2022

Quantum Graph Parameters

Parisa Darbari Kozekanan
University of Central Florida

QUANTUM GRAPH PARAMETERS

by

PARISA DARBARI KOZEKANAN
M.S. University of Central Florida, 2018

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2022

Major Professor: Gary T. Leavens

ABSTRACT

This dissertation considers some of the advantages, and limits, of applying quantum computing to solve two important graph problems. The first is estimating a graph's quantum chromatic number. The quantum chromatic number is the minimum number of colors necessary in a two-player game where the players cannot communicate but share an entangled state and must convince a referee with probability one that they have a proper vertex coloring. We establish several spectral lower bounds for the quantum chromatic number. These lower bounds extend the well-known Hoffman lower bound for the classical chromatic number.

The second is the Pattern Matching on Labeled Graphs Problem (PMLG). Here the objective is to match a string (called a pattern) P to a walk in an edge labeled graph $G = (V, E)$. In addition to providing a new quantum algorithm for PMLG, this work establishes conditional lower bounds on the time complexity of any quantum algorithm for PMLG. These include a conditional lower bound based on the recently proposed NC-QSETH and a reduction from the Longest Common Subsequence Problem (LCS). For PMLG where substitutions are allowed to the pattern, our results demonstrate that (i) a quantum algorithm running in time $O(|E|m^{1-\varepsilon} + |E|^{1-\varepsilon}m)$ for any constant $\varepsilon > 0$ would provide an algorithm for LCS on two strings X and Y running in time $\tilde{O}(|X||Y|^{1-\varepsilon} + |X|^{1-\varepsilon}|Y|)$, which is better than any known quantum algorithm for LCS, and (ii) a quantum algorithm running in time $O(|E|m^{\frac{1}{2}-\varepsilon} + |E|^{\frac{1}{2}-\varepsilon}m)$ would violate NC-QSETH. Results (i) and (ii) hold even when restricted to binary alphabets for P and the edge labels in G. Our quantum algorithm is for all versions of PMLG (exact, only substitutions, and substitutions/insertions/deletions) and runs in time $\tilde{O}(\sqrt{|V||E|} \cdot m)$, making it an improvement over the classical $O(|E|m)$ time algorithm when the graph is non-sparse.

TABLE OF CONTENTS

LIST OF FIGURES vii

CHAPTER 1: INTRODUCTION 1

Quantum Chromatic Number 2

Pattern Matching on Labeled Graphs 5

CHAPTER 2: TECHNICAL PRELIMINARIES 8

Lower Bounds on Quantum Chromatic Number 8

Linear Algebra 8

Graphs and Adjacency/Laplacian Matrices 11

Quantum Mechanics 13

Bra-ket Notation 13

States and Superposition 14

Pseudo-Telepathy Games 16

Quantum Chromatic Number 16

Pattern Matching on Labeled Graphs 20

Problem Definitions 20

Fine-grained Lower Bounds 22

Quantum Computing 23

Grover's Algorithm and Quantum Graph Algorithms 25

NC-QSETH and Superlinear Quantum Hardness Results 26

CHAPTER 3: LOWER BOUNDS ON QUANTUM CHROMATIC NUMBER 30

New bounds for the quantum chromatic number 32

First new bound for $\chi_q(G)$ 33

Second new bound for $\chi_q(G)$ 36

Implications for quantum chromatic number 39

Strongly regular graphs (SRGs) 39

Non-SRGs 40

Hoffman colorings 41

CHAPTER 4: QUANTUM ALGORITHMS FOR PATTERN MATCHING ON LABELED GRAPHS 43

Lower Bounds 43

Main Reduction to PMLG-sub over Large Alphabet 43

Hardness of PMLG-edit over Large Alphabet 46

Hardness of PMLG-sub over Binary Alphabet 47

Quantum Algorithm for PMLG 50

Construction 52

Adjacency Oracles 54

Modifications for PMLG-sub 55

Modifications for PMLG-exact 55

CHAPTER 5: CONCLUSION 56

Questions Related to Quantum Chromatic Number 56

Questions Related to PMLG 57

LIST OF REFERENCES 58

LIST OF FIGURES

Figure 2.1: A simple undirected graph to illustrate adjacency, degree, and Laplacian matrices 11

Figure 2.2: The Hadamard graph for $n = 4$. The vertex 0000 is adjacent to vertices 0011, 0101, 0110, 1001, 1010, and 1100. The corresponding edges are shown in blue. 20

Figure 3.1: A Clebsch Graph (16,5,0,2) 39

Figure 3.2: A Schlafi Graph (27,16,10,8) 42

Figure 4.1: Reduction from LCS to PMLG for $X = aababbbab$. Note that dashed edges are only shown from v_1 but similar edges are present from every v_i, $1 \leq i \leq n$. If $Y = baabbabaa$ then $P = \#^4b\#^4a\#^4a\#^4b\#^4b\#^4a\#^4b\#^4a\#^4a$. 44

Figure 4.2: The conversion of an edge in G with label α from u to v to two paths between u and v in G'. 47

Figure 4.3: The alignment graph G' for PMLG-edit constructed from the starting graph G shown above it. The edges with label I correspond to insertion and have weight 1, edges with label S correspond to substitution and have weight 1, edges with label D correspond to deletion and have weight 1, black edge correspond to exact match and have weight 0. 53

CHAPTER 1: INTRODUCTION

The problems studied in this dissertation lie in the intersection of quantum computing and computational problems on graphs. There are two main problems. The first is the quantum chromatic number, a generalization of the traditional chromatic number of a graph to the quantum setting. The quantum chromatic number relates to two-party games and the minimal amount of information that must be shared between the parties to successfully win the game. Here we prove new lower bounds on this value. The second is the problem of (approximately) matching a string or text to a walk within an edge labeled graph. We establish new computational complexity results that imply the conditional hardness of this problem even for quantum computers. We are also able to provide a quantum algorithm that provides a speed up over the traditional algorithm for many graphs.

These results reveal some of the limitations, and advantages, of applying quantum mechanics to solve problems on graphs. For the quantum chromatic number, we obtain a limitation on how much entanglement can reduce the size of a vertex coloring that two parties must share in order to convince a third party that they have a proper coloring. It has been shown that this can be less than the traditional chromatic number when quantum entanglement is utilized. However, our results provide a new, stronger lower bound on how much smaller this coloring can be in terms of the eigenvalues of the graph's adjacency matrix. For the problem of matching patterns to labeled graphs, we are able to show that for non-sparse graphs a quantum computer can offer some computational advantage over the traditional solution. However, a reduction from the longest common subsequence (LCS) problem also suggests the difficulty of using quantum computing to obtain an algorithm that always outperforms the classical solution.

We next present a brief historical overview of the two main problems considered here, as well as a high-level overview of our results.

Quantum Chromatic Number

The chromatic number of a graph is one of the major classical definitions in graph theory. Aside from applications including scheduling [63] and register allocation [24], finding a graph's chromatic number is one of the first problems considered in the study of NP-completeness, being one of Karp's original 21 NP-complete problems [55]. The chromatic number of a graph is the minimum number of distinct integers that must be assigned to the vertices of that graph such that no two adjacent vertices are assigned the same integer. An alternative, somewhat more complicated, characterization of the chromatic number can be given in terms of two-party one-round games as follows (a more formal definition is given in Chapter 2): Consider two players Alice and Bob, and a referee. The game played by the three of them is that Alice and Bob are given a graph $G = (V, E)$ and allowed to communicate and pick a strategy before the first round. After this, Alice and Bob are not allowed to communicate. Next the referee gives a vertex $u \in V$ to Alice and a vertex $v \in V$ to Bob. Alice and Bob must each report an integer, a and b respectively, such that

(i) $u = v$ implies $a = b$,

(ii) $(u, v) \in E$ implies $a \neq b$.

Alice and Bob win the game if they can always fulfill these conditions. In order for Alice and Bob to answer the referee correctly with probability 1, they must assign integers to the vertices in G, and the minimum number of distinct integers they assign to these vertices is the chromatic number.

One way to introduce the quantum chromatic number is through a generalization of the above game where Alice and Bob are allowed to share an entangled state that they can measure after being given their query vertex from the referee. This sort of game was studied under the name pseudo-telepathy [14, 18, 29, 17, 9, 13, 40, 39] before the term quantum chromatic number was formally defined by Peter Cameron [22] as the minimum number of integers Alice and Bob need to assign to the vertices of G given that they can share an entangled state. For some graphs, this can be smaller than the classical chromatic number [22]. In this way, the difference between the classical chromatic number and the quantum chromatic number demonstrates how sharing an entangled state can give players increased power in two-player single-round games.

More recent work on quantum chromatic numbers includes introducing a commuting variant [68] and a method for computing this commuting variation through solving a sequence of semi-definite programs (SDPs) [67]. The definition of quantum chromatic number is based on relativistic field theory, and the idea that the operators used by Alice and Bob to measure the shared entangled state act on the same space and mutually commute. Since we will not be establishing any new results on commuting quantum chromatic numbers, we refer the interested reader to Paulsen et al.'s work [67]. It should be noted that on of the main motivations for introducing quantum chromatic number is disproving Tsirelson's conjecture by showing that there exists a graph such that its commuting quantum chromatic number differs from its quantum chromatic number. However, Tsirelson's conjecture was subsequently proven false using different techniques [53], undercutting this motivation. It remains unknown whether the commuting quantum chromatic number can be smaller than the quantum chromatic number for some graphs. It also remains open whether computing the quantum chromatic number is even a decidable problem.

In lieu of an algorithm for computing the quantum chromatic number, our results provide a

lower bound on the quantum chromatic number by utilizing spectral graph theory. Spectral graph theory is the study of the adjacency matrix or Laplacian matrix of a graph. Spectral graph theory has shown that the eigenvalues of these matrices can provide fundamental information on a graph, such as information about its connectivity, number of components, and chromatic number. We refer the interested reader to Brouwer et al.'s work [16]. Our result will build on the well known Hoffman lower bound [50] that bounds the chromatic number of a given graph in terms of the eigenvalues of the graph's adjacency matrix. Hoffman proved that for a graph G where its adjacency matrix has eigenvalues $\mu_1 \geq \ldots \geq \mu_n$ that the chromatic number $\chi(G)$ satisfies $\chi(G) \geq 1 + \kappa$ where κ is the smallest integer such that

$$\mu_1 + \sum_{i=1}^{\kappa} \mu_{n+1-i} \leq 0.$$

We strengthen this result by proving that $\chi(G)$ can be replaced by the quantum chromatic number, $\chi_q(G)$. We also prove another lower bound for the quantum chromatic number,

$$\chi_q(G) \geq 1 + \min\{g, \frac{|\mu_n|}{\mu_2}\}$$

for graphs where $\mu_2 > 0$ and where g is the multiplicity of μ_n, the smallest eigenvalue. Finally, we investigate implications of these inequalities for the quantum chromatic number of various classes of graphs such as strongly regular graphs and some non-strongly regular graphs (defined formally in Chapter 2). For example, we demonstrate that the Kneser graph (formally defined in Chapter 4) $KG_{p,2}$ has $\chi_q = \chi = p - 2$.

Pattern Matching on Labeled Graphs

Pattern matching on labeled graphs (PMLG) is an interesting problem from both an applied and a theoretical standpoint. The exact case, where the given string or pattern must exactly match a walk in the given edge-labeled graph can be reduced to the classical problem of finding whether a nondeterministic finite automata (NFA) accepts a given string. For deterministic finite automata (DFA) and other restricted classes of graphs, this problem can be solved in linear time [37]. For general NFAs, a standard algorithm that keeps track of the current set of active states works for this task, and there exist several potential improvements [51, 69, 23]. We will consider the more general case where edits, such as substitutions, insertions, and deletions, can be made to the pattern. This approximate-PMLG was first considered in the context of matching patterns to hypertext [8, 61, 65]. It has more recently become an increasingly important problem in Computational Biology. This is mainly due to the use of labeled graphs as multi-genomic references where sets of reads obtained through sequencing must be mapped to a multi-genomic reference [66, 26, 33, 41, 58, 1] as well as other applications in computational biology, such as variant calling [25, 32, 49] and error correction [59, 64]. These data sets are often large, making the quadratic time algorithm discussed next impractical in most cases. In most of the software tools cited above, seed-and-extend heuristics are used as well to provide increased performance.

From a theoretical viewpoint, the problem has also received recent interest, particularly in the realm of fine-grained-complexity, described in more detail in Chapter 2. The classical algorithm for PMLG is a dynamic programming solution equivalent to finding a shortest path through an alignment graph of size $O(|E|m)$ like that shown in Figure 4.3 in Chapter 4. It was first shown by Equi et al. that an algorithm running in time $O(|E|^{1-\varepsilon}m + |E|m^{1-\varepsilon})$ for any constant $\varepsilon > 0$ would contradict the Strong Exponential Time Hypothesis even for

directed acyclic graphs (DAGs) in the exact matching case [35]. These results were later strengthened by Gibney et al. who showed that the same lower bounds can be based on likely weaker assumptions in circuit complexity [42].

Despite the extent of the applied and theoretical research, PMLG has yet to have been considered in terms of quantum computing. Several closely related problems have been considered though. Aaronson et al. considered quantum algorithms for the problem of determining whether a string is generated by a regular expression [2]. However, these regular expressions are represented in monoid form rather than has an NFA. For finding exact matches in a single string, which could be viewed as a path, there exists a quantum algorithm running in $\tilde{O}(\sqrt{n} + \sqrt{m})$ time[1] for a string of length n and pattern of length m [48]. Applied work focusing on the implementation of quantum algorithms for DNA read mapping to a single sequence under Hamming distance was presented in work by Sarkar et al. [70].

