CAPSTONE 2024

On Quantum Query Complexity

AUTHORS

Syed Danial Haseeb	12429
Ashnah Khalid Khan	22889
Muhammad Rasib Nadeem	22976

ADVISOR

Dr Jibran Rashid

ABSTRACT

This paper explores new function promises for the Bernstein-Vazirani algorithm, aiming to identify configurations that yield orthogonal basis vectors, a key factor in the efficiency of quantum algorithms. We investigate Svetlichny-like functions and demonstrate that, for these functions, the bits of s can be recovered classically in n queries. Through simulation and analysis, we find that the complexity of the function terms inversely affects the basis size. Extending our study to arbitrary functions involving bits of s, we show that recoverable bits can be classically determined in n queries, establishing that only linear complexity separation is achievable. Our results highlight the limitations and potential of new function promises in quantum query complexity, offering insights for the development of future quantum algorithms.

The project also includes interactive tutorials developed as learning material designed to supplement QWorld's[1] Quantum Annealing course module. The tutorials were focused on Quantum Gram-Schmidt (QGS) algorithm and Quantum Semidefinite Programming (SDP), their practical applications and relevance to quantum optimization techniques.

School of Mathematics & Computer Science

A

Contributions

Syed Danial Haseeb 12429

- Simulation software design (60%)
- Report preparation
- Literature review (50%)
- Data analysis

Muhammad Rasib Nadeem 22976

- Requirements specification
- Simulation software design (40%)
- Literature review (50%)
- Code review

Ashnah Khalid Khan 22889

- Quantum Gram-Schmidt tutorial
- Quantum Semidefinite Programming tutorial
- Visualisations

Contents

1 Overview

This paper is divided into two distinct yet complementary parts. The first part delves into our research on quantum query complexity, specifically exploring new function promises within the Bernstein-Vazirani (BV) algorithm framework. Our investigation seeks to identify alternative function promises that result in orthogonal bases, thereby offering potential avenues for the development of new quantum algorithms.

We begin by providing a comprehensive theoretical background on quantum state discrimination, the algorithm, and the significance of function promises. This is followed by a detailed analysis of various alternative function promises, including AND functions and Svetlichny-like functions, and their impact on orthogonality and query complexity. Our primary contribution is demonstrating that, classically, bits of generalised functions can be recovered in n queries, indicating that the quantum query complexity separation for these functions is at most linear. Additionally, we explore the limitations and potential of exhaustive searches for Boolean functions, drawing connections to existing work such as Recursive BV and the Hidden Linear Function Problem.

The second part of this paper focuses on the educational materials we developed to make complex quantum computing topics more accessible and comprehensible for future students. Recognising the challenges inherent in mastering concepts like the quantum Gram-Schmidt (QGS) process and semi-definite programming, we created detailed and pedagogically sound educational content. These materials aim to simplify these advanced topics, thereby fostering a deeper understanding and encouraging further research in quantum computing.

We hope that our work not only advances the field of quantum query complexity but also serves as a valuable resource for students and researchers alike.

Part I

Quantum Query Complexity

2 Introduction

2.1 Background

Quantum computing has garnered significant interest due to its potential to solve certain computational problems exponentially faster than classical computers. One of the key areas where quantum computing has demonstrated such potential is in the realm of query complexity. The Bernstein-Vazirani (BV) algorithm is a prime example, efficiently identifying a hidden string with only a single quantum query compared to the linear (in the length of the string) number of queries required classically.[2] This efficiency is largely due to the specific form of the function promise in the BV algorithm, which results in orthogonal basis vectors, facilitating quantum state discrimination.

The success of the BV algorithm and its influence on subsequent algorithms, like Simon's and Shor's, underscore the importance of exploring new function promises. These promises could potentially lead to thedevelopment of novel quantum algorithms with enhanced efficiency. Our research is motivated by this potential, aiming to identify and analyse new function promises that also yield orthogonal bases.

2.2 Contribution

This paper makes several key contributions to the field of quantum query complexity. First, we extend the theoretical framework of quantum state discrimination to minimise error probability when dealing with non-orthogonal quantum states. Second, we investigate alternative function promises, including AND functions and Svetlichny-like functions, evaluating their impact on orthogonality and query complexity.

