Quantum Semiprime Factorization: Leveraging Grover's Algorithm for Efficient Prime Decomposition

Jacob Collins[1], Jaime Raigoza[1], Sam Siewert[1]

[1] California State University, Chico, United States of America (note- full address at end of paper)

Abstract

In quantum computing, SP (semiprime) factoring is typically performed with Shor's algorithm, but it is possible to achieve the same results more efficientily via Grover's algorithm[1, 2].

Key terms: Semiprime, superposition, entanglement, RSA encryption, (add more).

The abstract with its heading should not be more than 100 mm long, which is equivalent to 25 lines of text. Leave 2 line spaces at the bottom of the abstract before continuing with the next heading.

1. Introduction

The practicality of applied quantum computing is a topic of much debate, and there are not many examples of quantum computers being able to out-perform their classical counterpart, yet.

Quantum algorithms have been in development at a theoretical level for decades, but the application of these methods is still in early stages. It is quite common to find code which implements a quantum algorithm like Grover's or Shor's, but very often these programs are hard-coded, and only work for a small handful of specific values[3].

Include a figure of our circuit diagram and briefly describe what it does, and how it does it.

1.1. Motivation

Why are we replicating the research, why is this important.

Quantum computers are a technology very much still under development. As such, one of the major limitations for applied quantum computing is the number of qubits required to perform a given task. If it can be verified that SP factoring with Grover's algorithm uses significantly less qubits than Shor's, then the application of these techniques will be feasible far sooner.

1.2. Semiprime Factoring

What is SP factoring and why is it important. RSA, etc.

2. Goals

The goal of this research endeavor is to estimate the problem scaling point at which some quantum algorithm may be able to out-perform our best classical solution to the same problem.

Grover's algorithm has an incredibly wide range of potential use-cases, and an intention for this paper is to provide a generalized reference point for how Grover's algorithm may be used to invert any function.

3. Literature Review

3.1. Grover's Algorithm

Grover's Algorithm is a quantum computing method that can be used to search a database in $O(\sqrt{n})$ time rather than the naive classical time complexity of $O(n)$[2].

Given a function $f(x) = y$, where x is unknown (index, prime factors, sum components, etc.), and y is known (array value, semiprime/product, sum, etc.), Grover's algorithm effectively takes the role of $f'(y)$, allowing for a potential speedup in finding whatever input(s) to $f(x)$ will return y.

This speedup comes from the fact that Grover's algorithm requires $O(\sqrt{y})$ iterations, each of which have a time complexity proportional to that of $f(x)$.

3.2. Shor's Algorithm

Description of Shor's algorithm.

3.3. Quantum Factoring Algorithm with Grover Search

S. Whitlock, et al. present a method of SP factoring with Grover's algorithm. The implementation in their paper is

highly optimized, and modifies the target value in their circuit from some semiprime N, to M, a reduced number uniquely tied to N, which has prime factors p and q.

While this approach is admirable, it introduces a level of complexity that may hinder a learner's understanding of the mechanics at play, so the implementation shown in section 4 forgoes this abstraction from N to M, and instead implements an oracle which searches directly for prime factors a and b of a semiprime N.

4. Methodology

Quantum algorithms take advantage of superposition and entanglement, which enables many methods of computation which are otherwise impossible with a classical computer. For example, by placing a set of n qubits (otherwise known as a qubit **register**) in superposition, that register simultaneously represents every value which can be represented in n bits, until measured. Once measured, a register in uniform superposition will collapse into one of these states with equal likelihood.

In a given register R_n with n qubits, values are represented in binary, such, for example, if some number $N = 3$ were to be represented classically with $n = 4$ bits, it might be seen in binary as 0011, and on a qubit register that number may be represented similarly as $|0011\rangle$.

4.1. Grover's Algorithm for General Function Inversion

Mention unitary matrices.

Describe how Grover's works and show figures of a generalized circuit for Grover's to invert functions

Reference figure 1.

Operations can be performed on a register in superposition, and these operations will affect each potential state individually. This property is utilized by Grover's Oracle to selectively target specific states which match the target value, and the diffuser is used to increase the probability of measuring the inputs which resulted in this marked target state.

Include one figure showing the oracle, and one figure showing the diffuser.

4.2. Semiclassical Arithmetic

Describe our original approach- bitwise addition, registers, operations, etc.

Mention why an alternative approach was needed.

Include a drawing of the circuit.

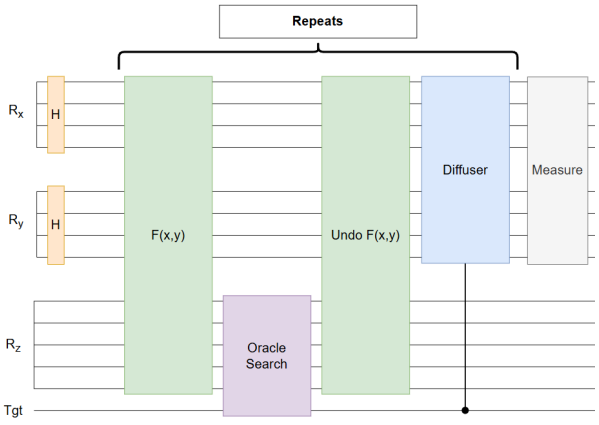

Figure 1. General circuit diagram for Grover's Algorithm, used to find the inputs x, y for which $f(x, y) = z$, for some known z.

4.3. Arithmetic in the Quantum Fourier Domain

Describe the updated methods of quantum arithmetic, and how it solved the previous problems.

Figures to visualize how the weighted phase shifts work to produce expected results matching addition, scaled addition, and register multiplication.

5. Results and Discussion

Include some charts showing our runtime and qubit requirements vs N.

5.1. Accuracy and Limitations

How precise and accurate were our measurements. Mention edge-cases like square semiprimes, etc.

How many qubits were we able to simulate, i.e., how large of semiprimes can we factor with our current systems.

5.2. Comparison to Shor's Algorithm

List strengths and weaknesses of Shor's, i.e., more documentation, more examples, larger code base to reference, but ours is faster and uses less qubits.

6. Conclusion

What have we learned so far, what does it mean, at what problem scale might we see quantum advantage based on our results, etc.

7. Future Work

Implementation of the optimization with M, S, p, and q, rather than just a and b from N.

Acknowledgments

Acknowledge the source paper.

References

[1] Whitlock S, Kieu TD. Quantum factoring algorithm using grover search, 2023. URL https://arxiv.org/abs/2312.10054.

[2] Grover LK. A fast quantum mechanical algorithm for database search, 1996. URL https://arxiv.org/abs/quant-ph/9605043.

[3] Skosana U, Tame M. Demonstration of shor's factoring algorithm for n = 21 on ibm quantum processors. Scientific Reports August 2021;11(1). ISSN 2045-2322. URL http://dx.doi.org/10.1038/s41598-021-95973-w.

Address for correspondence:

Jacob Collins
1565 Filbert Ave
jbcollins@csuchico.edu