We initiate the study of quantum algorithms for PMLG. We look at this problem in terms of both computational complexity and possible algorithmic improvements arising from quantum computing. First, we provide a reduction from the longest common subsequence (LCS) problem on two strings X and Y to approximate PMLG with subsitutions to the pattern on a labeled graph $G = (V, E)$ over a binary alphabet. This reduction proves that an algorithm running in time $O(|E|^{1-\varepsilon}m + |E|m^{1-\varepsilon})$ for $\varepsilon > 0$ would provide an algorithm running in time $O(|X|^{1-\varepsilon}|Y| + |X||Y|^{1-\varepsilon})$ for LCS (faster than any known quantum algorithm for LCS). Moreover, this reduction implies that no $\tilde{O}(|E|^{\frac{1}{2}-\varepsilon}m + |E|m^{\frac{1}{2}-\varepsilon})$ time algorithm is possible under a proposed variant of the Strong Exponential Time Hypothesis for quantum computing, NC-QSETH [21] (defined in Chapter 2). We then provide an algorithm running in time $\tilde{O}(m\sqrt{|V||E|})$, implying that quantum speedup is possible in the case where the

[1] $\tilde{O}(t(n))$ hides polylogarithmic factors, i.e., $t(n)\log^c n \in \tilde{O}(t(n))$ for all constants c.

graph is even slightly dense, i.e., $|E| = \Omega(|V|^{1+\varepsilon})$. As these combined results suggest, the reduction from LCS to PMLG creates a sparse graph. A subquadratic reduction from LCS to a dense graph would provide a new, improved algorithm for LCS, suggesting the difficulty of finding such a reduction. Moreover, the graph created by the reduction is cyclic. Results discussed in the conclusion of this dissertation suggest it may be difficult to construct a reduction from LCS to exact PMLG on DAGs.

CHAPTER 2: TECHNICAL PRELIMINARIES

This chapter provides the technical background for understanding the results and proofs presented in the rest of this dissertation. The first section is devoted to the technical preliminaries required for Chapter 3 and the second section is for Chapter 4.

Lower Bounds on Quantum Chromatic Number

There are two equivalent definitions of the quantum chromatic number. The first is in terms of pseudo-telepathy games, and the second is exclusively in terms of sets of projectors. Although our results primarily use the second definition, the motivation for the second definition is best understood through the first. To explain these, we must first provide some technical preliminaries on linear algebra, graph theory, quantum mechanics, and pseudo-telepathy games.

Linear Algebra

One of the most fundamental attributes of a matrix are its eigenvalues (also referred to as its *spectrum*) and its eigenvectors. These describe subspaces, known as eigenspaces, such that vectors are only scaled by some constant under the linear transformation represented by the matrix. The formal definition is presented below.

Definition 1 (Eigenvalues and Eigenvectors)**.** *For a matrix A, an* eigenvector of A *is a nonzero vector v such that there exists a complex number c where $Av = cv$. The scalar c is called the* eigenvalue of *matrix A corresponding to the eigenvector v.*

The *rank* of a matrix is the number of non-zero eigenvalues (counting multiplicities). For symmetric matrices the spectrum of the matrix consists of all real numbers. For a square matrix A of graph $G = (V, E)$, we use the notation $\mu_1^{\uparrow}(A) \leq \mu_2^{\uparrow}(A) \leq \cdots \leq \mu_n^{\uparrow}(A)$ to denote the eigenvalues of A in increasing sorted order where $\mu_1^{\uparrow}(A)$ is the smallest. The same notation, but without arrows for eigenvalues, means that the eigenvalues are in decreasing order, so μ_1 is the largest and μ_n is the smallest. The multiplicity of the maximum eigenvalue $\mu_{\max}$ is 1 since G is connected.[1]

For a matrix A, we let a_{ij} be the entry on the i^{th} row and j^{th} column of A.

Definition 2 (Trace)**.** *The* trace *of an $n \times n$ square matrix A is the sum of its main diagonal, that is* $\mathrm{Tr}(A) = \sum_{i=1}^{n} a_{ii}$.

The following properties of the trace follow directly from its definition:

- $\mathrm{Tr}(A + B) = \mathrm{Tr}(A) + \mathrm{Tr}(B)$.
- $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$.
- $\mathrm{Tr}(A)$ is the sum of the eigenvalues of A.

Definitions 3-7 are particularly important for understanding the definitions arising from quantum mechanics discussed later.

Definition 3 (Tensor Products)**.** *Let A be an $n \times m$ matrix and B be a $p \times q$ matrix. The*

[1]The Perron-Frobenius theorem implies that the maximum eigenvalue $\mu_{\max}$ has multiplicity 1 because the adjacency matrix A of the connected graph G is irreducible.

tensor product of A and B *is an* $(n \cdot p) \times (m \cdot q)$ *matrix denoted by* $A \otimes B$ *and defined by*

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1m}B \\ & \ddots & \\ a_{n1}B & \dots & a_{nm}B \end{bmatrix}.$$

Definition 4 (Conjugate Transpose)**.** *The* conjugate transpose *of an* $n \times m$ *matrix* A *is an* $m \times n$ *matrix* B *such that* $b_{ij} = \overline{a_{ji}}$ *for* $1 \leq i \leq m$, $1 \leq j \leq n$.[2] *The conjugate transpose of* A *is denoted by* $A^\dagger$.

Definition 5 (Self-Adjoint)**.** *For a square matrix* A*, if* $A = A^\dagger$*, then* A *is called* self-adjoint.

Definition 6 (Orthogonal Projector)**.** *A square matrix* P *is an* orthogonal projector *if* $P^2 = P = P^\dagger$.

Definition 7 (Mutually Orthogonal)**.** *A set of square matrices of the same dimension* $\{P_1, \dots, P_n\}$ *are* mutually orthogonal *if* $P_i P_j = 0$ *for all* $i \neq j$.

Definition 8 (Span)**.** *The span of a set of* d *dimensional vectors* V *over* $\mathbb{C}$ *consists of all linear combinations of vectors in* V*, i.e.,*

$$\text{span } V = \{x \in \mathbb{C}^d \mid v = \sum_{i=1}^{k} \alpha_i v_i, k \in \mathbb{N}, v_i \in V, \alpha_i \in \mathbb{C}\}.$$

The next type of matrix will come up later in our discussion of quantum computing.

Definition 9 (Unitary)**.** *A matrix* U *is unitary if* $UU^\dagger = I$.

[2] $\overline{a}$ is the complex conjugate of a.

Graphs and Adjacency/Laplacian Matrices

An unweighted, undirected graph with no loops or multiple edges is called a *simple graph*. All of the graphs considered in Chapter 3 are simple graphs. For undirected graphs if the

Definition 10 (Adjacency Matrix)**.** *Let $G = (V, E)$ be a graph whose vertex set is $V = \{v_1, \ldots, v_n\}$. An* adjacency matrix *A for G is $n \times n$ matrix where $a_{ij} = 1$ iff there exists an edge from v_i to v_j and 0 otherwise.*

Definition 11 (Degree Matrix)**.** *The degree matrix D of a graph $G = (V, E)$ where $V = \{v_1, \ldots, v_n\}$ is an $n \times n$ diagonal matrix. Letting $\deg(v_i)$ denote the degree of vertex v_i, the diagonal elements of D are $d_{ii} = \deg(v_i)$ for $1 \leq i \leq n$. All other entries of D are zero.*

Definition 12 (Laplacian Matrix)**.** *The Laplacian matrix of a graph $G = (V, E)$ is the $n \times n$ matrix $L = D - A$, where $|V| = n$, D is the degree matrix of G, and A is the adjacency matrix of G.*

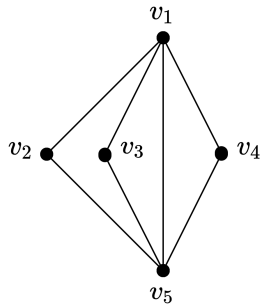

Figure 2.1: A simple undirected graph to illustrate adjacency, degree, and Laplacian matrices

As an example, the adjacency matrix A, degree matrix D, and Laplacian matrix L of the

graph in Figure 2.1 are:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix} \quad L = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 2 & 0 & 0 & -1 \\ -1 & 0 & 2 & 0 & -1 \\ -1 & 0 & 0 & 2 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{bmatrix}$$

The following graph related definitions will also be used in Chapter 3.

Definition 13. *Let $G = (V, E)$ be a given graph. The set $C \subset V$ is said to be a* clique *if any two distinct vertices in C are adjacent. A clique C is said to be a maximum-sized clique if there does not exist a clique with more vertices than C.*

For example, in Figure 2.1, a maximum-sized clique is the triangle containing vertices v_1, v_3, and v_5.

Definition 14 (Clique number)**.** *The clique number of a graph $G = (V, E)$, denoted by $\omega(G)$, is the number of vertices in a maximum-sized clique.*

Definition 15 (Orthogonal representation)**.** *The orthogonal representation of graph a $G = (V, E)$ is a collection of non-zero vectors x_v for $v \in V$ satisfying the orthogonality condition $x_v^\dagger x_w = 0$ for all $(v, w) \in E$.*

Definition 16 (Orthogonal rank)**.** *The orthogonal rank of a graph G is the smallest positive integer $\xi(G)$ such that there exists an orthogonal representation for G using vectors of dimension $\xi(G)$.*

Definition 17 (Normalized orthogonal rank)**.** *The normalized orthogonal rank of G is the smallest positive integer $\xi'(G)$ such that there exists an orthogonal representation, with the*

added restriction that the entries of each vector must all have the same modulus[3].

We will also be showing the implications of our lower bounds on the quantum chromatic number for strongly regular graphs (SRGs), defined below.

Definition 18 (Strongly regular graphs)**.** *A graph is* strongly regular *if there exists integers* λ *and* μ *such that every two adjacent vertices have* λ *common neighbors and every two non-adjacent vertices have* μ *common neighbors.*

A graph with v vertices, degree k, and parameters λ, and μ (as defined above), is said to be in the set $\mathrm{SRG}(v, k, \lambda, \mu)$.

Quantum Mechanics

In this subsection, we introduce some necessary notations and definitions from quantum mechanics.

Bra-ket Notation

We will use '*bra-ket*' notation, as is standard in quantum mechanics. An $n \times 1$ column vector will be denoted using the *ket* notation $|\phi\rangle = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_n \end{bmatrix}$. Its $1 \times n$ conjugate transpose is denoted

[3]Modulus here refers to the 2-norm $\|\cdot\|_2$

using the *bra* notation $\langle\phi| = \begin{bmatrix}\overline{\phi_1} & \dots & \overline{\phi_n}\end{bmatrix}$. An $n \times n$ matrix A can be applied to $|\phi\rangle$ as

$$A|\phi\rangle = \begin{bmatrix}\sum_{i=1}^{n} a_{1i}\phi_i \\ \vdots \\ \sum_{i=1}^{n} a_{ni}\phi_i\end{bmatrix}$$

The *inner product* of two $n \times 1$ vectors $|\phi\rangle$ and $|\psi\rangle$ is defined as $\langle\phi|\psi\rangle = \sum_{i=1}^{n} \overline{\phi_i}\psi_i \in \mathbb{C}$. The *outer product* of an $n \times 1$ vector $|\phi\rangle$ and an $m \times 1$ vector $|\psi\rangle$ is defined as

$$|\phi\rangle\langle\psi| = \begin{bmatrix}\phi_1\overline{\psi_1} & \phi_1\overline{\psi_2} & \dots & \phi_1\overline{\psi_m} \\ \phi_2\overline{\psi_1} & \phi_2\overline{\psi_2} & \dots & \phi_2\overline{\psi_m} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_n\overline{\psi_1} & \phi_n\overline{\psi_2} & \dots & \phi_n\overline{\psi_m}\end{bmatrix} \in \mathbb{C}^{n\times m}.$$

States and Superposition

The vectors used in quantum mechanics are referred to as *states*. In general, a state is a linear combination (superposition) of other states, formally defined below. Let $|i\rangle \in \mathbb{C}^n$ be the $n \times 1$ vector with 1 in the i^{th} row and 0 elsewhere.

Definition 19 (Superposition)**.** *A* quantum state *is a unit vector $|\phi\rangle$ that is a linear combination of vectors $|1\rangle, |2\rangle, \dots, |n\rangle$, i.e.,*

$$|\phi\rangle = \sum_{i=1}^{n} \alpha_i |i\rangle,$$

where for each i, $\alpha_i \in \mathbb{C}$, and $\sum_{i=1}^{n} |\alpha_i|^2 = 1$. The vectors $|1\rangle, |2\rangle, \dots, |n\rangle$ are called classical states *and the linear combination $|\phi\rangle = \sum_{i=1}^{n} \alpha_i|i\rangle$ is a* superposition *of classical states.*

An 'operator A' is a map between two mathematical spaces[4]. For our purposes, we can view operators as linear maps between finite-dimensional vector spaces of the same dimension, and thus, as square matrices. A *positive operator* can then be defined in several ways. Given the standard inner product on $\mathbb{C}^n$, a positive (semi-definite)[5] operator has the property that for all $v \in \mathbb{C}^n, \langle Av|v\rangle = \sum_{i=1}^n \sum_{j=1}^n \overline{a_{ij}v_j}v_i \geq 0$. It can be shown that this is equivalent to all of the eigenvalues of A being non-negative [73].