Our primary contribution is demonstrating that, classically, the bits of an n-bit secret string, s, in a generalised function of the form,

$$f(x) = \bigoplus_{i=1}^{n} s_i \wedge f_i(x),$$

can be recovered in n queries. This indicates that the quantum query complexity separation for these functions is at most linear. We also explore exhaustive searches for Boolean functions and discuss the implications of our findings in relation to existing work, such as Recursive BV and the Hidden Linear Function Problem.

3 Preliminaries

3.1 Query Complexity

Query complexity is a fundamental concept in quantum computing, measuring the efficiency of quantum algorithms in terms of the number of queries made to an oracle. This measure is particularly significant in the context of problems where the solution involves determining specific properties of a function with minimal access to the function's data.

Classical query complexity is often used as a baseline for comparing the efficiency of quantum algorithms. For instance, in the BV algorithm, the classical approach requires $O(n)$ queries to determine the hidden string s. However, the quantum approach, utilising a specific function promise, can solve the same problem with a single query.[3]

Promises

The efficiency of some quantum algorithms depends on specific function promises. In the case of the BV algorithm, the promise is that the function is of the form

$$f(x) = \bigoplus_{i=1}^{n} s_i \wedge x_i,$$

where s is a n-bit secret string. This form ensures that the function produces orthogonal basis vectors for different values of s, which is essential for effective quantum state discrimination.

Exploring new function promises that result in orthogonal bases could pave the way for the discovery of novel quantum algorithms with comparable or enhanced efficiencies. Our research is dedicated to identifying and analysing these promises, evaluating their impact on quantum query complexity, and determining if they can match the efficiency of the original BV promise.

By extending the theoretical framework of quantum query complexity and exploring alternative function promises, we aim to contribute to the broader understanding of quantum algorithm design and potentially uncover new pathways for exploiting quantum computational advantages.

3.2 State Discrimination

Quantum state discrimination collectively refers to quantum-informatics techniques, with the help of which, by performing a small number of measurements on a physical system, its specific quantum state can be identified, provided that the set of states in which the system can be is known in advance.

Orthogonal States

In the context of quantum algorithms, orthogonal states are crucial for efficient state readout protocols. When quantum states are orthogonal, they can be distinguished with certainty, allowing for accurate and reliable information extraction. This property is essential for the correct operation of many quantum algorithms, including the BV algorithm.

When dealing with orthogonal quantum states, one trivial strategy is to rotate the states so that they align with the computational basis, making it easy to distinguish between them.

Consider two orthogonal quantum states $|\psi_1\rangle$ and $|\psi_2\rangle$. To discriminate between these states, we can apply a unitary transformation, U that rotates the states such that $|\psi_1\rangle$ aligns with $|0\rangle$ and $|\psi_2\rangle$ aligns with $|1\rangle$. After this rotation, a measurement in the computational basis will yield the result $|0\rangle$ or $|1\rangle$ with high probability, corresponding to the original states $|\psi_1\rangle$ and $|\psi_2\rangle$, respectively (see Figure 1).

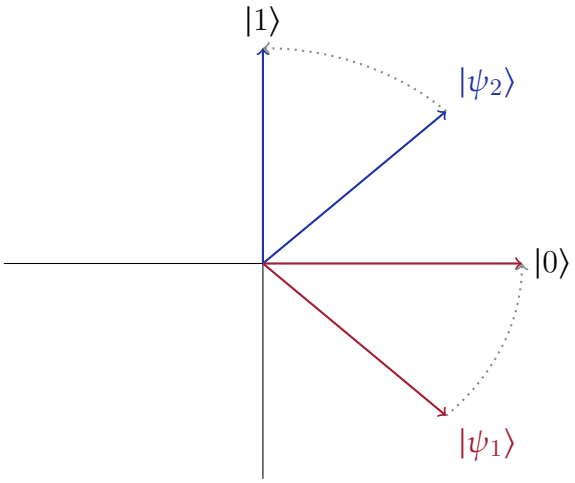

Figure 1: Orthogonal states can be perfectly distinguished.

