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].

Grover's Algorithm is used to search for one (or many) given state(s) among some set of possible outputs. In this case, given some semiprime, Grover's algorithm may be used to find the prime factors.

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

What are we doing, broadly?

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.

1.2. Semiprime Factoring

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

2. Goals

Demonstration of quantum advantage, or at least determination of the problem scale at which quantum advantage will pull ahead.

3. Literature Review

Discuss existing methods of SP factoring- both quantum and classical.

Mention reference paper, QFT arithmetic, CUDA-Q docs, shor's, grover's.

4. Methodology

Quantum algorithms take advantage of superposition and entanglement, allowing this form of computation to perform operations on many values simultaneously, rather than checking each value individually as is necessary in classical algorithms.

4.1. Using Grover's Algorithm to Invert Functions

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

4.2. Semiclassical Arithmetic

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

Mention why an alternative approach was needed.

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.

Address for correspondence:

Jacob Collins
1565 Filbert Ave
jbcollins@csuchico.edu