Definition 20 (Positive Operator-Valued Measure (POVMs))**.** *A set of positive operators* $\{P_1, \ldots, P_k\}$ *is called a* POVM measurement *if* $\sum_{i=1}^k P_i = I_n$*, where* I_n *is the* $n \times n$ *identity matrix.*

For example, the operators $\{|i\rangle\langle i| \mid 1 \leq i \leq n\}$ are POVMs of rank 1 since $\sum_{i=1}^n |i\rangle\langle i| = I_n$.

The state is measured using a set of POVMs $\{P_1, \ldots, P_k\}$ and obtain a *measurement* (or *outcome*) $j \in \{1, \ldots, k\}$. The probability of obtaining outcome j given state $|\phi\rangle = \sum_{i=1}^n \alpha_i|i\rangle$ is $\mathrm{Tr}(\rho P_j)$ where $\rho = \sum_{i=1}^n |\alpha_i|^2|i\rangle\langle i|$.

The last notion we require is entanglement. A quantum state is *separable* relative to two sets of basis vectors $\{|i_1\rangle, |i_2\rangle, \ldots, |i_n\rangle\}$ and $\{|j_1\rangle, |i_2\rangle, \ldots, |j_m\rangle\}$ if it can be written as the tensor product

$$\left(\sum_{h=1}^{n} \alpha_h|i_h\rangle\right) \otimes \left(\sum_{h=1}^{m} \beta_h|j_h\rangle\right), \qquad \alpha_h, \beta_k \in \mathbb{C}, \quad 1 \leq h \leq n, \quad 1 \leq k \leq m.$$

If a state is not separable, it is called *entangled*.

[4]The definition of 'mathematical space' depends on the properties the space is endowed with, e.g., inner product space.

[5]We will omit 'semi-definite' and say positive operator A.

Pseudo-Telepathy Games

A *pseudo-telepathy game* is a game for two players, conventionally called Alice and Bob. Formally, it is defined as a six-tuple (X, Y, A, B, P, W), where X and Y are input sets, A and B are called output sets. A set $P \in X \times Y$ is called the *promise set*. A set $W \in A \times B \times X \times Y$, is known as a *winning condition*, which is a relation between inputs and outputs that must be satisfied by Alice and Bob whenever the promise is fulfilled.

Before the first round, the players, Alice and Bob, can exchange any information. They are also allowed to share an entangled state of any dimension. When the game begins, Alice and Bob are not allowed to communicate anymore. Alice is given $x \in X$ and her task is to produce the output $a \in A$ based on x and measurements to the shared entangled state. Similarly, Bob is given $y \in Y$, and his task is to produce the output $b \in B$ based on y and the shared entangled state. The pair (x, y) is called the *question* and the pair (a, b) is called the *answer*. The question (x, y) is said to be *legitimate* if $(x, y) \in P$ and the answer (a, b) is said to be *appropriate* if $(x, y, a, b) \in W$. Alice and Bob will *win the round* if they are given a question that is not legitimate or their answer is appropriate, that is $(x, y) \notin P$ or $(x, y, a, b) \in W$. Alice and Bob *win* the game if they win every round. If Alice and Bob have a winning strategy they are mathematically certain to win every round.

For the quantum chromatic number, we will only consider single-round games.

Quantum Chromatic Number

We recall our discussion of quantum chromatic number from Chapter 2 and see how it combines with quantum mechanics to provide an alternative definition in terms of orthogonal matrices.

Recall from Chapter 1 that a proper vertex coloring of a graph $G = (V, E)$ assigns a natural number (or color) to every vertex in such a way that if $u, v \in V$ and $(u, v) \in E$, u and v are assigned different natural numbers. The minimum number of natural numbers required to form a proper graph coloring of G is called G's *chromatic number* and denoted $\chi(G)$. In the non-quantum version of our game, Alice and Bob are given a graph $G = (V, E)$ and allowed to communicate and pick a strategy prior to the first round. Once the first round starts a referee gives a vertex $u \in V$ to Alice and a vertex $v \in V$ to Bob. Alice and Bob must each report an integer, a and b respectively, where the answer is appropriate if and only if

(i) $u = v$ implies $a = b$,

(ii) $(u, v) \in E$ implies $a \neq b$.

It can be shown that $\{1, 2, \ldots, \chi(G)\}$, is the smallest set of values that Alice and Bob must assign to vertices of G in order to have a winning strategy.

Returning to Pseudo-Telepathy games, a natural question (asked by Cameron et al. [22]) is whether Alice and Bob having a shared entangled state allow them to use fewer, say c, vertex colors. Given the vertex u, the strategy now adopted by Alice is to measure the entangled state $|\phi\rangle$ using the POVMs $\{E_{ui} : 1 \leq i \leq c\}$ and if k is the outcome, report k. Given the vertex v the strategy now adopted by Bob is to measure the entangled state $|\phi\rangle$ using the POVMs $\{F_{vj} : j \in [c]\}$ and if h is the outcome, report h. Conditions (i) and (ii) from the definition of the non-quantum game are reflected in the definition of Quantum c-coloring in terms of these POVMs.

Definition 21 (Quantum c-coloring)**.** *A* quantum c-coloring *of the graph $G = (V, E)$ is two collections of POVMs $\{E_{v,i} : v \in V, i \in [c]\}$ and $\{F_{v,i} : v \in V, i \in [c]\}$ and entangled state $|\phi\rangle$ such that for all $u, v \in V$ and $i, j \in [c]$,*

(i) $u = v$ *and* $i \neq j$ *implies* $\langle\phi|E_{u,i} \otimes F_{v,j}|\phi\rangle = 0$,

(ii) $(u, v) \in E$ *implies* $\langle\phi|E_{u,i} \otimes F_{v,i}|\phi\rangle = 0$ *for all* $i \in [c]$.

Definition 22 (Quantum chromatic number)**.** *The* quantum chromatic number *of a graph* G, $\chi_q(G)$, *is the smallest* c *for which a quantum* c*-coloring exists.*

This was further simplified by Cameron et al. [22] who demonstrated that we can assume the shared entangled state is always the same entangled state and that Alice and Bob share a single set of POVMs per vertex, $\{P_{v,i} : v \in V, i \in [c]\}$. The strategy for Alice given the vertex u is to use the POVMs $\{P_{u,i} : i \in [c]\}$ for measurement and the strategy for Bob given the vertex v is to use the POVMs $\{\overline{P_{v,i}} : i \in [c]\}$.

Corollary 1 (Quantum c-coloring)**.** *A quantum* c*-coloring of the graph* $G = (V, E)$ *is a collection of orthogonal projectors* $\{P_{v,k} : v \in V, k \in [c]\}$ *in* $\mathbb{C}^{d\times d}$ *all of the same rank such that for all vertices* $v \in V$

$$\sum_{k\in[c]} P_{v,k} = I_d$$

for all edges $vw \in E$*, and for all* $k \in [c]$, $P_{v,k}P_{w,k} = 0_d$.

Here the dimension d and the projector's ranks are not fixed. Based on its previous definition, $\chi_q(G)$ is the minimum c for which a quantum c-coloring exists over all dimensions d. Letting d vary allows additional graph parameters to be defined. For a graph $G = (V, E)$, we define the minimum c for which a quantum c-coloring exists using projectors of rank r (exactly), and a maximally entangled state of dimension rc as the *rank* r *quantum chromatic number*, $\chi_q^{(r)}(G)$ [22]. Note that bounding the rank r of the projectors and enforcing the bound $\chi_q^{(r)}(G) \leq c$ also bounds the dimension d. This is since, for all graphs G, $d = \chi_q^{(r)}(G)\cdot r \leq c\cdot r$.

It is also shown in Cameron et al. [22] that for all graphs G that:

- $r < r'$ implies $\chi_q^{(r')}(G) \leq \chi_q^{(r)}(G)$;
- $\chi_q(G) = \inf_r\{\chi_q^{(r)}(G)\}$, and
- $\chi_q^{(1)}(G)$ is an upper bound for for $\chi_q(G)$.

A classical c-coloring also gives rise to a quantum c-coloring in dimension $d = 1$. The orthogonal projectors in the resulting quantum c-coloring are the scalars 0 and 1. These operators are rank 1, which implies for all graphs G,

$$\chi_q(G) \leq \chi_q^{(1)}(G) \leq \chi(G). \tag{2.1}$$

Moreover, the separation between $\chi(G)$ and $\chi_q^{(1)}(G)$ can be exponentially large for a class of graphs known as Hadamard graphs [71, 72].

Definition 23 (Hadamard Graph, [57])**.** *A* Hadamard graph $G_n = (V, E)$ *is the graph whose vertex set is the set of vectors in* $\{0, 1\}^n$*. There exists an edge between two vertices iff the hamming distance of their corresponding vectors is* $\frac{n}{2}$*.*

See Figure 2.2 for an example.

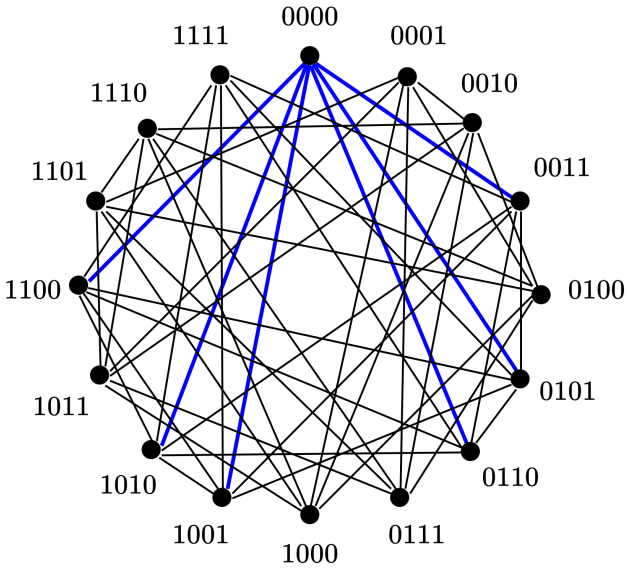

Figure 2.2: The Hadamard graph for $n = 4$. The vertex 0000 is adjacent to vertices 0011, 0101, 0110, 1001, 1010, and 1100. The corresponding edges are shown in blue.

Another important aspect of the rank r quantum chromatic number is that by bounding the rank, the problem of checking if $\chi_q(G) = c$ becomes decidable. Its computational complexity remains unknown, however.

Pattern Matching on Labeled Graphs

Problem Definitions

This section formally defines the PMLG problem and the variants that we will consider. In this section and Chapter 4 we allow graphs to be multi-graphs with multi-edges and self-loops.

Definition 24 (Walk matching a pattern)**.** *For a given directed edge-labeled graph $G = (V, E)$, a walk is a ordered list of edges in E, i.e., e_1, ..., e_m where e_i and e_{i+1} are incident to the same vertex for $1 \leq i < m$. Edges are allowed to be repeated in a walk. Letting $label(e)$ denote the edge label for an edge $e \in E$, we say a length m string (also referred to as the pattern) $P[1, m]$ matches a walk e_1, ..., e_m if $P[i] = label(e_i)$ for $1 \leq i \leq m$.*

Definition 25 (Exact Pattern Matching on Labeled Graphs (PMLG-exact))**.** *Given a directed graph $G = (V, E)$, having edges labeled with symbols from an alphabet Σ, a pattern P of length m also over alphabet Σ, determine if there is a walk in G that matches P.*

Let $d_H(P, P')$ be the *Hamming distance* between strings P and P', i.e., the minimum number of symbols in P that must be substituted so that the resulting string is identical to P'.

Definition 26 (Pattern Matching on Labeled Graphs with Substitutions (PMLG-sub))**.** *Given a directed graph $G = (V, E)$, having edges labeled with symbols from an alphabet Σ, a pattern P of length m also over alphabet Σ, and $\delta \geq 0$, determine if there exists a walk on G that matches a string P' such that $d_H(P, P') \leq \delta$.*

Let $d_E(P, P')$ be the *edit distance* between strings P and P', i.e., the minimum total number of symbol insertions, deletions, and substitutions that must be made to P so that the resulting string is identical to P'.

Definition 27 (Pattern Matching on Labeled Graphs with Substitutions (PMLG-edit))**.** *Given a directed graph $G = (V, E)$, labeled with symbols from an alphabet Σ, a pattern P of length m also over alphabet Σ, and $\delta \geq 0$, determine if there exists a walk on G that matches a string P' such that $d_E(P, P') \leq \delta$.*

We use $\circ$ to denote concatenation. Recall that a subsequence of a string $X = X[1] \circ X[2] \circ \cdots \circ X[n]$ is a string $X[i_1] \circ X[i_2] \circ \cdots \circ X[i_k]$ where $1 \leq i_1 < i_2 < \cdots < i_k \leq n$. Here k is the

length of the subsequence. A string is called a common subsequence of strings X and Y if it is a subsequence of both X and Y. Our results for PMLG will make use of the following well-known problem.

Definition 28 (Longest Common Subsequence (LCS))**.** *Given two strings X and Y and integer k, determine if there exists a subsequence contained in both X and Y that is of length at least k.*

As discussed in Chapter 1, both PMLG and LCS have been studied in terms of computational complexity. We next briefly discuss some of the main ideas utilized in fine-grained complexity that will also appear in this work.

Fine-grained Lower Bounds

Fine-grained complexity is primarily focused on proving conditional hardness results for problems with polynomial-time solutions. This hardness is most commonly based on popular conjectures such as the Strong Exponential Hypothesis (SETH) [52], the hardness of 3-SUM [38], and the hardness of the All-Pairs-Shortest-Path problem [76]. Examples of problems where conditional lower bounds of this sort have been established include the Edit Distance problem [10, 6], the Longest Common Subsequence problem [4, 5], Subtree Isomorphism [3], graph diameter [11], and PMLG [35, 42], to name only a few.

