Quantum Computation Project

The essence of problems where quantum computers fail to outperform classical computers

Nithin Puthalath Manoj   Adwaith Ram Kishore
Pavithra Rajan   Joel Mathew Cherian

November 24, 2022

Contents

Introduction

- Quantum Computers cannot significantly outperform classical computers while trying to solve **NP-complete** problems.
- Narrowed the problem statement by taking one such problem: **3-colouring graph** problem
- Real-life applications of graph colouring
- Analysed the problem via two approaches to see if there was a notable improvement in time complexity

Classical Approach: Backtracking

- For various NP-complete problems, a brute force approach is observed to be the fastest classical algorithm. For some, a refined brute-force techniques like **backtracking** can be adopted.
- Upper-bound time remains the same but the average time taken reduces
- Time complexity: $O(3^n)$, where n is the number of vertices

Method:

- Assign colours to each vertex one by one
- Before each assignment, verify if any of the adjacent vertices have the same colour
- If there are no violations proceed else backtrack a step and reassign colours

Grover's Search

Speeds up an unstructured search problem quadratically by using Grover's Amplitude Amplification Trick. Widely used as a sub-routine in many other algorithms to improve computation time.

- Given a list of N items, it achieves the search problem in roughly $\sqrt{N}$ steps, which is significantly faster than the classical approach, which is N steps.
- It takes advantage of the superposition principle of qubits
- It applies the Oracle function and the Diffusion Operator multiple times to the superposed state to amplify the amplitude of the qubit state we are searching for.

Case Study and Comparative Analysis

The 3 colouring problem can be solved via a quantum method by converting it to an equivalent problem that yield the same result.

- Compute the **chromatic number** and utilise Grover's search
- **NK method**: generation of superpositions of all possibilities and filtering
- Reducing the problem to a **SAT problem**

Chromatic number computation

Chromatic number is the minimum number of colours required to colour the graph in such a way that it doesn't violate the adjacent colour node rule.

- A quantum dynamic programming approach using **Byskov's algorithm** that generates all the Maximal Independent Sets. As Byskov's algorithm is a branching method, we can use Grover's search on it.
- This algorithm computes the chromatic number with a time complexity of $O(1.9140^n)$.
- This is still in exponential time!

NK method

This is a generalised approach to colour an n-vertex graph with k colours.

- Assign K qubits to each node and generate superpositions of all possible colourings
- Filter these possibilities by applying the outer product, horizontal concatenation and element wise product
- Eliminate possibilities that violate the rule
- A row entry should not contain 2 as it implies that two nodes have the same colour
- The time complexity will be $O(N^2)$ where N is the number of qubits.

If we expressed the time complexity in terms of k and n then it would be $O(k^{2n})$, which is still exponential time!

Reduction to SAT problem

- A Reduction-Based Problem Mapping for Quantum Computing relies on the property that NP-Complete problems can be reduced to each other in polynomial time.
- This can be implemented by choosing a particular problem S and transforming the general input in polynomial time and the computed output is converted back to the desired output for the original input.
- The unrestricted Boolean-SAT is chosen as the general problem which can be solved with the help of Grover's Quantum Search Algorithm.

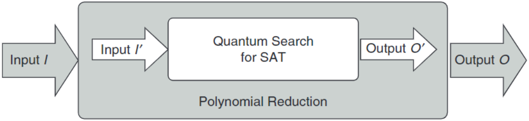


Figure: End-to-End Reduction Framework

Our Approach

- We have built a quantum-based 3-colouring problem solver using Qiskit.
- The solver is capable of deciding the 3-colourability of a graph while generating a possible solution.
- The solver is based on the **Reduction to SAT Problem** approach. This approach was the most alluring since it involved solving the SAT problem.
- There are a total of three major modules: *Graph to SAT converter*, *Quantum Circuit Generator*, and *SAT solution to Coloured graph converter*.
- The Quantum Circuit Generator is the most important component of the solver. It has two sub-modules: the Phase Oracle Generator and the Diffusion Operator Generator.

Quantum Circuit Generator

- **Step 1:** Qubits are used to represent variables from the SAT problem. Initially, in the circuit Hadamard gates are used to put all the qubits in uniform superposition.
- **Step 2 (Phase Oracle):** This oracle flips the phase of states based on the boolean function represented by the SAT problem. The oracle is made with a combination of X gates and Toffoli gates.
- **Step 3 (Diffusion Operator):** The Diffusion operator is build to scale with the number of the variables in the SAT problem. It is composed of X gates, Hadamard gates and Toffoli gates.
- **Step 4:** The Phase Oracle + Diffusion Operator unit is repeated multiple times (upto $2^{N/2}$ where N is number of variables in the SAT problem).
- **Step 5:** Run the quantum circuit on a quantum computer or a simulator (Eg: AER Matrix Product State Simulator)

Quantum Circuit Example

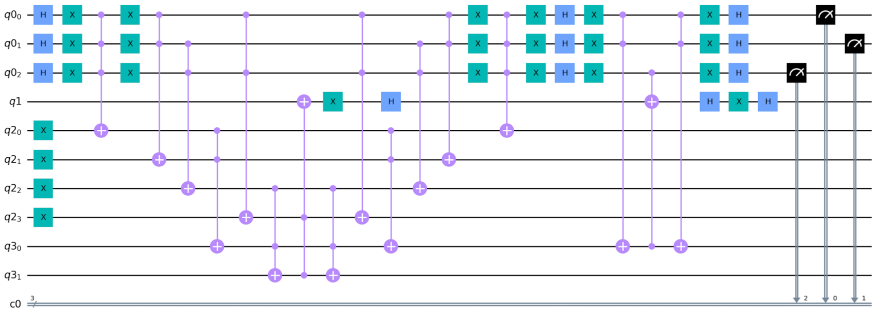
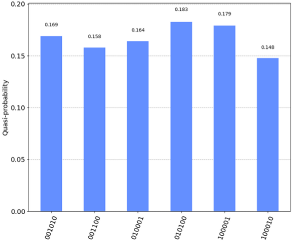


