University of Western Australia

Master of Physics

Quantum Solutions to Graph Similarity Problems

Dylan O'Donoghue

Department of Physics

Supervised by

Prof. Jingbo Wang
Department of Physics

1 October 2025

Acknowledgments

I would like to thank my supervisor, Jingbo Wang. Your support and guidance throughout the course of my studies has made this journey immensely rewarding. Thank you for pulling together an incredible team of like-minded students and mentors, and for your excellent management of the QUISA research group and the quantum optimisation subgroup. In taking on the role of connecting people together in meaningful pursuits of knowledge, you have created an community of collaborators which extends well beyond the usual barrier of academics and researchers into the real world. It is only through collaborations of this kind that science can impact society.

I also extend my thanks to Tavis, Edric, Pascal, Aidan, Andrew, and Leo for their support on the project. Tavis for sharing his valuable research, which made this whole project possible. Edric for the many insights in data analysis and QuOp. Pascal for the top-down view of the project as a quantum-HPC hybrid expert. Aiden, Andrew, and Leo, for the feedback, collaboration which guided the analysis and generation of these results.

Thank you to Benjamin Rossdeutscher. Our academic journeys have been in parallel for many years, and your personal and professional support have been a key part of not just this thesis but the path here.

Finally, I thank my family and friends for their continuous support throughout my degree. Without you, none of this would have been possible.

Declaration

This is to certify that:

1. This thesis comprises of my original work.
2. Due acknowledgment has been made in the text to all other materials used.
3. This thesis consists of less than 60 pages inclusive of tables and figures but exclusive of references and appendices.

I authorise the Head of the Department of Physics to make a copy of this thesis to any person judged to have an acceptable reason for access to the information, i.e, for research, study or instruction.

Student Name: Dylan O'Donoghue

ID: 23158061

Signature

Date

Supervisor: Professor Jingbo Wang

Signature

Date

Summary of Student Achievement

Chuck something in here.

Or maybe not?

Contents

List of Figures

List of Tables

Abstract

Brief summary.....

Chapter 1

Introduction

Quantum computing is a rapidly evolving field that leverages the principles of quantum mechanics to perform computational tasks. By utilising the unique properties of quantum bits (qubits), such as superposition and entanglement, quantum computers can solve certain problems more efficiently than classical computers. This has led to significant interest in exploring quantum algorithms and their applications across various domains.

Network graph models have become crucial components in systems that are used in everyday life, such as search engines and social networks. Graph similarity is often important in these contexts. For example, social networks are compared to identify certain interaction patterns, traffic networks are compared to help detect abnormal changes in traffic patterns, and web networks are compared for anomaly detection. In all these cases, the similarity between two graphs with overlapping sets of nodes is assessed, and the detection of changes in the patterns of connectivity is important. For graphs that match approximately, it is useful to obtain a quantitative measure of similarity.

Graph similarity measures are quantitative calculations based on comparisons made between the structure of network graphs. Different measures of graph similarity will produce a variety of results because of differences in how the structures of the graphs are analysed. There are many measures of graph similarity, in-

cluding Maximum Common Subgraph, Graph Edit Distance, Frobenius Distance, and Graph Eigendecomposition. All of these measures are successful in indicating the degree to which two graphs are similar and can also detect small differences between them.

However, many of these measures require a matching between the vertices of the two graphs, which can be computationally expensive to determine. The problem of finding an optimal vertex matching is known to be NP-hard, meaning that there is no known polynomial-time algorithm to solve it for all instances. As a result, various heuristic and approximation techniques have been developed to tackle the graph similarity problem.

A recently introduced algorithm, the non-variational QWOA was designed to solve hard combinatorial optimisation problems, generalising to non-binary and constrained problems, while simultaneously solving the challenges related to the variational approach [1]. It amplifies the probability of measuring high-quality solutions to a problem through a process of phase-separation and mixing via a continuous-time quantum walk (CTQW) on a mixing graph. In the original work, it was shown through classical simulation that non-variational QWOA found globally optimal solutions on problems such as weighted maxcut, k-means clustering, quadratic assignment, and the capacitated facility location problem within a small number of iterations.

