thus far. Although the reduced system for 143 only has three equations, the last equation in our general form is redundant in this case.

A natural question is whether the factoring equations for general N can be simplified to the reduced factoring system for N. The answer is yes, and we can prove this in several steps.

Theorem 1: Let N be a semiprime. Any solution to the factoring equations for N also satisfies the corresponding reduced factoring system for some binary values $r_1, \ldots, r_k$ depending only on N.

Proof. Let S be a solution to the factoring equations for N. The i_j are the indices on which p and q differ, so for each i_j, one of p_{i_j} and q_{i_j} is 1 and the other is 0. This means that their sum is 1, hence the first k equations of the reduced factoring system are necessarily true of S.

Since one of p_{i_1} and q_{i_1} is 0, at least one of the terms $p_{i_1} q_{i_2}$ and $p_{i_2} q_{i_1}$ is 0. The other term, whichever it is, is a product of binary values and is therefore itself a binary value. So there is indeed a binary value r_1 such that $p_{i_1} q_{i_2} + p_{i_2} q_{i_1} = r_1$. By the same reasoning, the equations with $r_2, \ldots, r_k$ also hold true over S for some binary values of the r_i.

It is necessary that any other solution to the factoring equations should satisfy the reduced factoring system for the same values of the r_i. Since N is semiprime, there is only one other solution S', obtained from S by swapping p and q. The equations $p_{i_j} + q_{i_j} = 1$ must still hold true for S' because swapping the values of p_{i_j} and q_{i_j} does not change their sum.

The final k equations of the reduced factoring system are also satisfied by S' because their left hand sides do not change when p and q are swapped: for each left hand sum, exchanging the values of each pair p_{i_j}, q_{i_j} yields the same sum with different term order.

Therefore, there are binary valaues $r_1, \ldots, r_k$ such that both solutions to the factoring equations of N satisfy the reduced factoring system. □

This alone does not prove that the factoring equations can be simplified to the reduced factoring system, since it leaves open the possibility that the reduced system has other solutions not corresponding to any factoring solution. Our next theorem precludes this possibility.

Theorem 2: The reduced factoring system for a semiprime N has precisely two solutions.

Proof. From Theorem 1, we know that the two solutions to the factoring equations for N serve as solutions for the reduced factoring system. These solutions are obtained from one another by exchanging the values of the p_i and q_i. We have $p_{i_j} \neq q_{i_j}$ for all i_j indexed in the reduced system, so swapping p and q in one solution produces a solution that differs in the p_{i_j} and q_{i_j}. Thus, the two solutions to the factoring equations generate distinct solutions to the reduced factoring system. Call these solutions S and S'.

Now suppose S'' is a third solution to the reduced factoring system. S'' must differ from S on the value of at least one of the p_{i_j} or q_{i_j}; without loss of generality, suppose the solutions differ on p_{i_1}. All solutions to the reduced factoring system must satisfy

$$p_{i_1} q_{i_2} + p_{i_2} q_{i_1} = r_1.$$

Using $p_{i_j} + q_{i_j} = 1$, we can substitute $q_{i_1} = 1 - p_{i_1}$ and $q_{i_2} = 1 - p_{i_2}$ and simplify to rewrite this equation as

$$p_{i_1} + p_{i_2} - 2p_{i_1} p_{i_2} = r_1.$$

Since $x + y - 2xy = x \oplus y$ (the exclusive OR function) for any binary x and y, this equation is the same as

$$p_{i_1} \oplus p_{i_2} = r_1.$$

For solutions S and S'' to both satisfy this equation while differing on p_{i_1}, they must also differ on p_{i_2}. Using the same reasoning as above, we find that since the two solutions differ on p_{i_2}, they must also differ on p_{i_3}. Applying this reasoning iteratively, we find that S and S'' must differ on all p_{i_j}. Hence they also differ on all q_{i_j}. But S' differs from S on all p_{i_j} and q_{i_j}, and so $S'' = S'$; any solution differing from S must be S'. Analogously, any solution differing from S' must be S. Therefore, the two solutions S and S' are the only solutions to the reduced factoring system. □

These theorems tell us that the only solutions to the reduced factoring system are those obtained from the solutions to the factoring equations. Therefore:

Corollary 1: For any semiprime N, the factoring equations and the reduced factoring system have the same solutions in the p_{i_j} and q_{i_j}.