Arbitrary States

When dealing with non-orthogonal states, the discrimination process becomes more challenging. In this case, the goal is to minimise the error probability of misidentifying the state of the system.

Theorem 3.1. *Given two arbitrary states separated by an angle $\Delta\theta$, the error probability of discriminating between them is minimised by applying a rotation gate, $R(\theta)$, where,*

$$\theta = \frac{\pi}{4} - \Delta\theta.$$

Proof. Consider two arbitrary quantum states, $|\psi_1\rangle$ and $|\psi_2\rangle$, with an angle $\Delta\theta$ between them. Without loss of generality, assume $|\psi_1\rangle = |0\rangle$. After applying a rotation, $R(\theta)$,

we have,

$$R(\theta)\,|\psi_1\rangle = \begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix}$$

The probability that $R(\theta)\psi_1$ is measured as $|1\rangle$, which is the error case, is $\sin^2\theta$. Similarly, the probability that $R(\theta)\,|\psi_2\rangle$ is measured as $|0\rangle$ is $\cos^2(\theta+\Delta\theta)$. Therefore, the total error probability is,

$$p = \sin^2\theta + \cos^2(\theta+\Delta\theta).$$

Now, our goal is to minimse p with respect to θ.

Taking the derivative, we get,

$$\begin{aligned}\frac{dp}{d\theta} &= 2\sin\theta\cos\theta - 2\sin(\theta+\Delta\theta)\cos(\theta+\Delta\theta)\\ &= \sin(2\theta) - \sin(2\theta+2\Delta\theta)\end{aligned}$$

Next, we set the derivative to zero and solve for θ,

$$\begin{aligned}\sin(2\theta) &= \sin(2\theta+2\Delta\theta)\\ 2\theta &= \pi - (2\theta+2\Delta\theta)\\ \theta &= \frac{\pi}{4} - \Delta\theta\end{aligned}$$

□

3.3 The Bernstein-Vazirani Algorithm

The BV algorithm is a cornerstone in quantum computing, showcasing the potential of quantum algorithms to outperform their classical counterparts. Introduced by Bernstein and Vazirani in 1997, this algorithm efficiently solves the problem of identifying a secret binary string with significantly fewer queries than any classical algorithm.[2]

Overview

The BV algorithm addresses the problem of finding an unknown n-bit binary string s given access to an oracle that computes the function $f(x) = s \cdot x$, where $\cdot$ denotes the bitwise inner product modulo 2. Classically, determining s would require $O(n)$ queries to the oracle, as each query provides information about a single bit of s. However, the BV algorithm leverages quantum parallelism to identify the entire string s with just a single query.

Significance

The BV algorithm has inspired the development of several other quantum algorithms, including Simon's algorithm and Shor's algorithm. Simon's algorithm extends the idea

Algorithm 1: Bernstein-Vazirani

Input: Oracle access to f
Output : Secret string, s

1. Initialize an n-qubit register and an auxiliary qubit to $|0\rangle^{\otimes n} |1\rangle$
2. Apply the Hadamard transform to all $n + 1$ qubits
3. Query the oracle, which encodes the function $f(x)$ into the phase
4. Apply the Hadamard transform again to the first n qubits
5. Measure the first n qubits, obtaining the hidden string

of finding hidden structures in functions, while Shor's algorithm applies the principles of quantum Fourier transforms to factorise large numbers efficiently.[3]

By exploring new function promises, our research aims to contribute to this ongoing development in quantum computing. Identifying function promises that result in orthogonal bases could lead to breakthroughs in algorithm design, providing new tools and techniques for solving complex computational problems more efficiently than classical approaches.

3.4 Svetlichny-like Functions

Svetlichny functions are a class of functions that extend Bell's inequalities to three-body systems, providing a framework for studying nonlocality in quantum mechanics. These functions involve multiple variables and logical operations, offering a rich space for exploring new function promises that could lead to orthogonal bases.

Lemma 3.2. *Given an oracle that implements a function $f : \{0,1\}^n \mapsto \{0,1\}$ in which f is promised to be of the form,*

$$f(x) = \bigoplus_{\sigma \in \{0,1\}^n} s_\sigma \wedge \bigwedge_{\sigma_i = 1} x_i,$$

the secret string, s, can be recovered classically in 2^n queries.