We seek to determine whether the non-variational QWOA will perform equally well on the problem of calculating graph similarity as it performed on its first problems. We will do this through classical simulation of the non-variational QWOA on small instances of the graph similarity problem. We will compare its performance to that of Grover's search algorithm [2], which is a well-known quantum algorithm for unstructured search problems. Grover's algorithm provides a quadratic speedup over classical search algorithms, making it a useful benchmark for evaluating the performance of other quantum algorithms.
In this report, we will first provide a brief overview of the non-variational QWOA and Grover's search algorithm. We will then describe our methodology for simulating these algorithms on small instances of the graph similarity problem. Finally,

we will present our results and discuss their implications for the use of quantum algorithms in solving combinatorial optimisation problems.

Chapter 2

Literature Review

2.1 Quantum Computing

Quantum computing is based on the principles of quantum mechanics, which describe the behaviour of particles at the atomic and subatomic levels.

This section provides a brief overview of the fundamental concepts of quantum computing that are relevant to understanding the non-variational QWOA and its application to the graph similarity problem.

2.1.1 Qubits and Quantum Superposition

The fundamental unit of information in classical computing is the bit, which can exist in one of two states, typically written as 0 or 1.

In quantum computing, the basic unit of information is the quantum bit or qubit. Unlike classical bits, which can only be in one state at a time, qubits can exist in a superposition of states.

Qubits are mathematically described by their state vectors in a 2^n dimensional complex vector space, where n is the number of qubits. A qubit's state can be represented as the weighted sum of its basis states. A common choice of basis states is called the standard basis, consisting of the states $|0\rangle$ and $|1\rangle$, which are

the positive and negative eigenvalues of the Pauli Z operator. A general state of a single qubit can be expressed as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

where α and β are complex numbers that satisfy the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. The Born rule states that the coefficients $|\alpha|^2$ and $|\beta|^2$ represent the probabilities of measuring the qubit in the states $|0\rangle$ and $|1\rangle$, respectively.

When multiple qubits are combined, their joint state is represented by the tensor product of their individual states. For example, the state of a two-qubit system can be expressed as:

$$\begin{aligned} |\psi_{12}\rangle = |\psi_1\rangle \otimes |\psi_2\rangle &= (\alpha_1\,|0\rangle + \beta_1\,|1\rangle) \otimes (\alpha_2\,|0\rangle + \beta_2\,|1\rangle) \\ &= \alpha_1\alpha_2\,|00\rangle + \alpha_1\beta_2\,|01\rangle + \beta_1\alpha_2\,|10\rangle + \beta_1\beta_2\,|11\rangle \end{aligned}$$

where $|00\rangle, |01\rangle, |10\rangle$, and $|11\rangle$ are the basis states of the two-qubit system.

This allows a quantum computer with n qubits to represent 2^n states simultaneously, which is a key feature that enables quantum parallelism, allowing quantum computers to perform certain computations more efficiently than classical computers.

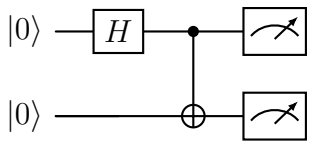
Despite this parallelism, the computational power is limited by the fact that measuring a quantum state collapses it to one of its basis states, yielding only a single outcome. Therefore, quantum algorithms must be carefully designed to manipulate the amplitudes of the quantum states such that the desired outcomes have higher probabilities of being measured.

2.1.2 Quantum Gates, Entanglement

Quantum gates are the building blocks of quantum circuits, analogous to classical logic gates. They manipulate the states of qubits through unitary transformations,

which are reversible operations that preserve the total probability of the quantum system. Common quantum gates include the Hadamard gate, which creates superposition, and the CNOT gate, which entangles two qubits.