This means that if all variables other than the p_{i_j} and q_{i_j} are determined, the remaining variable values can be determined by solving the reduced factoring system. One way to simplify to the reduced factoring system is to factor N, take note of the bits on which p and q differ, and derive the equations of the reduced system. As the reduced factoring system is a means to an end in solving the factoring problem, it is perhaps unsatisfying to simplify to the reduced system by first factoring N. However, it is sufficient for our purposes that there is a way to derive the reduced factoring system from the factoring equations. And, as we will see in section 5, this simplification probably cannot be done efficiently in general.

The simplification to a reduced factoring system can be performed for all N, not just those that are semiprime. However, in the general case, the reduced factoring system need not be unique because there may be many ways to split N into two nontrivial factors. We might also consider the factorization to be incomplete unless the full prime factorization of N is found. While the following algorithm could be made into a fully general factoring algorithm by applying it recursively to obtain the prime factorization of N, this paper will only consider N that are semiprime.

With this in mind, we can now explicitly enumerate the steps of the classical/quantum hybrid factorization algorithm suggested by [12] and [4]:

1. Produce the factoring equations for N
2. Classically simplify the factoring equations to their reduced factoring system

3. Construct the Hamiltonian corresponding to the reduced system's factoring potential

4. Optimize using a quantum computer

5. Reconstruct factors p and q, with $N = pq$, from their bit values

4 Which Numbers Were Factored?

When the hybrid algorithm is used to factor 143 and 56153, the quantum portions of the two factorizations are the same, because the factoring equations for both simplify to the same reduced factoring system:

$$\begin{aligned} p_a + q_a &= 1 \\ p_b + q_b &= 1 \\ p_b q_a + p_a q_b &= 1 \end{aligned}$$

For a number N to have been "factored" by [12], it is sufficient that the factoring equations for N reduce to this system.

But which numbers N lead to this reduced system? A clear necessary condition is that the factors p and q should differ on exactly two bit values. This condition is not sufficient, however; for some such pairs (p, q), the final equation of the reduced factoring system will instead be

$$p_b q_a + p_a q_b = 0.$$

Since the p_i and q_i are all binary values, the equation $p_a + q_a = 1$ implies that $(p_a, q_a) = (1, 0)$ or $(p_a, q_a) = (0, 1)$, and likewise for (p_b, q_b). Therefore,

$$(p_a, p_b, q_a, q_b) \in \{(0, 0, 1, 1),\ (0, 1, 1, 0),\ (1, 0, 0, 1),\ (1, 1, 0, 0)\}\,.$$

Observe that $(0, 0, 1, 1)$ and $(1, 1, 0, 0)$ lead to $p_b q_a + p_a q_b = 0$, while $(0, 1, 1, 0)$ and $(1, 0, 0, 1)$ lead to $p_b q_a + p_a q_b = 1$. This gives us a complete characterization of the numbers that were factored by [12]:

Corollary 2: The factoring equations of a semiprime $N = pq$ lead to the same reduced factoring system as 143 and 56153 if and only if:

1. p and q differ on precisely two bit values

2. The differing bit values, indexed by a and b, satisfy $(p_a, p_b, q_a, q_b) \in \{(0, 1, 1, 0),\ (1, 0, 0, 1)\}$

A special case is when p and q differ at bit indices 1 and 2, with 0 indexing the least significant bit. (Recall that all primes larger than 2 are odd and therefore must agree on the least significant bit.) In this case, the binary representations of p and q end with "011" and "101", with agreement on all higher order bits. In other words, if we arbitrarily choose $p < q$, we have

$$(p, q) = (8x + 3,\ 8x + 5)$$

for some $x \in \mathbb{N}$. Since any product N of such p and q is of the form described in Corollary 2, we see that, for any $x \in \mathbb{N}$ for which $8x + 3$ and $8x + 5$ are both prime,

$$N = (8x + 3)(8x + 5) = 64x^2 + 64x + 15$$

was factored (in the same sense as 143 and 56153) by the quantum experiment in [12].