Proof. Let $w(\sigma)$ denote the Hamming weight of a string, σ.

We will begin by quering the oracle with the set of strings, $\sigma \in \{0,1\}^n \mid w(\sigma) = 1$. This will allow us to recover the bits of s corresponding to the single-bit terms in the function.

Since we now have all $s_\sigma \mid w(\sigma) = 1$, we can query the oracle with the set of strings, $\sigma \in \{0,1\}^n \mid w(\sigma) = 2$, to recover the bits of s corresponding to the two-bit terms in the function.

This is possible because we already have the bits of s corresponding to the single-bit terms, allowing us to isolate the contributions of the two-bit terms and determine the corresponding bits of s.

By repeating this process for all possible Hamming weights, we can recover the entire secret string, s, in 2^n queries. □

Corollary. $s \in \{0,1\}^n \implies$ *The oracle separation between classical and quantum query complexity for Svetlichny-like functions is at most linear.*

Proof. By Lemma 3.2, we know $s \in \{0,1\}^{2^n} \implies$ the classical query complexity is 2^n.

If we reduce the bits of s to n and apply the same querying protocol, starting with strings with the lowest Hamming weight, we can recover the bits of s in n queries. □

3.5 Fourier-like Functions

In its most general form, we allow the bits of s to be ANDed with arbitrary Boolean functions of the input variables. This generalisation encompasses a wide range of possible function promises, making the analysis more complex and challenging.

Theorem 3.3. *Given an oracle that implements a function $f : \{0,1\}^n \mapsto \{0,1\}$ in which f is promised to be of the form,*

$$f(x) = \bigoplus_{i=1}^{n} s_i \wedge f_i(x),$$

the secret string, s, can be recovered classically in 2^n queries.

Proof. Since AND and XOR form a complete basis of Boolean functions, we can express any Boolean function, $f_i(x)$, as a combination of AND and XOR operations. More precisely, we can write any Boolean function, $g : \{0,1\}^n \mapsto \{0,1\}$ with at most 2^n terms,

$$g(x) = \bigoplus_{\sigma \in \{0,1\}^n} c_\sigma \wedge \bigwedge_{\sigma_i = 1} x_i, \tag{1}$$

for some choice of coefficients, $c \in \{0,1\}^{2^n}$.

If we represent each f_i promised in f in the form of (1) and rearrange, we will arrive at a form identical to the one in Lemma 3.2. Therefore, the secret string, s, can be recovered classically in at most 2^n queries. □

4 Theoretical Framework

4.1 Function Promises and Orthogonal Bases

Orthogonal bases play a critical role in the efficiency of quantum algorithms. In quantum computing, orthogonality allows for the clear and unambiguous discrimination between quantum states, which is essential for the correct operation of many quantum algorithms. This section explores the importance of orthogonal bases in quantum algorithms and provides examples of function promises that lead to such bases.

Importance of Orthogonal Bases in Quantum Algorithms

Orthogonal bases are fundamental in quantum algorithms because they enable perfect state discrimination. When quantum states are orthogonal, measurements can distinguish them with certainty, leading to higher accuracy and efficiency in quantum computations. This property is especially vital in algorithms where the goal is to extract specific information encoded in the quantum state.

In the context of query complexity, orthogonal states ensure that each query provides maximum information, thereby reducing the number of queries needed to solve a problem. This efficiency is evident in several landmark quantum algorithms, where the use of orthogonal bases significantly outperforms classical approaches.

Generalisation of New Function Promises

Identifying new function promises that result in orthogonal bases is a key objective of our research. By generalizing the principles observed in the BV algorithm and others, we aim to discover new function promises that could lead to novel quantum algorithms with enhanced efficiency.

One promising direction is exploring composite functions involving multiple variables and logical operations. For instance, functions of the form $s_1 \wedge x_1 \oplus s_2 \wedge x_2$ or more complex combinations like Svetlichny-like functions, which incorporate higher-order terms, are potential candidates for producing orthogonal bases. These functions are analysed for their potential to yield orthogonal states and their impact on quantum query complexity.