We can visually represent a series of quantum gates acting on qubits using a quantum circuit diagram. For example, the following diagram shows a simple quantum circuit with two qubits and two gates:



This diagram represents a circuit where the first qubit is put into superposition by the Hadamard gate (H), and then a CNOT gate acts on the second qubit depending on whether the first qubit is in a $|1\rangle$ state or the $|0\rangle$ state. Finally, both qubits are measured.

The final quantum state before measurement is given by:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

This state is an example of entanglement, a unique quantum phenomenon where the states of two or more qubits become correlated such that the state of one qubit cannot be described independently of the state of the other qubit(s). For instance, in the entangled state above, measuring the first qubit allows one to infer the state of the second qubit.

These correlations are fundamental to quantum algorithms. Without entanglement, quantum computers could be efficiently simulated by classical computers, negating any advantage.

2.1.3 Quantum Algorithms and Challenges

Quantum algorithms seek to leverage properties of qubits and quantum gates to solve specific problems more efficiently than classical algorithms. Notable examples

include Shor's algorithm for integer factorization, which runs exponentially faster than the best-known classical algorithms, and Grover's algorithm for unstructured search, which provides a quadratic speedup over classical search algorithms.

Large, universal, fault-tolerant quantum computers that can run these algorithms at useful scales will require millions of qubits. While researchers have made significant progress in error correction, coherence times, qubit connectivity, and gate fidelities, fault tolerant quantum computing may take decades to achieve. In the meantime, Noisy Intermediate-Scale Quantum (NISQ) computers already exist, and can run small-scale quantum algorithms that can provide insights into quantum behavior and inform the development of larger systems.

These NISQ devices are limited by the number of qubits, gate fidelities, and coherence times. As a result, quantum algorithms designed for NISQ devices must be efficient in terms of qubit usage and circuit depth to minimize the impact of noise and errors.

Even if NISQ devices cannot solve problems faster than the best classical computers using the best classical algorithms (particularly with the emergence of exascale supercomputing), they could still provide an economic advantage by using less energy or by being lower cost to build and operate.

NISQ algorithms are designed to work within these low-resource constraints. Examples include the Variational Quantum Eigensolver (VQE) for quantum chemistry and the Quantum Approximate Optimization Algorithm (QAOA) for combinatorial optimisation problems. These algorithms typically involve a hybrid approach, where a quantum computer is used to prepare and measure quantum states, while a classical computer optimises parameters based on the measurement results.

2.2 Combinatorial Optimisation Problems

Combinatorial optimisation problems consist of a discrete set of feasible solutions, which can be represented as a set S, with finite order $N = |S|$. Each of the N solutions $\mathbf{x} \in S$ can be represented as a vector of n variables $\mathbf{x} = (x_1, x_2, ..., x_n)$

and has an associated objective value (or cost), given by a objective function (or cost function) $f : S \to \mathbb{R}$, which gives a measure of the quality of any solution. The goal of the optimisation problem is to find the *optimal solution* $\mathbf{x}^*$ that maximises (or minimises) the objective (or cost) function:

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in S} f(\mathbf{x}) \text{ or } \mathbf{x}^* = \arg\max_{\mathbf{x} \in S} f(\mathbf{x})$$

The structure of the objective function is determined by the problem definition. The problem definition consists of a series of criteria which are defined on a subset of the variables in $\mathbf{x}$, and the objective function conveys the extent to which a solution satisfies the criteria.

Combinatorial optimisation problems are often NP-hard, meaning that no known polynomial-time algorithm can exactly solve all instances of the problem. Examples of NP-hard combinatorial optimisation problems include the travelling salesman problem and the knapsack problem.

The main challenge in solving these problems is that they often require a near-exhaustive search of the solution space, which usually grows exponentially or super-exponentially with problem size. An example of this super-exponential growth will be present in the graph similarity problem in section 2.3.