One key aspect of proving a conditional lower bound such as those discussed above is that the reduction used must be efficient enough to establish the desired hardness result. For example, consider the case where a Karp reduction from problem A of size n to problem B is being used, i.e., a single instance of problem B is being creating from the given instance of problem A. If the conjectured hardness of problem A is that there does not exist a $O(n^{2-\varepsilon})$

time algorithm for problem A for any $\varepsilon > 0$, then the reduction should run in strongly sub-quadratic time. Otherwise, no matter how efficient an algorithm there was for problem B, it would not (through the reduction, at least) imply a strongly sub-quadratic time algorithm for problem A. Our reduction from an instance of LCS on strings X and Y to an instance of PMLG runs in time $\tilde{O}(|X| + |Y|)$, allowing for all the desired hardness results for PMLG to be established.

The Strong Exponential Time Hypothesis (SETH) is that for all $\gamma < 1$ there exists some k where k-SAT cannot be solved in time $2^{\gamma n + o(n)}$ [52]. Support for SETH comes from the best-known algorithms for k-SAT requiring time approaching $2^{n+o(n)}$ as k increases. This was first applied to prove the hardness of problems with polynomial-time algorithms by Williams who reduced it to an intermediate problem known as the orthogonal vectors problem [75]. A subquadratic time reduction was later provided to edit distance [10] and the longest common subsequence problem [4], demonstrating that a strongly subquadratic algorithm for either would contradict SETH. We hold off on the description of the quantum extension of SETH until after the discussion on quantum computing.

Quantum Computing

Quantum algorithms are often represented as quantum circuits, where the system state is captured by the state of wires (each representing a basis vector and whose tensor product makes up the overall system state) and the unitary operators acting on the state are represented by quantum gates. A quantum circuit consisting of these wires and gates indicates the order in which the state is acted on by the operators, and thus describes an algorithm. IBM's Quisik provides an example of a modern set of programming tools based around this paradigm [28]. This paradigm is useful for describing fundamental quantum algorithms such

as Grover's search, discussed next. However, it is possible for programs written in a high-level, imperative programming style to utilize quantum computing as subroutines [74], and under the commonly used assumption of quantum random access memory, also possible for quantum algorithms to invoke classical algorithm subroutines with only a constant factor of overhead in terms of the number of quantum gates [7, 19]. In particular, assuming quantum random access, a classical algorithm running in T-time can be converted into a quantum subroutine in $O(T)$ time. This allows for high-level descriptions of algorithms that apply data structures and other programming constructs not easily expressible in the language of quantum circuits. A high-level description of quantum algorithms will be the approach taken in this work. As such, we will not be covering quantum circuits in any depth.

For quantum algorithms the input instance is most commonly expressed as an *input oracle*. These input oracles represent problem instances as functions that allow one to query the input. These are often treated as black boxes, where a single query allows one to recover a Boolean output. One example is an instance of 3-SAT on n variables taking as input an $0-1$ assignment to these variables and the output being either 1 representing the 3-SAT instance evaluating to true on this input and a 0 representing the 3-SAT instance evaluating to false on this input. We can more generally assume that the oracle can return a value $[0, m-1]$ represented as a binary encoding since this could be decomposed into $\lceil \log m \rceil$ Boolean oracles. It is also possible for the oracle descriptions to be provided algorithmically.

The query complexity of a quantum algorithm is defined as the number of times that the input oracle is queried by the algorithm. The time complexity of a quantum algorithm is the number of queries made by the algorithm in addition to the number of quantum gates used in the circuit. We will be concerned with both query complexity and time complexity. Using known lower bounds, or other common techniques for establishing lower bounds on query complexity, has an immediate limitation for proving superlinear lower bounds on quantum

time complexity for problems where the input represents something such as a graph. Once a linear number of queries have been made the entire input is obtained by the algorithm. Then, since the whole input as been obtained, the problem can be solved without making any additional queries. An alternative approach taken by Buhrman et al. [20] is to use conditional hardness results and efficient reductions, as is typically done in fine-grained complexity. The main hypothesis used here for establishing lower bounds, NC-QSETH is described in Section 2 following the description of some fundamental quantum algorithms.

Grover's Algorithm and Quantum Graph Algorithms

Grover's algorithm [44] is a fundamental building block of many quantum algorithms. Given an oracle $f : \{1, ..., N\} \mapsto \{0, 1\}$ it allows one to determine an index i such that $f(i) = 1$, or declare one does not exists with constant probability using $\tilde{O}(\sqrt{N})$ queries to the oracle. Grover's algorithm can be adapted to work for oracles f that map onto the natural number to return the index mapping to the minimum value [31], and also to count the number of indices mapping to non-zero values [15]. Although, as mentioned above, an oracle may be described algorithmically, Grover's algorithm is indifferent to the details of this description and treats f only as a black box.

Our quantum algorithm for PMLG will utilize Durr et al.'s quantum algorithm for computing the shortest path between two vertices in a directed graph [30]. We will assume that the representation of these graphs is in terms of adjacency lists rather than an adjacency matrix. Further details are given when necessary in Chapter 4. Durr et al.'s algorithm is based on adapting Dijkstra's algorithm well-known algorithm to apply Grover's search. On a graph $G = (V, E)$, Durr et al.'s algorithm has query complexity $O(\sqrt{|V||E|} \log^2 |V|)$. For the case where the graph G is a DAG, this was later improved by Khadiev and Safina to a query

complexity of $O(\sqrt{|V||E|}\log|V|)$ [56]. Both of these run in quantum time that is within poly-logarithmic factors of their query complexity and solve the *st*-shortest path problem correctly with constant probability greater than $\frac{1}{2}$. Our algorithm is based on the observation that we can modify the input oracles given for a PMLG instance to be oracles for the corresponding *sequence alignment graph* that can answer queries with only constant overhead. Finding the shortest path in this seqeuence alignment graph corresponds to finding an optimal walk in G. Our algorithm prevents the construction of the actual sequence alignment graph, which is defined formally in Section 4) and is of size $O(|E|m)$.

NC-QSETH and Superlinear Quantum Hardness Results

Utilizing Grover's search, Conjunctive Normal Form (CNF)-SAT can be solved in $2^{\frac{n}{2}+o(n)}$ time. Due to this, many problems with quadratic classical time complexity lower bounds based on SETH have only linear lower bounds when the analogous conjecture is made for quantum computation, that is, for some k large enough $2^{\frac{n}{2}+o(n)}$ time is required to solve k-SAT. However, unlike in the traditional case, with edit distance and LCS problems this causes a polynomially large gap between the best known quantum algorithm and the lower bound. This is since for both edit distance and LCS, the best known quantum algorithms are not strongly subquadratic. Motivated by this gap, Buhrman et al. defined the NC Quantum Strong Exponential Time Hypothesis, or NC-QSETH [21]. To formally define this, we need several definitions.

Definition 29 (Boolean Function Properties [21])**.** *Given a Boolean functions $f : \{0,1\}^n \to \{0,1\}$, and $P : \{0,1\}^{2^n} \to \{0,1\}$. We say property P holds on f if $P(tt(f)) = 1$ where $tt(f)$ is the truth table (list of output values ordered according to the query input order) of f.*

A popular example of a property is P being 1 iff there exists a 1 in $tt(f)$. This corresponds to

f being satisfiable. Another is P being 1 iff there only exists 1's in $tt(f)$. This corresponds to f being a tautology. In the case where we can query a Boolean circuit describing f, it is assumed that the description is at least as complex as a polynomial-sized depth 2 Boolean circuit consisting of AND and OR gates.

Definition 30 (White box time complexity $qTimeWB_\epsilon(P)$ [21])**.** *The time taken by a quantum computer to decide whether property P holds with probability at least ε given a Boolean circuit description of f is called the white box time complexity of P, written $qTimeWB_\epsilon(P)$.*

Definition 31 (Black box time complexity $qTimeBB_\epsilon(P)$ [21])**.** *The time taken by a quantum computer to decide whether property P holds with probability at least ε given only oracle access to f is called the black box time complexity of P, written $qTimeBB_\epsilon(P)$.*

Let C be a circuit representation of some Boolean function and γ be some subset of Boolean circuits, e.g., polynomial-sized depth 2 Boolean circuits consisting of AND and OR gates.

Definition 32 (Compression Oblivious $\mathcal{CO}(\gamma)$ Properties [21])**.** *The set of* compression oblivious properties $\mathcal{CO}(\gamma)$ *is exactly the set of properties P such that*

$$qTimeBB_\epsilon(P|_{S_\gamma}) \geq \Omega(Q_\varepsilon(P))$$

where

- $S_\gamma = \{tt(C) \mid C$ *is an element of* $\gamma\}$,
- $qTimeBB_\epsilon(P|_{S_\gamma})$ *is the black box time complexity to determine if P holds when the input set of oracles is restricted to those whose truth tables are in S_γ ,*
- $Q_\varepsilon(P)$ *is the quantum query complexity, or the minimum number of queries necessary to determine with probability at least ε that property P holds, taken over all oracles.*

In short, compression oblivious properties are those where the restriction to only f with $tt(f) \in S_\gamma$ still makes the query complexity lower bounded by the query complexity across all oracles.

We also need the notion of NC and branching programs. Here, NC refers to the set of polynomial-sized circuits of poly-logarithmic depth consisting of fan-in 2 gates (AND, OR, NOT gates with two inputs), rather than the set of all decision problems decidable using such circuits as is used in other contexts. A *branching program* consists of a set of n Boolean variables x_1,..., x_n and a leveled DAG with a single source vertex *start* on the first level and two sink vertices labeled *accept* and *reject* respectively on the last level. Let ℓ be the number of levels, also referred to as the branching programs length. Each level besides the first and last consists of W vertices called the branching program's width, and being leveled, edges occur only from level i to level $i+1$ for $1 \leq i \leq \ell - 1$. Every edge is labeled x_j or $\bar{x_j}$ for some $1 \leq j \leq n$. A particular assignment to the variables x_1, ..., x_n is said to satisfy the branching program if there exists a path from *start* to *accept* following only edges whose labels evaluate to true under the assignment. Barrington's theorem [12] is a famous result that states for every branching program of constant width there is an equivalent circuit in NC with $O(\log n)$ depth, and for every circuit in NC with depth d there is an equivalent branching program with width 5 and length 4^d. Equivalent here means that the branching program is satisfied by the Boolean assignment on n variables if and only if the circuit in NC is. Hence, it makes sense to talk about the NC representations of particular subsets of branching programs, $\mathcal{S}$, as is done next.

The set $\mathcal{S}$ used for defining NC-QSETH is a set of non-deterministic branching programs where, given the matrix representation of their truth table, the truth table contains a 'path' satisfying certain requirements. To avoid further descriptions of these paths we refer the reader to Buhrman et al's work [21] for more details. NC-QSETH states that for compression

oblivious properties satisfied by NC circuits for branching programs in $\mathcal{S}$, the white box quantum time is bounded from below by the quantum query complexity on all branching programs in $\mathcal{S}$, represented by their NC description.

Definition 33 (NC-QSETH [21])**.** *For the class of representations NC, i.e., the set of polynomial sized circuits of polylogarithmic depth consisting of fan-in 2 gates, for all properties $P \in \mathcal{CO}(NC \cap \mathcal{S})$, we have $qTimeWB_\epsilon(P|_{NC\cap\mathcal{S}}) \geq \Omega(Q_\epsilon(P|_{\mathcal{S}}))$.*

Burnham et al. prove the following conditional lower bounds based on NC-QSETH, restated here as a lemma for our purposes.

Lemma 1 (NC-QSETH LCS lower bounds [21])**.** *Under NC-QSETH the longest common subsequence problem (LCS) on two sequences of length n cannot be solved in time $O(n^{1.5-\varepsilon})$ for any constant $\varepsilon > 0$.*

Ultimately, the reduction presented in this work is from the LCS problem on strings X and Y to an instance of PMLG $G = (V, E)$ and $P[1, m]$ where $|E| = |X|$ and $m = |Y|$. Therefore, the details for NC-QSETH described prior to Lemma 1 are not needed to understand how an algorithm running in time $O(|E|^{\frac{1}{2}-\varepsilon}m + |E|m^{\frac{1}{2}-\varepsilon})$ for PMLG contradicts NC-QSETH. However, the above exploration of NC-QSETH illustrates some of the consequences of such an algorithm. Such an algorithm would imply that having white box access to an NC circuit representing inputs in $\mathcal{S}$ would provide an algorithm with quantum query complexity superior to the query lower bound possible without white box access. In other words, having access to the underlying circuit description provides a benefit over simply being able to query the circuit for different inputs, contradicting NC-QSETH. Of course, a more direct result is that a $O(|E|^{1-\varepsilon}m + |E|m^{1-\varepsilon})$ time algorithm would give a quantum algorithm for LCS running in time $O(|X||Y|^{1-\varepsilon} + |X|^{1-\varepsilon}|Y|)$, which itself would still be a new and interesting result.

CHAPTER 3: LOWER BOUNDS ON QUANTUM CHROMATIC NUMBER

An earlier version of this work appeared in [78].

Elphick and Wocjan [34] proved that many spectral lower bounds for $\chi(G)$ are also lower bounds for $\chi_q(G)$ by using a linear algebra technique called pinching. Definition 34 defines what pinching is.