As a consequence of Dirichlet's theorem on primes in arithmetic progressions, we know that there are infinitely many x for which $8x+3$ is prime, and likewise for $8x + 5$. It is unknown whether there are infinitely many prime pairs $(8x + 3, 8x + 5)$, although the asymptotic density of such pairs is the subject of several conjectures on prime constellations. For example:

Corollary 3: If the Bateman-Horn conjecture [2] is correct, then:

1. There are infinitely many $x \in \mathbb{N}$ for which $(8x + 3, 8x + 5)$ is a prime pair.

2. The number $P(y)$ of $x < y$ for which $8x + 3$ and $8x + 5$ are both prime is given asymptotically by

$$P(y) \sim C \int_2^y \frac{dt}{(\ln t)^2}.$$

   Here, the constant C is given by the infinite product

$$C = \prod_{p \text{ prime}} \frac{1 - N(p)/p}{(1 - 1/p)^2},$$

   where $N(p)$ is the number of solutions to $(8x+3)(8x+5) \equiv 0 \pmod{p}$. In particular, we have $C \approx 7.922$ in this case.

Contingent upon the Bateman-Horn conjecture, Xu et. al. arguably factored infinitely many N when they factored 143. Arbitrarily large N can be found by searching for sufficiently large prime pairs of the form $(8x + 3, 8x + 5)$.

Searches for large N can be sped up using some facts about divisibility. Each $(p, q) = (8x + 3, 8x + 5)$ is a pair of twin primes, and every twin prime pair—with the sole exception of $(3, 5)$—is of the form $(6j - 1, 6j + 1)$. (The polynomials $6j$, $6j + 2$, and $6j + 4$ are all even, while $6j + 3$ generates no primes other than 3 because it is divisible by 3.) It follows that each prime pair of the form $(p, q) = (8x + 3, 8x + 5)$ is furthermore of the form

$$(p, q) = (24x' + 11,\ 24x' + 13).$$

Given a target value t, one way to find a number $N \approx t$ that was factored by [12] is to iteratively generate a random value $x' \approx \sqrt{t}/24$ and check whether $24x' + 11$ and $24x' + 13$ are both prime. One such number is $N_{2000} \approx 7.86 \cdot 10^{2000}$, which has 2001 decimal digits. The full value of N as well as p and q are provided in the appendix.

Several known large prime pairs are nearly of the form described in Corollary 2. For example, one of the largest known pairs of sexy primes (i.e. primes differing by 6) is [1]:

$$\begin{aligned} p = &\quad \left(48011837012\left[(53238 \cdot 7879\#)^2 - 1\right] + 2310\right) \cdot \frac{53238 \cdot 7879\#}{385} + 1 \\ q = &\quad p + 6 \end{aligned}$$

Here, $n\#$ is the primorial function, which is the product of the primes less than or equal to n. The product $N = pq$ has more than 20000 decimal digits. Furthermore, the primes p and q differ on precisely two bits—they are of the form $(p, q) = (8x + 1, 8x + 7)$, so the binary representations of p and q end, respectively, in "001" and "111" (with agreement on all higher order bits).

However, the differing bits for this pair satisfy $(p_1, p_2, q_1, q_2) = (0, 0, 1, 1)$, so the last equation in the reduced factoring system for pq is $p_2q_1 + p_1q_2 = 0$. Had Xu et. al. factored a semiprime of the form $N = (8x + 1)(8x + 7)$, e.g. $N = 391 = 17 \times 23$, then their experiment would also have performed the quantum portion of the factorization of this sexy prime product.

Before we conclude this section, there is an important point to address. When we say that N_{2000} was "factored" by [12], what we really mean is that their NMR experiment performed the quantum portion of the hybrid factoring algorithm for N_{2000}. Xu et. al. also performed the classical portion of the algorithm on 143, while Dattani and Bryans carried out the classical portion for 56153. Before we say that N_{2000} was truly factored by [12], we should perform the classical portion of the factorization—that is, the reduction from the factoring equations to the reduced factoring system—using an algorithm with no prior knowledge of the solution.

There are at least several classical algorithms which perform this reduction rapidly for N_{2000}. One is a new factoring algorithm which runs very

quickly on $N = pq$ when p and q only differ on low order bits. The idea is to guess the highest index k such that $p_k \neq q_k$, write p and q in the form $p = 2^{k+1}x + a$ and $q = 2^{k+1}x + b$, and exhaustively check values of a and b until the quadratic equation $(2^{k+1}x + a)(2^{k+1}x + b) = N$ has an integer solution for x. The full algorithm is as follows:

Data: A composite integer N that is not a perfect square
Result: Integers p and q, with $1 < p, q < N$, such that $N = pq$

```
for k ← 0 to ⌊log_2 N⌋ + 1 do
    for all 2^k ≤ a ≤ 2^{k+1} − 1 and 0 ≤ b ≤ 2^k − 1 do
        Solve the quadratic equation (2^{k+1}x + a)(2^{k+1}x + b) = N
        if the larger root x is an integer then
            p ← 2^{k+1}x + a
            q ← 2^{k+1}x + b
            return p, q
        end
    end
end
```

The algorithmic runtime is $O(2^{2k})$, which is significantly worse than trial division in worst and typical cases. For N_{2000}, however, we have $k = 2$, so the algorithm runs almost instantly on modern hardware. The algorithm runs in polynomial time in the rare cases where $k = O(\log \log N)$.

A more conventional means of reducing the factoring equations for N_{2000} to their reduced factoring system is to use the Fermat factoring method. This factoring algorithm works very well when N has a factor near $\sqrt{N}$. This is the case for N_{2000}, since its prime factors are $\lfloor\sqrt{N_{2000}}\rfloor$ and $\lceil\sqrt{N_{2000}}\rceil + 1$. The Fermat method is worth considering even when we don't know ahead of time that there is a factor near $\sqrt{N}$, since several reasonably fast variants exist—a modification from Lehman [6] runs in time $O(N^{1/3})$, while another from McKee [7] achieves heuristic runtime $O(N^{1/4+\epsilon})$. The vanilla Fermat method is sufficient for N_{2000} and finds the larger prime factor during the second iteration of the main loop.

Both of these methods require negligible time for N_{2000}, after which it is trivial to derive the three equations $p_1 + q_1 = 1$, $p_2 + q_2 = 1$, and $p_2q_1 + p_1q_2 = 1$. Having exhibited the classical reduction to the reduced factoring system, we are now justified in saying that N_{2000} was factored by [12].

5 A Closer Look at the Hybrid Algorithm

In section 4, we remarked that [12] performed the quantum portion of the hybrid factorization algorithm for infinitely many numbers, conditional on the Bateman-Horn conjecture. We begin this section by proving—unconditionally—that a finite quantum computer is sufficient for the factorization of infinitely many semiprimes. We make use of a theorem from the Polymath8b project:

Theorem 3 [9]: There are infinitely many triples of primes $p_n < p_{n+1} < p_{n+2}$ such that $p_{n+2} - p_n \leq 398130$.

This allows us to prove the following corollary:

Corollary 4: There are infinitely many pairs of primes differing only on some of their 19 least significant bits.

Proof. Let $p_n < p_{n+1} < p_{n+2}$ be a triple of primes with $p_{n+2} - p_n \leq 398130$, and let $m \cdot 2^{19}$ be the largest multiple of 2^{19} less than p_n. We can bound p_{n+2} as:

$$\begin{aligned} p_{n+2} &\leq p_n + 398130 \\ &< (m+1) \cdot 2^{19} + 398130 \\ &< (m+2) \cdot 2^{19} \end{aligned}$$

So we have:

$$m \cdot 2^{19} < p_n < p_{n+1} < p_{n+2} < (m+2) \cdot 2^{19}$$

If $p_{n+1} < (m+1) \cdot 2^{19}$, then

$$m \cdot 2^{19} < p_n < p_{n+1} < (m+1) \cdot 2^{19},$$

in which case p_n and p_{n+1} occur between consecutive multiples of 2^{19}, hence they only differ on some of their least significant 19 bits. Otherwise, if $p_{n+1} > (m+1) \cdot 2^{19}$, then

$$(m+1) \cdot 2^{19} < p_{n+1} < p_{n+2} < (m+2) \cdot 2^{19},$$

meaning that p_{n+1} and p_{n+2} differ only in their lowest 19 bits. Either way, we have a pair of primes differing only on some of their 19 least significant bits. □

All but finitely many of these pairs are odd prime pairs, which necessarily agree on the least significant bit. There are, therefore, infinitely many pairs

of primes (p, q) with 18 or fewer bit differences. The corresponding products pq can be factored by the hybrid factoring algorithm with 36 qubits, since their reduced factoring systems have 36 variables (namely the p_i and q_i with $p_i \neq q_i$). We can reduce the qubit requirement further with a trick from [8]: using the equations $p_{i_j} + q_{i_j} = 1$, we can substitute for each q_{i_j} as $q_{i_j} = 1 - p_{i_j}$. The result is a further simplified system with half as many variables, meaning that 18 qubits is enough for an infinite number of factorizations.