Research in sub-exponential combinatorial optimisation algorithms includes the topics of:

- Exact algorithms that find optimal solutions for special problem instances with certain properties.
- Approximate algorithms that find solutions which are near the optimal solutions for general problem instances.

Citations Here

This thesis will focus on approximate algorithms.

To quantify how close a solution $\mathbf{x}_i$ is to the optimal solution, the approximation ratio A is used:

$$A_i = \frac{f(\mathbf{x}_i)}{f(\mathbf{x}^*)}$$

This value compares the objective value of a solution against the maximum possible objective value. A successful approximate combinatorial optimisation algorithm should find solutions which give approximation ratios close to 1.

2.3 Graph Similarity Problem

The graph similarity problem involves determining how similar two graphs are to each other. This can be useful in various applications, such as social network analysis, bioinformatics, and pattern recognition.

There is a need to distinguish between the graph similarity problem and the graph isomorphism problem.

A pair of graphs G_1 and G_2 are considered isomorphic if and only if there exists a bijection between their vertex sets such that any two vertices in G_1 are adjacent if and only if their corresponding vertices in G_2 are also adjacent. In this definition, the graphs are understood to be undirected, non-labelled, non-weighted graphs. However, isomorphism may apply to all other notions of graph by adding requirements to preserve any additional structures.

The graph isomorphism problem is then the task of correctly creating the bijection (or equivalently, proving that one exists). Algorithms that solve this problem are trivially verifiable, as checking the correctness of the solution only involves checking that the vertex bijection correctly preserves the edges.

In contrast, the graph similarity problem is the problem of calculating how close a pair of graphs are to being isomorphic. There are multiple methods of quantifying graph similarity. Some examples include:

- *Graph Edit Distance (GED):* Counts how many 'edits' are required to transform one graph into another, where 'edits' include inserting/deleting a vertex or edge, or substiting one vertex or edge with another. The lower the distance, the more similar the two graphs are.
- *Maximum Common Subgraph (MCS):* Finds the largest induced subgraph that is common to the two given graphs. The larger the subgraph, the more

similar the two graphs are.

- *Frobenius Distance:* Calculates the distance between two graphs using the difference between their adjacency matrices. The lower the distance, the more similar the two graphs are.
- *Edge Overlap*: Finds the proportion of matching entries in the adjacency matrices of the graphs. The greater the proportion, the more similar the two graphs are.

The choice of method often depends on the specific application and the properties of the graphs being compared. Considerations such as cost, time, and interpretability inform the choice of which method to use.

The case of non-labelled graphs is particularly hard to solve, as most algorithms assume a known bijection between the vertices of the given graphs. This assumption does not hold in general for real-world applications for graph similarity. The methods listed above are variant with respect to the correspondence between the vertices. While some correspondence-invariant methods exist, they are limited in interpretability.

Come back to this paragraph.

In this thesis, the assumption of a known vertex correspondence will be discarded. This greatly increases the difficulty of the problem, as we must find the node correspondence between the graphs that maximises their similarity (or, equivalently, minimises their distance). This turns the graph similarity problem into a combinatorial optimisation problem.

2.3.1 Problem Definition

The formal definition for the unlabelled graph similarity problem is as follows:

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two finite, simple graphs with $|V_1| = |V_2|$, and considered *unlabelled* in the sense that vertex identifiers carry no intrinsic meaning

Let $\Pi(V_1, V_2)$ denote the set of all bijections $\pi : V_1 \to V_2$.

Given a base similarity function $f : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ that compares labelled graphs

from the class $\mathcal{G}$ of all finite simple graphs, the *unlabelled similarity* between G_1 and G_2 is defined as

$$\mathrm{Sim}(G_1, G_2) = \max_{\pi \in \Pi(V_1, V_2)} f(G_1, \pi(G_2))$$

where $\pi(G_2)$ denotes the relabelling of G_2 induced by π.

The unlabelled graph similarity problem is the problem of computing $\mathrm{Sim}(G_1, G_2)$.