Definition 34 (Pinching)**.** *Let $P_k \in \mathbb{C}^{m\times m}$ for $k \in [c]$ be orthogonal projectors such that they form a resolution of the identity, that is,*

$$\sum_{k\in[c]} P_k = I_m. \tag{3.1}$$

The operation $\mathcal{D} : \mathbb{C}^{m\times m} \to \mathbb{C}^{m\times m}$ defined by

$$X \mapsto \mathcal{D}(X) = \sum_{k\in[c]} P_k X P_k \tag{3.2}$$

is called pinching. We say that it annihilates X if $\mathcal{D}(X) = 0$.

The following theorem is stated as Theorem 1 in [34]. For the sake of completeness, we include a condensed proof below. We use $\{e_v : v \in V\}$ to denote the standard basis in $\mathbb{C}^n$.

Theorem 1. *Let $\{P_{v,k} : v \in V, k \in [c]\}$ be an arbitrary quantum c-coloring of G in $\mathbb{C}^d$. Then, the following block-diagonal projectors*

$$P_k = \sum_{v\in V} e_v e_v^\dagger \otimes P_{v,k} \in \mathbb{C}^{n\times n} \otimes \mathbb{C}^{d\times d} \tag{3.3}$$

define a pinching operation that annihilates $A \otimes I_d$, that is,

$$\sum_{k\in[c]} P_k(A \otimes I_d)P_k = 0. \tag{3.4}$$

Proof. We have

$$\begin{aligned}
\sum_{k\in[c]} P_k &= \sum_{k\in[c]}\sum_{v\in V} e_v e_v^\dagger \otimes P_{v,k} && (3.5)\\
&= \sum_{v\in V} e_v e_v^\dagger \otimes \sum_{k\in[c]} P_{v,k} && (3.6)\\
&= \sum_{v\in V} e_v e_v^\dagger \otimes I_d && (3.7)\\
&= I_n \otimes I_d. && (3.8)
\end{aligned}$$

This shows that the orthogonal projectors P_k form a resolution of the identity, that is, form a pinching operation.

For $v, w \in V$, let a_{vw} denote the entries of the adjacency matrix. For $k \in [c]$, we have

$$P_k(A \otimes I_d)P_k = \sum_{v,w\in V} a_{vw} \cdot e_v e_w^\dagger \otimes P_{v,k}P_{w,k}. \tag{3.9}$$

Whenever $a_{vw} = 1$, or equivalently $(v, w) \in E$, the corresponding orthogonal projectors $P_{v,k}$ and $P_{w,k}$ must be orthogonal. This shows that the above sum is equal to 0_d, that is, the corresponding pinching operation annihilates $A \otimes I_d$. □

Remark 1. *In the classical case the projectors P_k are a diagonal in the standard basis of $\mathbb{C}^n$ and each projector corresponds to a color class, that is, each projector P_k projects onto the subspace spanned by the standard basis vectors corresponding to the vertices that have been colored with the k^{th} color.*

The following lemma is a standard interlacing result in matrix analysis. We describe it in detail since it is the main result that we rely on to prove the new stronger bounds on the quantum chromatic number.

Lemma 2. *Let $X \in \mathbb{C}^{n\times n}$ be a hermitian matrix and let $S \in \mathbb{C}^{n\times m}$ be a matrix such that $S^\dagger S = I_m$, that is, its m column vectors $s_1, \ldots, s_m$ are orthonormal vectors. Then,*

$$\mu_i^\uparrow(X) \leq \mu_i^\uparrow(S^\dagger X S) \tag{3.10}$$

for $i \in [m]$.

Proof. This follows from the Courant-Weyl-Fisher theorem stating

$$\mu_i^\uparrow(X) = \min_{\mathcal{M}_i} \max_{x\in\mathcal{M}_i} x^\dagger X x, \tag{3.11}$$

where the minimum is taken over subspaces $\mathcal{M}_i$ of dimension i and the maximum is taken over unit vectors $x \in \mathcal{M}_i$. By multiplying X by $S^\dagger$ and S from the left and right, respectively, we effectively restrict the subspaces $\mathcal{M}_i$ to have the form

$$\mathcal{M}_i = \{Sy : y \in \mathcal{N}_i\} \tag{3.12}$$

where $\mathcal{N}_i$ is a subspace of $\mathbb{C}^m$ of dimension $i \in [m]$. □

We obtain the following corollary by using the identity $\mu_i^\uparrow(-X) = -\mu_i(X)$ and applying the above lemma to the matrix $-X$.

Corollary 2. *Let $X \in \mathbb{C}^{n\times n}$ and $S \in \mathbb{C}^{n\times m}$ with $S^\dagger S = I_m$ as in Lemma 2. Then*

$$\mu_i(X) \geq \mu_i(S^\dagger X S). \tag{3.13}$$

In the following, we will only use this corollary with $i = 1$, that is,

$$\mu_{\max}(X) \geq \mu_{\max}(S^\dagger X S). \tag{3.14}$$

First new bound for $\chi_q(G)$

Theorem 2 (First bound on quantum chromatic number)**.** *Let $\chi_q(G)$ be the quantum chromatic number of a connected graph G with adjacency matrix A. Let κ be the smallest integer such that*

$$0 \geq \mu_{\max}(A) + \sum_{i=1}^{\kappa} \mu_i^{\uparrow}(A) \tag{3.15}$$

holds. Then the quantum chromatic number is bounded from below by

$$\chi_q(G) \geq 1 + \kappa. \tag{3.16}$$

Proof. Let $\{P_{v,k} : v \in V, k \in [c]\}$ be any quantum c-coloring in dimension d. Construct the corresponding collection $\{P_k : k \in [c]\}$ of block-diagonal projectors as in Theorem 1.

Let $z \in \mathbb{C}^n$ denote the unique eigenvector of A corresponding to the largest eigenvalue $\mu_{\max}(A)$. Let $f_j \in \mathbb{C}^d$ for $j \in [d]$ denote the standard basis vectors. Let $s_1, \ldots, s_m$ be an orthonormal basis of the subspace

$$\mathcal{S} = \operatorname{span}\{P_k(z \otimes f_j) : k \in [c], j \in [d]\}. \tag{3.17}$$

Its dimension m satisfies

$$d \leq m \leq cd. \tag{3.18}$$

For the lower bound, observe that the d orthogonal vectors $z \otimes f_j$ are contained in $\mathcal{S}$ since

$$z \otimes f_j = (I_n \otimes I_d)\Big(z \otimes f_j\Big) = \big(\sum_{k\in[c]} P_k\big)(z \otimes f_j) = \sum_{k\in[c]} P_k(z \otimes f_j). \tag{3.19}$$

For the upper bound, observe that there are exactly cd vectors in eq. (3.17) and, thus, m cannot be larger than cd.

Let $S \in \mathbb{C}^{nd\times m}$ be the matrix with $s_1, \ldots, s_m$ as column vectors. The following two arguments show that the largest eigenvalue of the matrix $S^\dagger(A \otimes I_d)S$ is equal to $\mu_{\max}(A)$ and its multiplicity is equal to d. First, there exist d orthogonal vectors $y_1, \ldots, y_d \in \mathbb{C}^m$ such that $Sy_j = z \otimes f_j$ since the latter vectors are contained in the subspace $\mathcal{S}$, or equivalently, the column space of S. We have

$$\begin{aligned} S^\dagger(A \otimes I_d)Sy_j &= S^\dagger(A \otimes I_d)(z \otimes f_j) && (3.20)\\ &= \mu_{\max}(A)S^\dagger(z \otimes f_j) && (3.21)\\ &= \mu_{\max}(A)S^\dagger Sy_j && (3.22)\\ &= \mu_{\max}(A)y_j. && (3.23) \end{aligned}$$

Second, using Corollary 2, the largest eigenvalue of $S^\dagger(A \otimes I_d)S$ cannot be greater than the largest eigenvalue of $A \otimes I_d$.

We can always choose the orthonormal basis vectors $s_1, \ldots, s_m$ such that for each $i \in [m]$

there exists a unique $k_i \in [c]$ with

$$P_{k_i} s_i = s_i \text{ and } P_k s_i = 0 \text{ for all } k \neq k_i. \tag{3.24}$$

This is because $\mathcal{S} = \bigoplus_{k\in[c]} \mathcal{S}_k$, where $\mathcal{S}_k = \text{span}\{P_k(z \otimes f_j) : j \in [d]\}$ since the projectors P_k form a resolution of the identity.

We now see that the diagonal entries of the matrix $S^\dagger(A \otimes I)S$ must all be zero since

$$(S^\dagger(A \otimes I)S)_{ii} = s_i^\dagger (A \otimes I_d) s_i = s_i^\dagger P_{k_i}(A \otimes I_d) P_{k_i} s_i = 0. \tag{3.25}$$

For the last equality we used that $P_k(A \otimes I_d)P_k = 0$ for all $k \in [c]$.

So using Lemma 2, we obtain

$$\begin{aligned}
0 &= \text{tr}(S^\dagger(A \otimes I)S) \\
&= \sum_{i=1}^{m} \mu_i^\uparrow (S^\dagger(A \otimes I)S) \\
&= \sum_{i=1}^{m-d} \mu_i^\uparrow (S^\dagger(A \otimes I)S) + d \cdot \mu_{\max}(A) \\
&\geq \sum_{i=1}^{m-d} \mu_i^\uparrow (A \otimes I) + d \cdot \mu_{\max}(A).
\end{aligned}$$

Now let κ_d be the smallest integer such that

$$0 \geq \sum_{i=1}^{\kappa_d} \mu_i^\uparrow (A \otimes I) + d \cdot \mu_{\max}(A). \tag{3.26}$$

Using (3.18), we have

$$(c-1)d = cd - d \geq m - d \geq \kappa_d. \tag{3.27}$$

Note that $\kappa_d \geq (\kappa - 1)d + 1$ must hold because otherwise the condition that $\kappa = \kappa_1$ is minimal would be violated. This implies $c - 1 \geq \lceil \kappa_d/d \rceil = \lceil \kappa - 1 + 1/d \rceil = \kappa$. In particular, this hold for a quantum c-coloring attaining $\chi_q(G)$ so that $\chi_q(G) \geq 1 + \kappa$. □

A weaker version of Theorem 2, with $\chi(G)$ replacing $\chi_q(G)$, was proved by Hoffman [50] in 1970. This theorem immediately implies that

$$1 + \frac{\mu_1}{|\mu_n|} \leq \chi_q(G) \leq \chi(G), \tag{3.28}$$

which was proved in [34] using different techniques.

The proof of the following bound generalises a proof due to Haemers ([45], [46]) from the classical to the quantum chromatic number.

Second new bound for $\chi_q(G)$

Theorem 3 (Second bound on quantum chromatic number)**.** *For any connected graph G with $\mu_2 > 0$:*

$$\chi_q(G) \geq 1 + \min\left\{ g, \frac{|\mu_n(A)|}{\mu_2(A)} \right\}, \tag{3.29}$$

where g is the multiplicity of $\mu_n(A) = \mu_{\min}(A)$.

Proof. Consider an arbitrary quantum c-coloring in dimension d. Assume that $c \leq g$.

Let $\mathcal{S}$ be defined as in the proof of the previous theorem in (3.17). Let $\mathcal{T}$ be the subspace spanned by the eigenvectors corresponding to the cd smallest eigenvalues $\mu_1^{\uparrow}(A \otimes$

$I), \ldots, \mu^{\uparrow}_{cd}(A \otimes I)$. We now show that there exists a non-zero unit vector y with

$$y \in \mathcal{S}^{\perp} \cap \mathcal{T}. \tag{3.30}$$

To this end, define $\mathcal{R} = \operatorname{span}\{z \otimes f_j : j \in [d]\}$. Observe that both $\mathcal{S}^{\perp}$ and $\mathcal{T}$ are contained in the subspace $\mathcal{R}^{\perp}$ and

$$\dim \mathcal{R}^{\perp} = nd - d < nd = (nd - cd) + cd \leq \dim \mathcal{T} + \dim \mathcal{S}^{\perp}. \tag{3.31}$$

For $i \in [c]$, define $y_i = P_i y$. Let m be the number of y_i that are non-zero. We now show that at least two of them (w.l.o.g. y_1 and y_2) must be non-zero, that is, $m \geq 2$. First of all, at least one must be non-zero because otherwise we would have $0 \neq y = (I_n \otimes I_d)y = \big(\sum_{k\in[c]} P_k\big)y = \sum_{k\in[c]} y_k = 0$. Now assume that only y_1 were non-zero, or equivalently, $y = P_1 y$. But this leads to the contradiction

$$0 > \mu^{\uparrow}_c(A) = \mu^{\uparrow}_{cd}(A \otimes I_d) \geq y^{\dagger}(A \otimes I_d)y = y^{\dagger}P_1(A \otimes I_d)P_1 y = 0, \tag{3.32}$$

where the first inequality holds because $c \leq g$, the second inequality holds because $y \in \mathcal{T}$, and the last equality holds because $P_1(A \otimes I_d)P_1 = 0$. The latter follows from the fact that P_1 is one of the projectors forming a pinching that annihilates $A \otimes I_d$.

Define the orthonormal vectors $s_i = y_i/\|y_i\|$ for $i \in [m]$ and S to be the matrix whose columns are s_i. Define the matrix $X = A \otimes I_d - \Delta \cdot zz^{\dagger} \otimes I_d$, where $\Delta = \mu_{\max} - \mu_{\min}$ and z is the (unit) eigenvector corresponding to $\mu_{\max}$.