Note that the hybrid algorithm can factor a semiprime using 18 qubits when the prime factors differ on any 18 bit positions. The prime pairs mentioned in Corollary 4 are examples of a special case in which all differences occur on low order bits.

We have seen that a particular 4 qubit experiment performed the quantum factorization for infinitely many products conditional on the Bateman-Horn conjecture, and that a finite quantum computer is unconditionally sufficient for infinitely many factorizations. We contend, however, that these facts are probably not indicative of meaningful progress toward factoring RSA moduli. The 4 qubit experiment in [12]—as well as the hypothetical 18 qubit experiment suggested by Corollary 3—only succeed in factoring very specific numbers which would never be chosen as RSA moduli.

First, the hybrid factoring algorithm can factor $N = pq$ with a small number of qubits only when p and q agree on most bits. Heuristically, two randomly chosen primes of similar bit length should agree on about half of their bits. We expect, therefore, that if $N = pq$ is a strong RSA modulus, then p and q differ on about $(\log N)/4$ bits, meaning that about $(\log N)/4$ qubits are needed to run the hybrid factoring algorithm after applying the substitution trick from [8]. By way of comparison, a naïve implementation of Shor's algorithm requires about $4 \log N$ qubits. While the hybrid algorithm does appear to use fewer qubits than Shor's algorithm by a factor of 16, the two algorithms are fundamentally quite different; thus, it is not obvious that this advantage will persist after accounting for error correction and reduced-qubit variants of Shor's algorithm.

Second, and more importantly, the hybrid algorithm does not improve on classical factoring algorithms in terms of overall runtime. The proof of this begins with a lemma.

Lemma 1: The reduced factoring system for any N can be solved in linear time classically.

The algorithm works as follows. From the first equation of the reduced factoring system,

$$p_{i_1} + q_{i_1} = 1,$$

we know that one of p_{i_1} and q_{i_1} is 1 and the other is 0. Arbitrarily let p be the factor with $p_{i_1} = 1$, so that $p_{i_1} = 1$ and $q_{i_1} = 0$. Once this choice is made, all other variable values are determined and can be deduced efficiently. For example, in the equation

$$p_{i_1} q_{i_2} + p_{i_2} q_{i_1} = r_1,$$

substituting $p_{i_1} = 1$ and $q_{i_1} = 0$ yields

$$q_{i_2} = r_1.$$

We therefore have the value of q_{i_2}, and substituting into the equation

$$p_{i_2} + q_{i_2} = 1$$

allows us to determine p_{i_2} too. We can then use the values of p_{i_2} and q_{i_2} in combination with the equations

$$p_{i_2} q_{i_3} + p_{i_3} q_{i_2} = r_2,$$

$$p_{i_3} + q_{i_3} = 1$$

to determine the values of p_{i_3} and q_{i_3}. We solve the system by repeating this process until all p_{i_j} and q_{i_j} are determined.

With this lemma in tow, we are now ready to conclude the following:

Corollary 5: The classical portion of the hybrid factoring algorithm cannot be done efficiently unless factoring is in P. Simplifying to the reduced factoring system is as difficult as factoring.

Proof. Suppose there exists a polynomial time algorithm for simplifying the factoring equations to their reduced factoring system. By combining this algorithm with the linear time algorithm for solving the reduced factoring system, we obtain a polynomial time factoring algorithm. □

In this light, it seems odd to say that the factorizations of 143, 56153, and N_{2000} were "quantum" factorizations—the asymptotically difficult parts were done classically.

If solving the reduced factoring systems is easy in general, then why did the computer algebra systems used by [12] and [4] stop without performing this easy final step? This may be because they lacked the information needed to proceed. The linear time algorithm for solving these systems starts by arbitrarily letting p be the factor with a 1 at bit index i_1. A different solution—with all variable values inverted—can be obtained by simply swapping p and

q. The values of the bits on which p and q agree can be determined because their values are the same regardless of which factor is called p and which is called q. Had [12] and [4] provided just a little extra information, e.g. by arbitrarily letting $p < q$, their computer algebra systems likely would have determined all variable values, leaving nothing to be solved by a quantum computer.