4.2 Svetlichny-like Functions

Another promising direction for discovering new function promises involves considering more complex, higher-order functions. Svetlichny-like functions, which incorporate multiple variables and logical operations, offer an intriguing avenue for exploration. These functions are inspired by Svetlichny's work on nonlocality in quantum mechanics, which extends Bell's inequalities to three-body systems [4].

Definition and General Form

For $n = 3$, a Svetlichny-like function can be expressed in the general form:

$$f(x) = x_1 + x_2 + x_3 + x_1 \cdot x_2 + x_1 \cdot x_3 + x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_3$$

In this expression, any three of the seven terms can have a bit of the hidden string s anded with them. For example:

$$f(x) = s_1 \cdot x_1 + x_2 + x_3 + s_2 \cdot x_1 \cdot x_2 + x_1 \cdot x_3 + s_3 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_3$$

Classical Recovery of Bits of s

We first demonstrate that, classically, the bits of s can be recovered in n queries to an oracle. We start by analyzing a more general case:

$$f(x) = c_1 \cdot x_1 + c_2 \cdot x_2 + c_3 \cdot x_3 + c_4 \cdot x_1 \cdot x_2 + c_5 \cdot x_1 \cdot x_3 + c_6 \cdot x_2 \cdot x_3 + c_7 \cdot x_1 \cdot x_2 \cdot x_3$$

We show that each bit of c can be systematically recovered in $|c|$ queries to the oracle.

This general case can be trivially modified to prove our claim for the Svetlichny case. The systematic recovery process involves querying the oracle with carefully chosen inputs to isolate the contributions of each term, thereby allowing the determination of the corresponding bits of c.

Simulation and Basis Enumeration

We simulated all possible Svetlichny-like functions and enumerated the bases they produced. Our findings indicate that the closer a Svetlichny-like function is to the original BV promise—i.e., the bits of s are anded with single bits of x instead of compound terms—the larger the size of the basis obtained.

Specifically:

- If all the bits of s are with the singular bits of x, as in the BV algorithm, a complete basis is achieved.
- If one of the bits of s is on another term, the basis size is halved.
- If two of the bits of s are on other terms, the basis size is quartered, and so on.

Orthogonal Basis and Max-Clique Reduction

An interesting implementation detail is how we solved the problem of finding the largest subset of mutually orthogonal vectors from a given set. This problem initially appeared to require looping through all subsets, but we managed to reduce it to a max-clique problem, significantly reducing its practical complexity due to the availability of off-the-shelf solutions.

The max-clique problem, a well-known problem in graph theory, involves finding the largest clique (a subset of vertices, all adjacent to each other) in a given graph. By mapping the problem of finding mutually orthogonal vectors to this graph-theoretic problem, we leveraged efficient algorithms to make the problem tractable.

Implications for Quantum Query Complexity

Our analysis of Svetlichny-like functions provides valuable insights into the structure and properties of function promises that can potentially lead to orthogonal bases. While these functions introduce complexity beyond the original BV promise, they offer a framework for understanding how higher-order terms and interactions affect the orthogonality and query complexity of the resulting quantum states.

5 Generalised Function Promises

In addition to exploring specific function forms such as the AND function and Svetlichny-like functions, we also investigated more generalized function promises. These generalized functions involve arbitrary Boolean functions combined with the bits of the hidden string s.

5.1 Arbitrary Functions

For a given n, a generalized function promise can be expressed as:

$$f(x) = s_1 \cdot f_1(x) + s_2 \cdot f_2(x) + s_3 \cdot f_3(x)$$

where $f_1(x)$, $f_2(x)$, and $f_3(x)$ are arbitrary Boolean functions. This form allows for a broad range of possible functions, making the analysis significantly more complex.