This problem is known to be NP-hard, meaning that there is no known polynomial-time algorithm to solve it for all instances. As a result, various heuristic and approximation techniques have been developed to tackle the graph similarity problem.

2.4 Variational Quantum Algorithms and the QAOA

2.4.1 VQAs

Variational Quantum Algorithms (VQAs) are a class of hybrid quantum-classical algorithms that leverage the strengths of both quantum and classical computing to solve optimisation problems. They are particularly well-suited for NISQ devices due to their relatively low resource requirements and robustness to noise.

VQAs implement a parametrised quantum circuit (PQC) whose unitary evolution is governed by a set of d classically tunable parameters $\theta \in \mathbb{R}^d$. The quantum device prepares the state

$$|\psi(\theta)\rangle = U(\theta)\,|0\rangle^{\otimes n}$$

where $U(\theta)$ is a depth-restricted circuit composed of parameterised single- and multi-qubit gates, and n is the number of qubits in the input to the quantum computer. Commonly, $d = \mathcal{O}(\mathrm{poly}(n))$, since we seek efficient circuits that are polynomial in resources. citation here.

The objective function of the algorithm is typically expressed as the expectation

value of a Hermitian operator H with repect to the variational state:

$$C(\theta) = \langle\psi(\theta)| H |\psi(\theta)\rangle$$

This cost function is estimated on the quantum device via repeated projective measurements in appropriate bases. A classical optimiser updates θ with the aim of minimising (or maximising) $C(\theta)$, thus driving the quantum circuit towards a state that approximates the optimal configuration with repect to the target problem. The algorithm halts after the output from the quantum computer is 'good enough' at approximating the optimal configuration.

To summarise, VQAs consist of a two-step loop: a quantum computer evolves the initial state to a final state via a PQC, then a classical computer measures the expectation value of the final state and uses this data to update the PQC. At the end of this loop, the quantum computer's output is now a good approximation of the optimal configuration.

2.4.2 QAOA

The Quantum Approximate Optimisation Algorithm (QAOA) is a subclass of VQAs designed to solve combinatorial optimsiation problems, introduced by Farhi et al.

2.4.3 Challenges

Although VQAs are a promising platform for NISQ computing, their effective performance relies on the computationally expensive tuning of an often prohibitively large number of variational parameters. Additionally, the performance of these variational algorithms is highly dependent on the initialisation of the variational parameters.

rewrite in own words

Therefore, it can be argued that VQAs replace one difficult optimisation problem with another difficult optimisation problem.

2.5 Quantum Walk-based Optimisation

The NVQWOA

Chapter 3

Methods

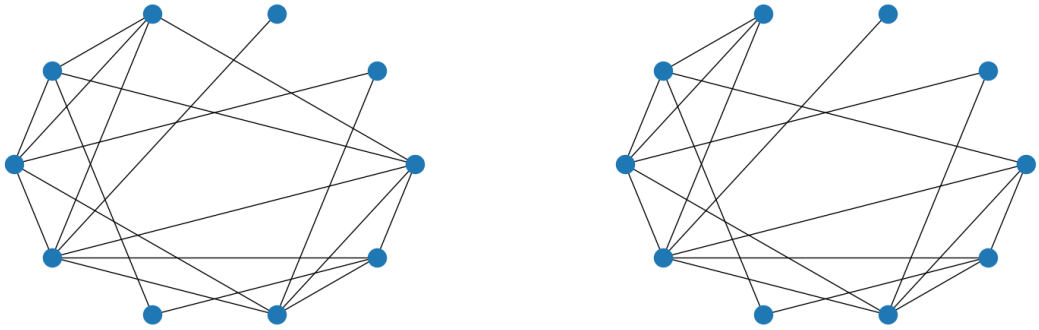
Classical simulations of the non-variational QWOA would not be possible without HPC resources, which were provided by the Pawsey Supercomputing Centre. The simulations were performed using QuOp, a Python library for parallel simulation of quantum variational algorithms. [3, 4]

3.1 Single Problem Instances

A random graph with 10 nodes was generated using the Erdős-Rényi model with an edge probability of 0.5. One edge was then removed to create a second graph, slightly dissimilar to the first. The graphs are shown in Figure 3.1. Using this construction, it is known that the optimal solution to the graph similarity problem is 0.98 (The adjacency matrices differ in two entries).