There is one final point that should be addressed. In [12], Xu et. al. state that their classical simplification rules can be applied in time polynomial in $\log N$. It is possible, then, that they had envisioned a slightly different hybrid factoring algorithm, in which polynomial time is spent applying some classical simplifications and a quantum computer is used to solve whatever remains.

We cannot rule out the possibility that such an algorithm could achieve an advantage over Shor's algorithm, but there is reason to be doubtful. Burges points out [3] that the number of variables in the factoring equations, with carry bits included, is $O((\log N)(\log\log N))$. Empirically, and in accordance with some back of the envelope arithmetic, the hidden constant here seems to be close to 1—there really are about $(\log N)(\log\log N)$ variables in the factoring equations.

The quantum portion of the factorization will need as many qubits as there are variables, so achieving a practical advantage over Shor's algorithm requires the elimination of many variables. It isn't clear that a significant number of variables can be eliminated in classical polynomial time—with about $(\log N)(\log\log N)$ unknowns and just $O(\log N)$ equations, Gaussian elimination certainly will not suffice. Quick classical simplification may leave all but a handful of variables, in which case this modified hybrid algorithm would require asymptotically more qubits than Shor's algorithm. For modern RSA moduli with 1024 to 4096 bits, $\log\log N$ will range from about 10 to 12, so that the modified hybrid algorithm needs to eliminate well over half of the variables just to match the number of qubits used by a naïve implementation of Shor's algorithm.

6 Conclusion

It is not incorrect to say that 143, 56153, and N_{2000} were factored using a quantum computer—the experiment in [12] really did perform the quantum portion of a factoring algorithm for these numbers, and the classical portions for 56153 and N_{2000} have now been performed too.

However, the greater significance of these developments remains unclear. Progress in classical factoring is usually measured in terms of strong RSA

moduli, which are, by design, particularly difficult to factor. One could of course choose specific numbers which are much larger and can be factored quickly. A similar principle applies here; while 143 and 56153 were probably not chosen so as to be easy, they were factored using a 4 qubit quantum computer largely because they are easy to factor. Moreover, asymptotically speaking, most of the work in these factorizations was done classically. This point is easy to miss when working with numbers such as 143 and 56153 because they are not large enough for this asymptotic behavior to become apparent.

The experiment performed by Xu et. al. is intrinsically interesting and may lead to larger scale quantum computations. The insight provided by Dattani and Bryans is also valuable and calls attention to the factoring equations as a promising domain for future study.

But as things stand, this approach to quantum factorization does not seem to attain a real advantage over Shor's algorithm. It would be simple to push the record for quantum factorization even larger than N_{2000} by searching for larger semiprimes of the form $(8x+3)(8x+5)$, even though such a result obviously should not be taken as a sign that quantum factorization methods have improved—and certainly not that RSA moduli with more than 2000 decimal digits can be factored on modern hardware.

Acknowledgement

This work was supported by the National Science Foundation (grant CCF-1420750). The author is also grateful to Eric Bach for many helpful discussions over the course of this research.

Appendix

Here we provide the full value and prime factors of N_{2000}, which was factored with a quantum computer in precisely the same sense as 56153: the quantum portion of the factorization was performed by [12] in 2012, and the classical portion was performed later. Neither step used any prior knowledge of the solution.

$$
\begin{aligned}
N_{2000} = {} & 7858653588670966314503258423345019253421014934807067949098667 \\
& 6081252716028643828314627766874719420466679216634720340926555 \\
& 8086907258219649750448860530761410896212386458650652124289173 \\
& 7822249557481051631596091484002527962413903527028324333646485 \\
& 9154829891957186553387660728037474310158841745357100183333912 \\
& 0528999728968749683349830032309892148365208855644606696124006 \\
& 6877033866187973554440789967453580275830548870876658344758727 \\
& 2463612425340014879134178545496056509506531427731301319092366 \\
& 4534726291669733245246270021518228734755698203665669434940749 \\
& 9740708425859067737556168219236757864611405371193815812521074 \\
& 3739522502884759347707347507667096076977398291004932719106888 \\
& 8528971901247531821087658865767378437679893142541692881366183 \\
& 4539730866534384835011228473289888230233529260801532870050631 \\
& 1074268812191695660132740043833616632543358572329295327729695 \\
& 4815343536520617896812279708813311084071653210347117069340640 \\
& 5333933189063704682395576691603199493023332686608830565633876 \\
& 3948683583606536026526599552843261208707443625669687256265442 \\
& 6990051067722147626741300280615820611068155250682069171281973 \\
& 9278940617743110919272353341106444458478241067985666660506442 \\
& 0378075179219255381152986600082238426830997144108510537320411 \\
& 6136819067235274797622338835768781658591717163050357272428102 \\
& 0942123748259936259056338942542354115744657980820211111223402 \\
& 7997803991538925603362276708605427635193993471586067921422426 \\
& 7366864403868131185226611400298923980872182716909708512693817 \\
& 3235283279062282826429511354621302945791725882066178315492294 \\
& 3414821255536930897349966811682462484727532026979388536866971 \\
& 4861019482336534487153307092842595505420537839279423870579041 \\
& 7126727328604197863118632593917603958122022060729791897675562 \\
& 7259445271268019316955677347324825679044799394644406793830441 \\
& 4948045845321355131879186865085278709067458905295184313709095 \\
& 4583985002903598515789660656334066212798162636895605890730329 \\
& 5222242820337598216537661365980470100623210884196768759082173 \\
& 8913478192556888771867608110501339077213007715 99
\end{aligned}
$$