Since y is in the column space of S, the smallest eigenvalue of $S^{\dagger}XS$ is at most $y^{\dagger}Xy$, which in turn is at most $\mu^{\uparrow}_{cd}(A \otimes I_d) = \mu^{\uparrow}_c(A) = \mu_n(A)$ as $y \in \mathcal{T}$ and $c \leq g$. Also, it holds that $\mu_{\max}(S^{\dagger}XS) \leq \mu_{\max}(X) = \mu_2(A)$.

We now show that the trace of $S^\dagger XS$ is equal to 0. The diagonal entries of $S^\dagger XS$ are all zero because

$$(S^\dagger XS)_{ii} = s_i^\dagger X s_i \propto y_i^\dagger X y_i \tag{3.33}$$

$$= y^\dagger P_i \big(A \otimes I_d - \Delta \cdot zz^\dagger \otimes I_d\big) P_i y \tag{3.34}$$

$$= y^\dagger P_i (A \otimes I_d) P_i y - \Delta \cdot \sum_{j\in[d]} y^\dagger P_i (zz^\dagger \otimes f_j f_j^\dagger) P_i y \tag{3.35}$$

$$= 0 \tag{3.36}$$

where we used that $P_i(A \otimes I_d)P_i = 0$ and $y \perp P_i(z \otimes f_j)$ for each j. The latter holds as $y \in \mathcal{S}^\perp$ and $P_i(z \otimes f_j) \in \mathcal{S}$.

Combining that $S^\dagger XS$ is traceless with the above bounds on its minimum and maximum eigenvalues yields that

$$0 = \operatorname{tr}(S^\dagger XS) = \sum_{i\in[m]} \mu_i^\uparrow (S^\dagger XS) \leq \mu_n(A) + (m-1)\mu_2(A), \tag{3.37}$$

which completes the proof. □

A weaker version of this bound, with $\chi(G)$ replacing $\chi_q(G)$, is already known, for example in Corollary 3.6.4 in [16].

We note that both Theorems are also valid for weighted adjacency matrices of the form $W \circ A$, where W is an arbitrary Hermitian matrix and $\circ$ denotes the Hadamard product (also called the Schur product). An example of using a weighted adjacency matrix is to replace A with the normalized adjacency matrix $\mathcal{A} = D^{-1/2}AD^{-1/2}$, where D is the diagonal matrix of vertex degrees, in both bounds for $\chi_q(G)$. This choice of weight matrix reproduces the lower bound for $\chi(G)$ in Theorem 2.2 in Coutinho et al. [27], once account is taken of

the differences in notation.

Implications for quantum chromatic number

Strongly regular graphs (SRGs)

Elphick and Wocjan [34] discussed implications of their results for the quantum chromatic number. For example, they demonstrated using an inertial bound that the Clebsch graph (seen in Figure 3.1) has $\chi_q(G) = 4$. Since the Clebsch graph has spectrum $(5^1, 1^{10}, -3^5)$ this also follows immediately from Theorem 3. The generalised quadrangle, $GQ(2, 4)$, on 27 vertices has $\chi = 6$, but in [34] the authors were only able to show that $\chi_q \geq 5$. The spectrum of $GQ(2, 4)$ is $(10^1, 1^{20}, -5^6)$, so from Theorem 3 it follows that $\chi_q = 6$.

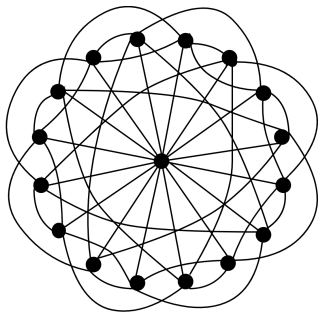

Figure 3.1: A Clebsch Graph (16,5,0,2)

Both of these graphs are strongly regular, and Theorems 2 and 3 can be used to calculate the quantum chromatic number of many strongly regular graphs (SRGs). For example the Kneser graph $K_{p,2}$ (with $p \geq 4$) has $\chi = p - 2$ and spectrum $((p-2)(p-3)/2^1, 1^{p(p-3)/2}, (3-p)^{p-1})$, which using Theorem 3 implies $\chi_q = \chi = p - 2$. The Hoffman-Singleton graph, SRG(50, 7, 0, 1) has spectrum$(7^1, 2^{28}, -3^{21})$ and $\chi = 4$. Theorem 2 implies $\chi_q = 4$ also.

Fiala and Haemers [36] identified (see their Theorem 10.1) all SRGs with $\chi = 5$. So, using Theorem 2, SRG$(15, 8, 4, 4)$ and SRG$(25, 8, 3, 2)$ have $\chi_q = 5$; and using Theorem 3, SRG$(21, 10, 3, 6)$ and SRG$(25, 16, 9, 12)$ have $\chi_q = 5$. The Gewirtz graph, SRG$(56, 10, 02)$, has $\chi = 4$ and spectrum $(10^1, 2^{35}, -4^{20})$; so using Theorem 2 it has $\chi_q = 4$.

The Higman-Sims graph is SRG$(100, 22, 0, 6)$. Its spectrum is equal to $(22^1, 2^{77}, -8^{22})$ and it has $\chi = 6$ (see [36]). Theorem 2 implies $\chi_q \geq 4$ and Theorem 3 implies $\chi_q \geq 5$. We do not however know whether $\chi_q = 5$ or 6. Similarly the M_{22} graph is SRG$(77, 16, 04)$ with spectrum $(16^1, 2^{55}, -6^{21})$ has $\chi = 5$ (see [36]). Theorems 2 and 3 imply $\chi_q \geq 4$, but we do now know whether $\chi_q = 4$ or 5.

Non-SRGs

The orthogonality graph, $\Omega(n)$, has vertex set the set of $\pm 1-$vectors of length n, with two vertices adjacent if they are orthogonal. With $4|n$ (see [62]), it is known that $\chi_q(\Omega(n)) = n$ but $\chi(\Omega(n))$ is exponential in n. A proof that $\chi_q(\Omega(n)) = n$ is as follows. It is immediate from the definition of $\Omega(n)$ that $\xi'(\Omega(n)) = n$, and it is known that $\chi_q(G) \leq \xi'(G)$, where $\xi'(G)$ is the normalized orthogonal rank of G [77]. However, using Theorem 2 and results in section 4.3 of [43] on the eigenvalues of orthogonality graphs we have that:

$$\chi_q(\Omega(n)) \geq 1 + \frac{\mu_1}{|\mu_n|} = 1 + \frac{1 \cdot 3 \cdots (n-3) \cdot (n-1)}{1 \cdot 3 \cdots (n-3)} = n.$$

Theorem 4. *The orthogonality graph $\Omega(n)$ has a quantum coloring in dimensions pn, where p is a positive integer.*

Proof. We can construct a quantum coloring of $\Omega(n)$ using n colors as follows. Let dimension

$d = n$, let $U = \text{diag}(1, \omega, \ldots, \omega^{n-1})$ be a unitary matrix where $\omega = e^{2\pi i/n}$, and let z_v denote the ± 1 vector of length n assigned to vertex v. Then let

$$P_{v,k} = U^k z_v z_v^\dagger (U^\dagger)^k : v \in V, k \in [c].$$

It is straightforward that this collection of orthogonal projectors satisfy the completeness and orthogonality conditions in Definition 21, so this completes the quantum coloring with $d = n$.

For $d = pn$, where $p > 1$, let

$$\widetilde{P}_{v,k} = P_{v,k} \otimes I_p : v \in V, k \in [c].$$

This new collection of orthogonal projectors also satisfy the completeness and orthogonality conditions in Definition 21. □

We note in passing that with $4|n$, a proof that $\omega(\Omega(n)) = n$, would provide a proof of the Hadamard Conjecture, which dates from 1867.

Vertex Transitive$(12, 27)$ and Vertex Transitive$(12, 54)$ are examples of non-SRGs for which Theorem 3 is exact with $\chi_q = 4$. Barbell graphs and irregular complete $q-$partite graphs have $\chi_q = \chi$, using Theorem 2.

Hoffman colorings

Any graph for which

$$1 + \frac{\mu_1}{|\mu_n|} = \chi(G),$$

is said to have a Hoffman coloring. All such graphs therefore have $\chi_q(G) = \chi(G)$. Examples include SRG$(49, 12, 5, 2)$ which has $\chi_q = 7$ and the Schlafli graph SRG$(27, 16, 10, 8)$ (seen in Figure 3.2) which has $\chi_q = 9$. Haemers and Touchev investigated graphs with Hoffman colorings, and Table 1 in [47] lists many such SRGs with up to 100 vertices.

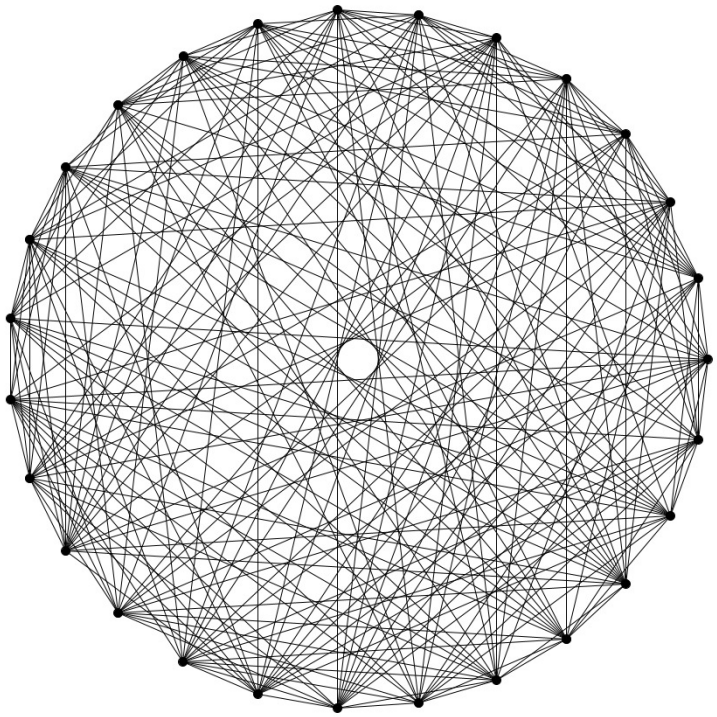

Figure 3.2: A Schlafi Graph (27,16,10,8)

CHAPTER 4: QUANTUM ALGORITHMS FOR PATTERN MATCHING ON LABELED GRAPHS

Lower Bounds

Our main subject for this section will be the proof of Theorem 5, stated below.

Theorem 5. *There exists a reduction from LCS with strings X and Y over alphabet Σ to PMLG with substitutions over a binary alphabet. This requires $O((|X| + |Y|) \log(|X| + |Y|) \cdot \log^2 |\Sigma|)$ time (on a classical computer) and outputs a graph $G = (V, E)$, where $|E| = O(|X| \log(|X|+|Y|) \cdot \log^2 |\Sigma|)$ and pattern $P[1, m]$, where $m = O(|Y| \log(|X|+|Y|) \cdot \log^2 |\Sigma|)$.*

Theorem 5 leads immediately to Corollary 3.

Corollary 3. *An algorithm for PMLG with substitutions to the pattern over a binary alphabet running in quantum time $\tilde{O}(|E|^{1-\varepsilon}m + |E|m^{1-\varepsilon})$ for any constant $\varepsilon > 0$ would provide an algorithm running in quantum time $\tilde{O}(|X||Y|^{1-\varepsilon} + |X|^{1-\varepsilon}|Y|)$ for LCS.*

It should be noted that no strongly sub-quadratic quantum algorithms for LCS are known.

Theorem 5, in addition to Lemma 1, leads immediately to Corollary 4.

Corollary 4. *Under NC-QSETH, PMLG cannot be solved in quantum time that is $O(|E|^{\frac{1}{2}-\varepsilon}m+ |E|m^{\frac{1}{2}-\varepsilon})$ for any constant $\varepsilon > 0$.*

Main Reduction to PMLG-sub over Large Alphabet

Our hardness result will be proven through a reduction from the decision version of LCS to PMLG. We first present a simplified version of the reduction for larger edge label alphabets

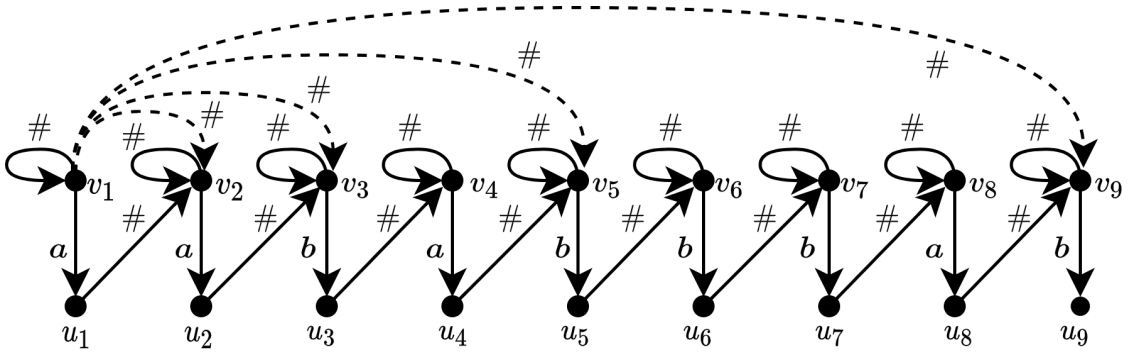

Figure 4.1: Reduction from LCS to PMLG for $X = aababbbab$. Note that dashed edges are only shown from v_1 but similar edges are present from every v_i, $1 \leq i \leq n$. If $Y = baabbabaa$ then $P = \#^4b\#^4a\#^4a\#^4b\#^4b\#^4a\#^4b\#^4a\#^4a$.

and then show how to modify it to obtain the result for binary alphabets. In the decision version of LCS we are given as input two sequences X and Y and an integer k and have to decide whether there exists a common subsequence to X and Y of length k. Suppose $|Y| \geq |X|$ and let $n = |Y|$.