5.2 Classical Recovery of Bits

We demonstrate that, for certain choices of the Boolean functions f_1, f_2, and f_3, the bits of s can still be recovered classically in n queries. This involves reducing the problem to our previous general case:

$$c_1 \cdot x_1 + c_2 \cdot x_2 + c_3 \cdot x_3 + c_4 \cdot x_1 \cdot x_2 + c_5 \cdot x_1 \cdot x_3 + c_6 \cdot x_2 \cdot x_3 + c_7 \cdot x_1 \cdot x_2 \cdot x_3$$

Using this reduction, we show that, with appropriate queries, each bit of s can be systematically determined, thus ensuring that the classical query complexity remains linear. This result implies that the quantum query complexity separation for these generalized functions cannot be more than linear.

5.3 Cases with Impossible Recovery

However, we also identify cases where the bits of s are not recoverable, regardless of the number of queries. These cases occur when the structure of the functions f_1, f_2, and f_3 introduces dependencies or correlations that obscure the individual bits of s. In such scenarios, no classical or quantum queries can reliably extract the hidden information.

5.4 Implications for Quantum Query Complexity

The analysis of generalized function promises reveals several important insights:

- When the bits of s are recoverable, they can be extracted with linear query complexity, indicating that the quantum advantage in these cases is limited to a constant factor.

- For functions where the bits of s are not recoverable, the problem becomes fundamentally intractable, highlighting the limitations of certain function promises.

This comprehensive analysis underscores the challenges and potential of using generalized function promises in quantum algorithms. While some functions maintain the linear separation, others present insurmountable obstacles, emphasizing the need for careful selection and design of function promises to achieve desired quantum advantages.

5.5 Exhaustive Search of Boolean Functions

Given the promising results from specific function promises like the AND function and Svetlichny-like functions, we extended our investigation to a more systematic exploration. This involved conducting an exhaustive search across all possible Boolean function choices to identify any that might result in orthogonal bases.

Problem Definition

The goal of this exhaustive search is to find Boolean functions $f_1, f_2, \ldots, f_n$ such that the function promise

$$f(x) = s_1 \cdot f_1(x) + s_2 \cdot f_2(x) + \cdots + s_n \cdot f_n(x)$$

yields orthogonal bases. If such functions can be identified, they could lead to new quantum algorithms with the same or better efficiency as the BV algorithm.

Challenges Due to Large Search Space

The primary challenge of this approach is the vastness of the search space. The number of possible Boolean functions grows exponentially with the number of variables. For n variables, there are 2^{2^n} possible Boolean functions. This exponential growth makes a brute-force search impractical, even for relatively small n.

Implications for Quantum Algorithm Design

The insights gained from the exhaustive search of Boolean functions have important implications for the design of quantum algorithms. By understanding which functions and structures tend to produce orthogonal bases, we can better target our efforts in algorithm development. This knowledge can inform the creation of new quantum algorithms that exploit these function promises to achieve superior performance.

6 Connections to Existing Work

6.1 Recursive BV Algorithm

The Recursive BV algorithm is an extension of the original BV algorithm that further explores the efficiency of quantum algorithms in identifying hidden information. This section examines the Recursive BV algorithm and its relevance to our work on new function promises.

Overview of the Recursive BV Algorithm

The Recursive BV algorithm builds on the principles of the original BV algorithm, applying them in a recursive manner to solve more complex problems. In the Recursive BV algorithm, the hidden string is divided into smaller substrings, each of which is identified using the BV procedure. The results from these smaller problems are then combined to recover the entire hidden string efficiently [5].

This recursive approach enhances the scalability of the BV algorithm, making it applicable to larger instances and more complex structures of hidden information. By leveraging the efficiency of the BV promise at each recursive step, the algorithm maintains a high level of performance.

6.2 The Hidden Linear Function Problem

The Hidden Linear Function Problem (HLFP) is another significant area in quantum computing, closely related to our research on function promises. This section discusses the HLFP and its connections to our work.

Overview of the HLFP

The HLFP involves identifying a hidden linear function given oracle access to a function that encodes this hidden linear relationship. The problem is formally defined as follows: given an oracle that computes a function $f(x) = A \cdot x + b$, where A is a hidden matrix and b is a hidden vector, the task is to determine A and b using the fewest possible queries.[6]

Quantum algorithms can solve the HLFP more efficiently than classical algorithms by exploiting the linear structure of the problem. This efficiency arises from the ability of quantum algorithms to process superpositions of inputs and leverage quantum interference.