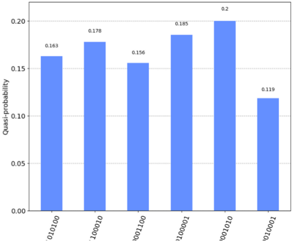
Figure: Quantum circuit generated by our solver for a trivial single node graph

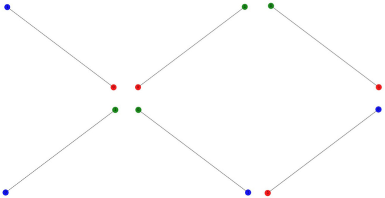
Results with a 2-node and 3-node Graph

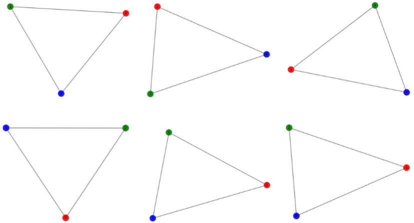
2-nodes



3-nodes



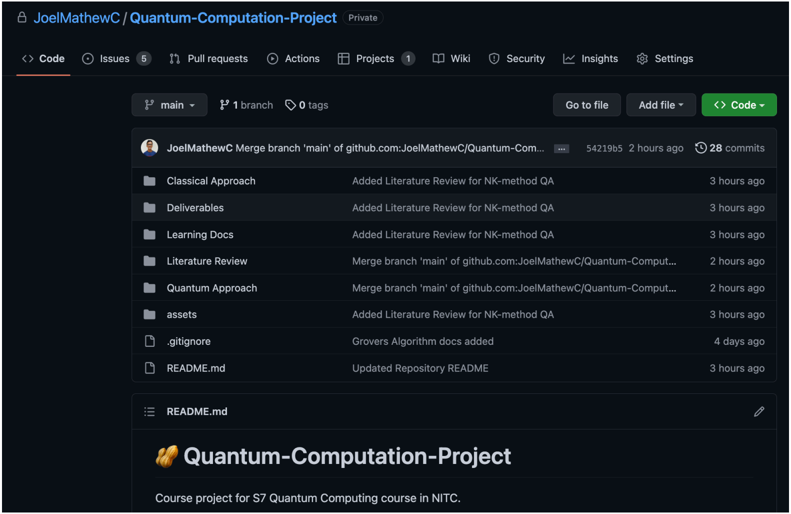




Our Contributions

- Reviewed the most recent literature revolving the topic
- Documented the entire learning process as we realised one of the main challenges in this field is the lack of systematic documentation and comprehensive code structuring
- Implemented all of the components from scratch and without the usage of pre-defined libraries to avoid black-box learning.
- This implementation is built to be scalable and it can be improved upon to generate more efficient solvers.
- Comprehensive analysis to prove the given problem statement

Repository


JoelMathewC / Quantum-Computation-Project Private
Code
Issues 5
Pull requests
Actions
Projects 1
Wiki
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
JoelMathewC Merge branch 'main' of github.com:JoelMathewC/Quantum-Com...
54219b5 2 hours ago
28 commits
Classical Approach
Added Literature Review for NK-method QA
3 hours ago
Deliverables
Added Literature Review for NK-method QA
3 hours ago
Learning Docs
Added Literature Review for NK-method QA
3 hours ago
Literature Review
Merge branch 'main' of github.com:JoelMathewC/Quantum-Comput...
2 hours ago
Quantum Approach
Merge branch 'main' of github.com:JoelMathewC/Quantum-Comput...
2 hours ago
assets
Added Literature Review for NK-method QA
3 hours ago
.gitignore
Grovers Algorithm docs added
4 days ago
README.md
Updated Repository README
3 hours ago
README.md
Quantum-Computation-Project
Course project for S7 Quantum Computing course in NITC.

Conclusion

- By taking an instance of an NP-Complete problem, we have seen that quantum computers do not significantly outperform classical computers.
- Any computational problem that can be solved by a classical computer can also be solved by a quantum computer. Conversely, any problem that can be solved by a quantum computer can also be solved by a classical computer with the principle that sufficient time is given.
- For all applications, a classic computer will not only continue to **co-exist** with quantum computers but also dominate in most of the arenas.
- Though Quantum Computers will not be able to solve all our problems as we expect it to, it is **not** a reason for research to **not** be done in this domain.

Bibliography

- Hu, S., Liu, P., Chen, C. R., Pistoia, M., & Gambetta, J. (2019). Reduction-based problem mapping for quantum computing. Computer, 52(6), 47-57.
- Shimizu, K., & Mori, R. (2022). Exponential-time quantum algorithms for graph coloring problems. Algorithmica, 1-19.
- Clerc, M. (2020). A general quantum method to solve the graph K-colouring problem.
- Aaronson, S. (2008). The limits of quantum. Scientific American, 298(3), 62-69.
- D' Hondt, E. (2009). Quantum approaches to graph colouring. Theoretical computer science, 410(4-5), 302-309.
- https://qiskit.org/textbook/

Thank you