We will first construct our graph G based on the string X. We start by making two sets of vertices u_1, u_2, ..., u_n and v_1, v_2, ..., v_n. We add directed edges (v_i, u_i) with labels $X[i]$ for $1 \leq i \leq |X|$. All remaining edges are labeled with a new symbol # that is not found in either X or Y. We then create edges (u_i, v_{i+1}) for $1 \leq i \leq n-1$. Next, for v_i, $1 \leq i \leq n$ we create edges (v_i, v_i), (v_i, v_{i+1}), (v_i, v_{i+2}), (v_i, v_{i+4}), ..., (v_i, v_{i+2^c}) for the largest c such that $i + 2^c \leq n$ and the edge $(u_{|X|}, u_{|X|})$.

Next, we create the pattern P based on the string Y. Let

$$P = \#^{\lceil \log n \rceil + 1}\, Y[1]\, \#^{\lceil \log n \rceil + 1}\, Y[2]\, \#^{\lceil \log n \rceil + 1}\, \ldots\, \#^{\lceil \log n \rceil + 1}\, Y[n].$$

and let $\delta = n - k$. This construction is shown in Figure 4.1.

The following lemma will help to establish the correctness of the above reduction.

Lemma 3. *The shortest path between any v_i and v_j for any $j > i$ has at most $\lceil \log n \rceil$ edges.*

Proof. Let i' be the largest value such that $i \leq i' \leq j$ and there exists edge $(v_i, v_{i'}) \in E$. By construction $i' = i + 2^x$ for some $x \geq 0$. We claim $i' > \frac{j-i}{2} + i$. Otherwise

$$i' = i + 2^x \leq \frac{j-i}{2} + i$$

implies $i + 2^{x+1} \leq j$,contradicting that i' index was the largest possible. Since the distance between the current index and j can always be at least halved, by repeatedly apply the same process, we need at most $\lceil \log n \rceil$ additional edges before reaching j. □

The correctness of the reduction can now be established by the following lemma.

Lemma 4. *There exists an LCS of length at least k for strings X and Y iff and there exists a walk in G that matches P after at most δ substitutions to P.*

Proof. First assume there exists an LCS of length $k' \geq k$, with $X[i_1]$, $X[i_2]$,..., $X[i_{k'}]$ matching $Y[j_1]$, $Y[j_2]$,..., $Y[j_{k'}]$. We obtain a walk on G as follows: starting at vertex v_{i_1}, we traverse the self-loop (v_{i_1}, v_{i_1}) until we reach the $Y[j_1]$ in P, substituting symbols in P to # as necessary. Then we follow edge (v_{i_1}, u_{i_1}) matching $Y[j_1]$ in P. We now traverse the edge (u_i, v_{i+1}) and the shortest path from v_{i+1} to v_{i_2}, which by Lemma 3 has at most $\lceil \log n \rceil$ edges. We next traverse the self-loop (v_{i_2}, v_{i_2}) until reaching the symbol $Y[j_2]$ in P, at which point we match $Y[j_2]$ with the edge (v_{j_2}, u_{j_2}). This process is repeated until $Y[n]$ is matched. Exactly $n - k' \leq n - k = \delta$ symbols in P are substituted to #.

Next suppose there exists a walk in G that matches P with $\delta' \leq \delta$ substitutions. This implies that $n - \delta'$ of the non-#-symbols in P are not substituted and instead matched with

symbols on edges (v_i, u_i). By construction, once the edge (v_i, u_i) is traversed, the next edge with a non-#-label traversed is an edge $(v_{i'}, u_{i'})$ where $i' > i$. Hence, the non-# symbols in P matched with edges in G correspond to a common subsequence of X and Y of length $n - \delta' \geq n - \delta = k$. □

Hardness of PMLG-edit over Large Alphabet

Lemma 5. *For an instance of PMLG constructed as above, there exists an optimal solution where only substitutions are made in P.*

Proof. Any substring of P consisting of only # can be matched without edit cost from any vertices v_i, $1 \leq i \leq |X|$. From the vertices u_i, $1 \leq i \leq |X|$, an edge with # needs to be traversed regardless so it would be suboptimal to delete any # in P that could be matched on an edge (u_i, v_{i+1}). Combining these, an optimal solution never deletes a substring of P of the form $\#^x$, $x \geq 1$. This leaves only substrings that contain some symbol $Y[i]$. However, the cost for deleting any such substring is at least the cost of substituting $Y[i]$ to a #-symbol. We conclude that no deletions need to be made to P in an optimal solution.

For insertions, a similar argument holds. Any insertion of a substring of the form $\#^x$, $x \geq 1$, is clearly suboptimal since there exist enough #-symbols to traverse from any two vertices v_i and v_j. An insertion that includes a non-#-symbol is also unnecessary, since the edge matched against that symbol could have been not traversed for the same cost. □

Due to Lemma 5, Lemma 4 continues to hold when deletions and insertions are also allowed to P.

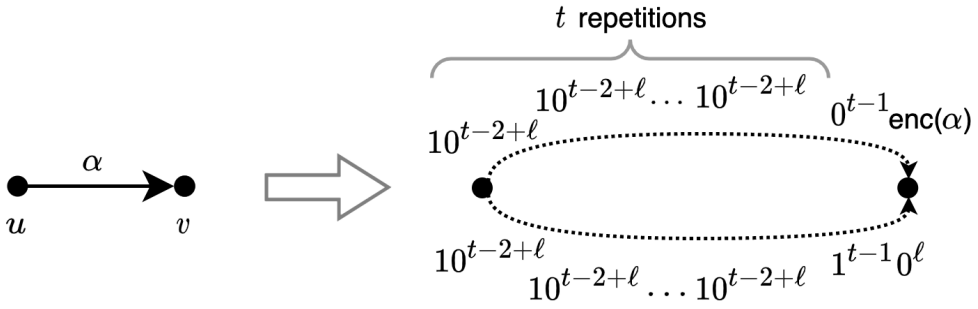

Figure 4.2: The conversion of an edge in G with label α from u to v to two paths between u and v in G'.

Hardness of PMLG-sub over Binary Alphabet

Some additional work must be done to convert the above reduction to work for a binary alphabet in G and P.

Let $\Sigma' = \Sigma \cup \{\#\}$, $\sigma = |\Sigma'| \geq 3$, and $\ell = 2\lceil \log \sigma \rceil$. We will create our own constant weight binary code (i.e., one where all codewords have the same number of 1's) for Σ'. We first take $t = \lceil \log \sigma \rceil$. This makes $\binom{\ell}{t} \geq \sigma$ and allows us to assign to every symbol in Σ' a distinct binary string of length ℓ containing exactly t 1's. Let $\text{enc}(\alpha)$ denote this encoding for $\alpha \in \Sigma'$. Controlling the number of 1's allows us to compute the cost of an optimal solution, as described next. We modify the earlier reduction by replacing every edge (u, v) (allowing for $v = u$) having label $\alpha \in \Sigma'$ with:

- A directed path from u to v that matches $(10^{t-2+\ell})^t 0^{t-1}\,\text{enc}(\alpha)$. These paths are called *symbol paths*;
- A parallel directed path starting and ending at the same vertices (or vertex) that matches the string $(10^{t-2+\ell})^t 1^{t-1} 0^{\ell}$. These are called *escape paths*.

This is illustrated in Figure 4.2. We denote the resulting graph as G'. The pattern P' is created by replacing every symbol $P[i]$ with $(10^{t-2+\ell})^t 1^{t-1}\,\mathrm{enc}(P[i])$ for $1 \leq i \leq n$. Let

$$\begin{aligned}\delta' &= t(n-k) + (t-1)(|P| - (n-k)) \\ &= t(n-k) + (t-1)(n(\lceil \log n \rceil + 2) - (n-k)).\end{aligned}$$

Note that $|V'|$, $|E'|$, and $|P'|$ are $O(n \log n \cdot \log^2 \sigma)$ where $n = |Y| \geq |X|$ and σ is the size of the original alphabet of X and Y plus 1. For the remainder of this section, the definitions of G, P, and δ are fixed from the previous section and the definitions of G', P', and δ' remain fixed to the ones just given.

The next lemma will simplify future analysis by establishing constraints on any optimal walk for P' in G'.

Lemma 6. *Any walk in G' matching P' with at most δ' mismatches must start at some vertex corresponding to an original vertex in V.*

Proof. First, consider a walk starting at some vertex w internal to a subdivided path (i.e., one not corresponding to an original vertex in G) and where w does not have an edge with 1 leaving it. By construction, this creates at least t mismatches for every complete subdivided path traversed. To see this, for each substring $(10^{t-2+\ell})^t 1^{t-1}\,\mathrm{enc}(P[i])$ in at most one of the 1's in the prefix $(10^{t-2+\ell})^t$ can be matched to an edge and at least one of the 0's is mismatched to a 1 edge as well.

If the walk starts at a non-original vertex with an edge leaving it labeled 1, but not a edge internal to subpath labeled 1^t in any escape path. Then for each substring $(10^{t-2+\ell})^t 1^{t-1}\,\mathrm{enc}(P[i])$ in P', the substring $\mathrm{enc}(\alpha)$ is forced to traverse a subpath labeled 0^t, causing at least t mismatches once again. Finally, suppose the walk starts at a non-original vertex with an edge

leaving it labeled 1, but an internal to subpath labeled 1^t in some escape path. Then for each substring $(10^{t-2+\ell})^t 1^{t-1}\,\mathrm{enc}(P[i])$ in P', the prefix $(10^{t-2+\ell})^t$ once again again causes t mismatches.

In all cases, at least $t|P| > \delta'$ mismatches are causes in total. □

The following lemma establishes the correctness of the reduction in the binary case.

Lemma 7. *There exists an LCS of length at least k for strings X and Y iff and there exists a walk in G' that matches P' after at most δ' substitutions to P'.*

Proof. First, suppose there exists an LCS of length at least k. Follow the walk in G' corresponding to the walk in G that requires at most δ substitutions to P. When doing so, take the symbol path when the symbol in P matched the corresponding edge in G, and the escape path otherwise. This incurs $t-1$ mismatches per subdivided edge corresponding to a match and t mismatches per subdivided edge corresponding to a mismatch. Hence the total number of mismatches is at most $t(n-k) + (t-1)(|P| - (n-k)) = \delta'$.

Next, suppose there exists a walk in G' matching P' with at most δ' mismatches. By Lemma 6, substrings of the form $(10^{t-2+\ell})^t 1^{t-1}\,\mathrm{enc}(P[i])$ are matched (after substitutions) to subpaths in the walk that start at the beginning of symbol or escape paths. If $\mathrm{enc}(P[i]) \neq \mathrm{enc}(\alpha)$, then the number of mismatches for that substring is t since the number of mismatches for matching the escape path is t (Hamming distance of $1^{t-1}\,\mathrm{enc}(P[i])$ and $1^{t-1}0^\ell$), versus the symbol path, which is at least t (Hamming distance of $1^{t-1}\,\mathrm{enc}(P[i])$ and $0^{t-1}\,\mathrm{enc}(\alpha)$). If $\mathrm{enc}(P[i]) = \mathrm{enc}(\alpha)$, by matching the symbol path, the number of mismatches is $t-1$ (Hamming distance of $1^{t-1}\,\mathrm{enc}(P[i])$ and $0^{t-1}\,\mathrm{enc}(\alpha)$). We conclude that in an optimal solution the total number of mismatches is t times the number of mismatched symbols between P and the corresponding walk in G, plus $t-1$ times the number of matched symbols

between P and the corresponding walk in G. Hence, if the number mismatches for G' is at most $\delta' = t(n-k) + (t-1)(|P| - (n-k))$, the number mismatched symbols in G is at most $n - k = \delta$. By Lemma 4, this implies the LCS of X and Y is at least k. □

This completes the proof of Theorem 5.

Quantum Algorithm for PMLG

The following section describes an algorithm that can be easily adapted for PMLG-exact, PMLG-sub, and PMLG-edit. We will first describe the algorithm for PMLG-edit and then the modifications necessary for PMLG-exact and PMLG-sub.

Durr et al.'s shortest path algorithm assumes that the graph is represented by oracles such that for a vertex $v \in V$ it can return the vertices adjacent to v along with an edge weight [30]. In particular, if we assume some arbitrary ordering on the vertices in V, given that the outdegree v_i is $deg^+(v_i)$, the oracles $f_i : [deg^+(v_i)] \mapsto \{1, ..., |V|\} \times \mathbb{N}$ have

$$f_i(j) = (\text{the } j^{th} \text{ vertex adjacent to vertex } v_i, \text{ the weight on the corresponding edge}).$$

Under this representation of the input graph, Durr et al. provide an algorithm that finds the shortest path from a vertex $s \in V$ to a vertex $t \in V$, uses $O(\sqrt{|V||E|}\log^2|V|)$ oracle queries, and runs in quantum time within logarithmic factors of the query complexity.