Part II

Interactive Learning Material

7 Introduction

7.1 Background

During our research on quantum query complexity, we realized the significant challenge of conveying and understanding complex quantum computing topics and the unavailability of educational materials that did so comprehensively. With this motivation, we added an additional component to our project with an emphasis on developing instructional materials that can effectively bridge the gap between complex quantum theories and practical understanding.

The project was extended to develop learning material for key advanced quantum computing techniques explored during our research. These materials consisted of tutorials in the form of Jupyter notebooks[7], visual and interactive demos[8][9], and assessments, designed to supplement QWorld's Quantum Annealing course module. The topics covered in these tutorials included the Quantum Gram-Schmidt (QGS) algorithm and Quantum Semidefinite Programming (SDP), with a particular focus on their applications and relation to our research parallel. These tutorials are intended for students, educators, and researchers interested in quantum computing, and particularly quantum optimization techniques.

7.2 Contribution

The learning material we have developed will be used as supplementary material in the hands-on training workshops for QWorld's Quantum Annealing course module.

We hope that our contribution of these resources will aid in academic learning and prepare students for potential research and application in the field, paving the way for the next generation of quantum computing scientists and engineers.

8 Quantum Gram-Schmidt Algorithm

This tutorial was intended for intermediate learners with a basic understanding of quantum computing and linear algebra concepts to build upon the idea of orthogonality and its importance in quantum computing. The course was also designed to include assessments and programming exercises to provide hands-on practice and reinforcement of the concepts explained in the tutorials. The course also assumes learners are equipped with a basic level of programming expertise in Python.

8.1 Course Structure

The course is divided into 3 sections:

- Jupyter notebook lectures
- Programming exercises
- Quizzes

8.2 Learning Objectives

- Understand what 'orthogonality' is and what it means for a set of vectors to be orthogonal.
- Demonstrate the difference between orthogonal and un-orthogonal vectors using 2D and 3D visualizations in Python.
- Reinforce the concept of vector projections.
- Explain how vector projections can be utilized to produce an orthogonal vector given a pair of vectors.
- Demonstrate the finding of orthogonal vectors from vector projections using 2D and 3D visualizations in Python.
- Utilize Python and NumPy to deduce the projection of a vector on another vector.
- Explain the significance of the classical Gram-Schmidt algorithm in transforming a set of n vectors each with n entries, into a set of vectors where each vector is orthogonal to all other vectors in the set.
- Understand the step-by-step implementation of the classical Gram-Schmidt algorithm.
- Program the implementation of classical Gram-Schmidt algorithm in Python to obtain a set of orthogonal vectors for any vectors of size n.

- Introduce the Quantum Gram-Schmidt (QGS) algorithm, and understand its step-by-step implementation.
- Explain the quantum advantage of QGS over classical Gram-Schmidt algorithm.
- Program the implementation of the QGS algorithm in Python to obtain a set of orthogonal vectors for any vectors of size n.
- Discuss the applications of orthogonality in practical scenarios such as machine learning, image reconstruction, geometric transformation in computer graphics, cryptography, signal processing etc.
- Discuss the importance of orthogonality in quantum computing applications, specifically quantum optimization techniques.
- Discuss how assuming we knew all quantum states at the beginning of our research problem, we could easily obtain an orthogonal set of vectors from it using QGS.

8.3 Assessments

The assessments are designed to be used in tandem with each tutorial notebook to not only test the understanding of the taught topics, but also help learners practice programming implementations of common techniques and algorithms.

- Determining if a set of vectors is an orthogonal set of vectors.
- Finding the projection of a vector on another vector.
- Finding a set of orthogonal vectors from a given set of vectors using the classical Gram-Schmidt algorithm.
- Finding a set of orthogonal vectors from a given set of vectors using the Quantum Gram-Schmidt algorithm.

9 Quantum Semidefinite Programming

This tutorial was designed for learners with a grasp on the foundations of semidefinite programming, and build upon this to understand and explore Quantum semidefinite programming and advanced Quantum SDP applications. The tutorial utilizes the extensive resources on Quantum SDP developed by Jamie Sikora.[10] It also includes assessments and programming exercises, and assumes learners are equipped with a basic level of programming expertise in Python.