This construction was chosen because it is simple to implement and ensures that the graphs are similar, but not identical. Constructing graphs with high similarity is important to demonstrate the algorithm's ability to identify small differences between graphs, which is a key requirement for many practical applications of graph similarity, such as threat detection, social network analysis, or bioinformatics.

Since the space of permutations grows as $n!$, it becomes classically infeasible to



(a) Graph 1 (b) Graph 2

Figure 3.1: The two graphs used in the first $n = 10$ simulation. Graph 2 is derived from Graph 1 by removing one edge. Note that the nodes have been matched for visual clarity here, but node matching information is not available to the algorithm.

simulate larger graphs. However, there is reason to believe the $n = 10$ case is still illustrative of the algorithm's behaviour (the structure of the problem and the nature of the QWOA make it likely that similar patterns will emerge in larger instances). This $n = 10$ case requires $10! = 3,628,800$ permutations to be evaluated. Since every permutation corresponds to a unique basis state, this requires at least 22 qubits of storage on a quantum computer (since $2^{21} = 2,097,152 < 10! < 2^{22} = 4,194,304$).

The non-variational QWOA was simulated for 5 iterations. The initial and final probability distributions over the possible similarity scores are shown in figure 3.2. The initial distribution is approximately normal, centred around a similarity score of 0.5, which is expected since most permutations will yield a similarity score around this value. After 5 iterations of the algorithm, the distribution has shifted significantly towards higher similarity scores, with a pronounced peak at 0.7. This indicates that the algorithm has successfully concentrated probability mass on the optimal solution.

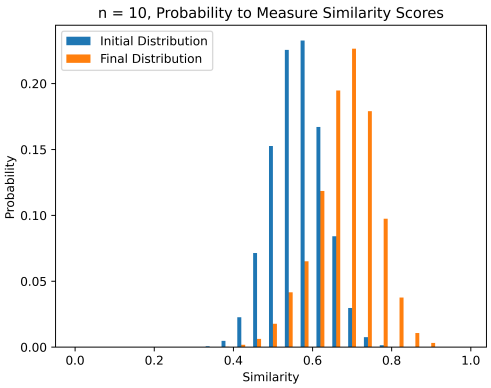


Figure 3.2: The initial and final probability distributions over similarity scores after 5 iterations of the non-variational QWOA for the $n = 10$ case. The initial distribution is approximately normal, centred around a similarity score of 0.55. After 5 iterations, the distribution has shifted significantly towards higher similarity scores, with a pronounced peak at the optimal score of 0.7.

3.2 Multiple Problem Instances

Oh yeah.

Chapter 4

Results

Chapter 5

Discussion

bapity boo

Chapter 6

Future Work

Chapter 7

Conclusions

Chapter 8

References

Bibliography

[1] Tavis Bennett, Lyle Noakes, and Jingbo Wang. Non-variational quantum combinatorial optimisation. *arXiv preprint arXiv:2404.03167*, 2024.

[2] Lov K. Grover. A fast quantum mechanical algorithm for database search. *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, pages 212–219, 1996.

[3] Edric Matwiejew and Jingbo B. Wang. Quop_mpi: a framework for parallel simulation of quantum variational algorithms. *arXiv preprint arXiv:2110.03963*, 2022.

[4] Edric Matwiejew and Nicholas Slate. Edric-matwiejew/quop_mpi: 1.0.0 - 2021-09-30 - feature release, 9 2021.

Chapter 9

Appendices