The factors p and q of N_{2000} are:

$$
\begin{aligned}
p &= 2803329018982781891486381028341973671001958263886711448992797 \\
&\quad 2023077573466184019856174159159447050303144033043302399488862 \\
&\quad 5755807334891984723417248391195798511529358464318438457215803 \\
&\quad 8581215423268615906902043435822702561637034410888078889337930 \\
&\quad 5823085777969549542225363553903141015819572257825120283776375 \\
&\quad 4852121150315412579003033689347963903028321025971033728216669 \\
&\quad 7997616074509285483406631291333496910119005254554208194223557 \\
&\quad 5076708997238155866594923487707327059188532911156387038095442 \\
&\quad 4820694249395054341241068814822470274905779216555296657837043 \\
&\quad 8972058955997522078331247605130502671096739999057510906833686 \\
&\quad 6543292914524819266520289037118633531973122493807450373456997 \\
&\quad 1210216451105190777794366325414442783870837756702270263733803 \\
&\quad 8698036648461564366770249777919041182785647866286542108916166 \\
&\quad 0771790987628494174474663220751511171111687343971109459304399 \\
&\quad 5246232258741074751400185237794139144445504938364602686085775 \\
&\quad 9976314593532814332421121316116268425045967584130933657787488 \\
&\quad 7881265136771626766143459 \\
q &= p + 2
\end{aligned}
$$

References

[1] J. K. Andersen, T. Alm, and M. Fleuren. Gigantic Sexy and Cousin Primes. https://groups.yahoo.com/neo/groups/primeform/conversations/topics/6637, 2005.

[2] P. T. Bateman and R. A. Horn. A Heuristic Asymptotic Formula Concerning the Distribution of Prime Numbers. *Mathematics of Computation*, 16, 1962.

[3] C. J. C. Burges. Factoring as Optimization. *Microsoft Research*, MSR-TR-200, 2002.

[4] N. S. Dattani and N. Bryans. Quantum Factorization of 56153 With Only 4 Qubits, 2014.

[5] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda. A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem. *Science*, 292, 2001.

[6] R. S. Lehman. Factoring Large Integers. *Mathematics of Computation*, 28, 1974.

[7] J. McKee. Speeding Fermat's Factoring Method. *Mathematics of Computation*, 68, 1999.

[8] S. Pal, S. Moitra, V. S. Anjusha, A. Kumar, and T. S. Mahesh. Hybrid Scheme for Factorization: Factoring 551 Using a 3-Qubit NMR Quantum Adiabatic Processor, 2016.

[9] D. H. J. Polymath. Variants of the Selberg Sieve, and Bounded Intervals Containing Many Primes. *Mathematical Sciences*, 1: 12, 2014.

[10] G. Schaller and R. Schützhold. The Role of Symmetries in Adiabatic Quantum Algorithms. *Quantum Information and Computation*, 10, 2010.

[11] J. A. Smolin, G. Smith, and A. Vargo. Oversimplifying Quantum Factoring. *Nature*, 414, 2001.

[12] N. Xu, J. Zhu, D. Lu, X. Zhou, X. Peng, and J. Du. Quantum Factorization of 143 on a Dipolar-Coupling Nuclear Magnetic Resonance System. *Physical Review Letters*, 108, 2012.