9.1 Course Structure

The course is also divided into 3 sections:

- Jupyter notebook lectures
- Programming exercises
- Quizzes

In addition, this tutorial incorporates visual representations of semidefinite programming to aid learners further in their understanding of the topic.

9.2 Learning Objectives

- Understand the concepts of positive semidefinite, positive definite, negative semidefinite, and negative definite matrices.
- Explain the concept of feasibility and infeasibility, bounded and unboundedness of a semidefinite program.
- Demonstrate the feasible and infeasible regions in a semidefinite program using 3D models.
- Explain the application of Quantum SDP in state distinguishability.
- Describe the use of SDPs for the Quantum Fidelity Function.
- Understand the application of Quantum SDP in solving the MAXCUT as a Boolean Quadratic Problem.
- Understand the modelling of Quantum Merlin-Arthur (QMA) Protocols as semidefinite programs to solve them.

9.3 Assessments

- Distinguish between positive/negative definite/semi-definite matrices.
- Determine the feasibility and infeasibility of a region in an SDP.
- Determine if an SDP is bound or unbound.

10 Connections to Existing Work

The learning materials for the Quantum SDP topics were developed using the extensive list of SDP resources created most notably by Jamie Sikora, John Watrous[11]**WatrousSDP2**, and multiple quantum computing course materials from the University of Waterloo, University of Maryland and the University of California, Berkeley.

References

[1] QWorld, *Qworld - quantum computing network*, https://qworld.net/, Accessed: 2024-05-31, 2024.

[2] E. Bernstein and U. Vazirani, "Quantum complexity theory," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1411–1473, 1997.

[3] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, "Quantum algorithms revisited," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1969, pp. 339–354, 1998.

[4] G. Svetlichny, "Distinguishing three-body from two-body nonseparability by a bell-type inequality," in *Physical Review D*, APS, vol. 35, 1987, p. 3066.

[5] A. M. Childs. "Lecture notes on quantum algorithms." 19 September 2022. (2022), [Online]. Available: https://www.cs.umd.edu/~amchilds/qa/qa.pdf.

[6] R. O'Donnell, *Analysis of Boolean Functions*. Cambridge University Press, Jun. 5, 2014.

[7] *Learning material/tutorial*, https://drive.google.com/drive/folders/1x3rWHSz3vuJ-g20ZmZpcCfw9HnFd6ziw?usp=sharing, Accessed: 2024-05-31, 2024.

[8] *Quantum gram-schmidt symposium demo slides*, https://www.canva.com/design/DAGGxU7-E64/Qbzpk6nproCNOKmce4_56w/edit?utm_content=DAGGxU7-E64&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton, Accessed: 2024-05-31, 2024.

[9] *Quantum semidefinite programming symposium demo slides*, https://www.canva.com/design/DAGGxac7mq0/KgCYSPz3Rai6G-k7dH6f2Q/edit?utm_content=DAGGxac7mq0&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton, Accessed: 2024-05-31, 2024.

[10] J. Sikora, *Semidefinite programming resources*, https://sites.google.com/site/jamiesikora/sdp-resources?authuser=0, Accessed: 2024-05-31, 2024.

[11] J. Watrous, "Semidefinite programming," University of Waterloo, Lecture Notes, 2008, Available at https://johnwatrous.com/wp-content/uploads/TQI-notes.07.pdf.

Acknowledgements

We would like to extend our deepest gratitude to Dr Jibran Rashid, our advisor, for his invaluable guidance and for originating the idea for this research. His insights and encouragement were crucial to the development and completion of this work.

We also wish to thank Dr Imran Rauf for his assistance with the reduction to the max-clique problem, which significantly contributed to the tractability and success of our simulations.

TEX

This capstone project report was written in TEX (created by Donald Knuth) and typeset using the LATEX typesetting system (created by Leslie Lamport); specifically, its `memoir` class (created by Peter Wilson). The body text is set 12pt with Palatino (designed by Hermann Zapf), which includes *italics* and SMALL CAPS.