We assume that the graph $G = (V, E)$ is given as input for PMLG with $|E| = \Omega(|V|)$. Otherwise, the graph is not connected and we can solve the problem on different (weakly connected) components separately. We assign an arbitrary ordering to V. For the PMLG-edit, Amir et. al. [8] provide a construction of a graph G' based on G and length m pattern

P with special vertices s and t such that there exists a path from s to t whose edge weight sum to at most δ iff there exist a walk in G matching P after at most δ modification to P. Their construction results in a graph that has $\Theta(|V|m)$ vertices and $\Theta(|E|m)$ edges and is based around duplicating the graph m times with modifications made based on the symbols in P. We call this graph G' the *alignment graph*. Let (v_i, v_j, w) denote the edge from v_i to v_j with weight w. Formally, the alignment graph consists of vertices:

$$V' = \{v_i^j \mid 1 \leq i \leq |V|, 1 \leq j \leq m+1\} \cup \{s, t\}.$$

and edges

$$\begin{aligned}
E' = &\{(s, v_i^1, 0) \mid 1 \leq i \leq |V|\} && (4.1)\\
&\cup \{(v_i^j, v_{i'}^{j+1}, 0) \mid 1 \leq j \leq m, (v_i, v_{i'}) \in E, \mathit{label}((v_i, v_{i'})) = P[j]\} && (4.2)\\
&\cup \{(v_i^j, v_{i'}^{j+1}, 1) \mid 1 \leq j \leq m, (v_i, v_{i'}) \in E, \mathit{label}((v_i, v_{i'})) \neq P[j]\} && (4.3)\\
&\cup \{(v_i^j, v_i^{j+1}, 1) \mid 1 \leq j \leq m, 1 \leq i \leq |V|\} && (4.4)\\
&\cup \{(v_i^j, v_{i'}^j, 1) \mid 1 \leq j \leq m, (v_i, v_{i'}) \in E\} && (4.5)\\
&\cup \{(v_i^{m+1}, t, 0) \mid 1 \leq i \leq |V|\} && (4.6)
\end{aligned}$$

The interpretation of the graph is as follows:

- Edges (4.2) correspond to $P[j]$ being matched to edge $(v_i, v_{i'})$ and have weight 0.
- Edges (4.3) correspond to $P[j]$ being substituted to match edge $(v_i, v_{i'})$.
- Edges (4.4) correspond to the symbol $P[j]$ being deleted. In this case the current vertex, v_i, the walk is on is unaltered.
- Edges (4.5) correspond to the symbol $\mathit{label}(v_i, v_{i'})$ being inserted in P between the

$(j-1)^{th}$ symbol and the j^{th} symbol. If $j-1=0$, this means inserting prior to $P[1]$.

An example of an alignment graph is provided in Figure 4.3.

Lemma 8 ([8])**.** *There exists an st-path in the alignment graph G' with total weight δ iff there exists a walk in G that P matches after δ edits.*

The key insight into efficiently using Durr et al.'s shortest path algorithm on directed graphs is that the alignment graph does not need to be constructed in full in order to simulate the input oracles used by the algorithm. If we did have to construct the alignment graph, the $\Theta(|E|m)$ edges needed would result in no speed up over the classical case. Instead, only the vertices in the alignment graph are constructed, the values output by the adjacency oracles are computed by the new oracles we provide in constant time, given the original input oracles for the starting graph G and pattern P. In what follows, we describe the construction of the vertices and these adjacency oracles.

Construction

Let $G=(V,E)$ be the original graph given where $V=\{v_1,...,v_{|V|}\}$. Our algorithm allows for insertions and deletions in addition to substitutions. Assume we have substitution cost S, deletion cost D, and insertion cost I. The alignment graph $G'=(V',E')$ is constructed as follows: Recall that the vertex set is $V'=\{v_i^j \mid 1\leq i\leq |V|, 1\leq j\leq m+1\}\cup\{s,t\}$. The linearized index for s is 0, for t it is $(m+1)|V|+1$, and for v_i^j, $1\leq i\leq |V|$, $1\leq j\leq m+1$, it is $(j-1)|V|+i$.

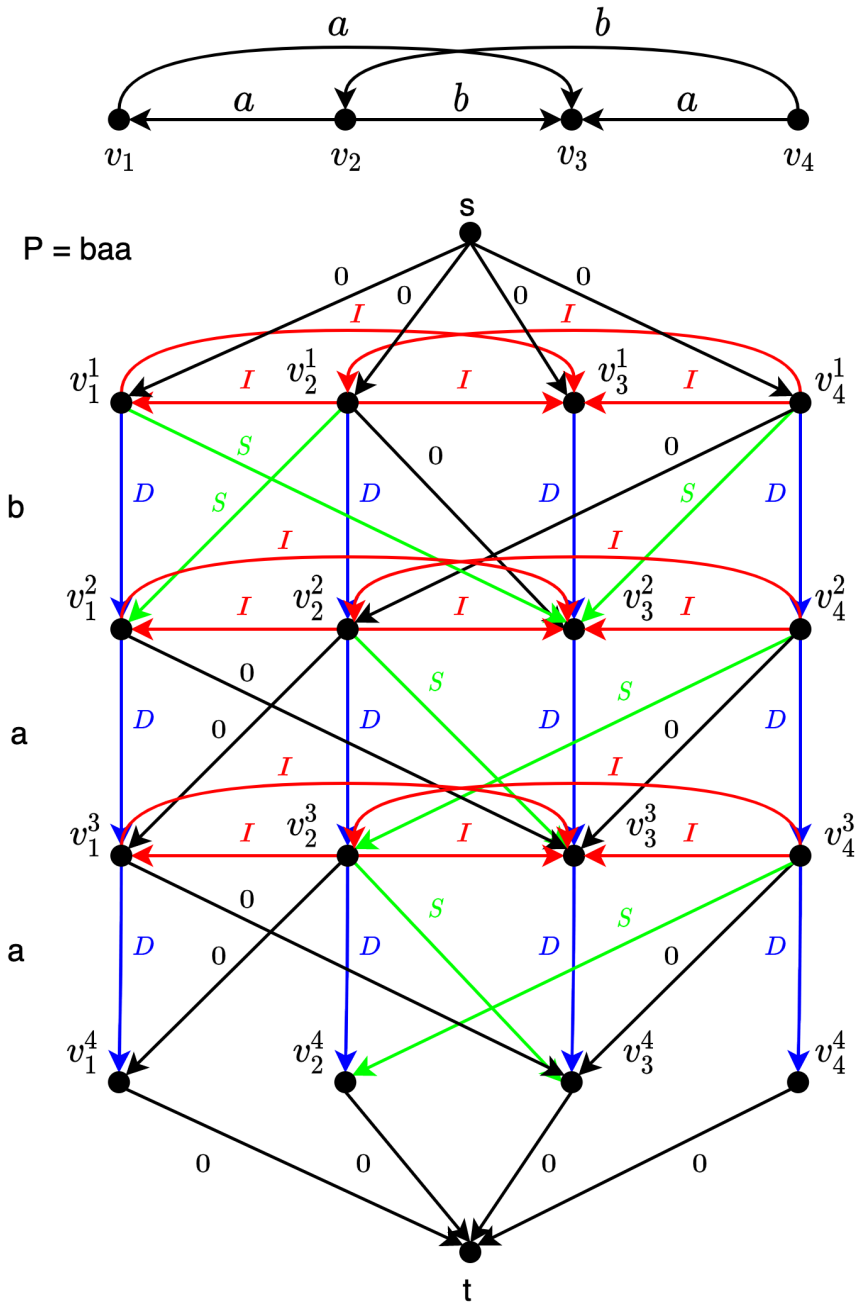

Figure 4.3: The alignment graph G' for PMLG-edit constructed from the starting graph G shown above it. The edges with label I correspond to insertion and have weight 1, edges with label S correspond to substitution and have weight 1, edges with label D correspond to deletion and have weight 1, black edge correspond to exact match and have weight 0.

Adjacency Oracles

We next define a set of adjacency oracles for G'.

The first gives adjacency's of a given vertex $v_i^j \in V'$, $1 \leq i \leq |V|$, $1 \leq j \leq m$. These will all be of the form $f_i^j : [deg^+(v_i^j)] \mapsto [|V'|] \times \mathbb{N}$. Recall that we have adjacency oracle $f_i : [deg^+(v_i)] \mapsto [|V|]$ for $v_i \in V$. The out-degree degree for a vertex v_i^j, is $deg^+(v_i^j) = 2deg^+(v_i) + 1$, we define

$$f_i^j(k) = \begin{cases} ((j-1)|V| + f_i(k), I) & 1 \leq k \leq deg^+(v_i) \\ (j|V| + f_i(k - deg^+(v_i)), 0) & deg^+(v_i) + 1 \leq k \leq 2d^+(v_i) \text{ and} \\ & label((v_i, v_{f_i(k-deg^+(v_i))})) = P[j] \\ (j|V| + f_i(k - deg^+(v_i)), S) & deg^+(v_i) + 1 \leq k \leq 2deg^+(v_i) \text{ and} \\ & label((v_i, v_{f_i(k-deg^+(v_i))})) \neq P[j] \\ (j|V| + i, D) & k = 2deg^+(v_i) + 1 \end{cases}$$

For $1 \leq i \leq |V|$, $j = m + 1$ is $deg^+(v_i^j) = 1$, and $f_i^j(1) = ((m+1)|V| + 1, 0)$. Finally, the out-degree of s is $deg^+(s) = |V|$ and for $1 \leq k \leq |V|$, $f_s(k) = (k, 0)$, and the out-degree of t is 0.

Applying the quantum shortest path algorithm of Durr et al [30] while utilizing the oracles above yields an algorithm running in quantum time $\tilde{O}(\sqrt{|V'||E'|})$. Using that $|V'| = m|V| + 2$ and $|E'| = \Theta(m|E|)$ this yields an algorithm with quantum query/time complexity $\tilde{O}(m\sqrt{|V||E|})$. In the case where $|E| = \Omega(|V|^{1+\varepsilon})$ for any $\varepsilon > 0$, this provides an improvement over the classical algorithm.

Modifications for PMLG-sub

To obtain an algorithm for PMLG-sub, we only want to allow for the edges representing exact matches and edges representing subsitutions in Figure 4.3. In terms of the oracles, for $1 \leq i \leq |V|$, $1 \leq j \leq m$, this makes the outdegree of vertex v_i^j equal to $deg^+(v_i^j) = deg^+(v_i)$ and

$$f_i^j(k) = \begin{cases} (j|V| + f_i(k), 0) & 1 \leq k \leq deg^+(v_i), \quad \ell((v_i, v_{f_i(k)})) = P[j] \\ (j|V| + f_i(k), S) & 1 \leq k \leq deg^+(v_i), \quad \ell((v_i, v_{f_i(k)})) \neq P[j] \end{cases}$$

As before, for $1 \leq i \leq |V|$, $j = m + 1$ is $deg^+(v_i^j) = 1$, and $f_i^j(1) = (m|V| + 1, 0)$. The out-degree of s is $deg^+(s) = |V|$ and for $1 \leq k \leq |V|$, $f_s(k) = (k, 0)$, and the out-degree of t is still 0.

Modifications for PMLG-exact

To obtain an algorithm for PMLG-exact, we only want to allow for the edges representing exact matches in Figure 4.3. In terms of the oracles, for $1 \leq i \leq |V|$, $1 \leq j \leq m$, this makes the outdegree of vertex v_i^j, $deg^+(v_i^j)$, equal to the number of edges leaving v_i with label $P[j]$, denoted $deg^+(v_i)|_{P[i]}$. Letting k_1, k_2, ..., $k_{deg^+(v_i)|_{P[i]}}$ denote the indices of vertices adjacent to v_i where $label((v_i, v_{k_h})) = P[j]$, $1 \leq h \leq deg^+(v_i)|_{P[i]}$, we have

$$f_i^j(k) = (j|V| + f_i(k_h), 0) \qquad 1 \leq h \leq deg^+(v_i)|_{P[i]}$$

For $1 \leq i \leq |V|$, $j = m + 1$ is $deg^+(v_i^j) = 1$, and $f_i^j(1) = (m|V| + 1, 0)$. The out-degree of s is $deg^+(s) = |V|$ and for $1 \leq k \leq |V|$, $f_s(k) = (k, 0)$, and the out-degree of t is 0.

CHAPTER 5: CONCLUSION

To conclude, we present several interesting open questions related to the topics addressed in this dissertation.

Questions Related to Quantum Chromatic Number

Our lower bounds on quantum chromatic number leave open many open questions regarding this graph parameter. Perhaps the most fundamental question regarding quantum chromatic number is whether it is decidable. The difficulty here lies in the possibly unbounded dimension of the projectors assigned to each vertex. This question was even asked in the work by Cameron et al. where quantum chromatic number was introduced [22]. Being well studied, this question may be difficult to answer, however the more recent semi-definite programming solution for commuting quantum chromatic number [67] may give a starting point. Some more specific, and perhaps more approachable, questions related to our lower bound are listed below.

- The pentagon (C_5) demonstrates that both of the bounds in this dissertation are not lower bounds for the vector chromatic number [54] or the fractional chromatic number [60]. The orthogonal rank, $\xi(G)$, is incomparable to $\chi_q(G)$. The first question we ask here is whether $\xi(G)$ can replace $\chi_q(G)$ in Theorems 2 and 3?

- We have shown that the Kneser graph $K_{p,2}$ has $\chi_q = \chi$, but is this true for all Kneser graphs? Are there any strongly regular graphs with $\chi_q < \chi$?

- In Corollary 1 the dimension d is any finite positive integer. Let $\chi_d(G)$ denote the smallest c for which graph G admits a quantum c-coloring in dimension d. From

Theorem 4, we know that for the orthogonality graph $\Omega(n)$, we have $\chi_1(\Omega(n)) = \chi(\Omega(n))$ which is exponential in n, but $\chi_{pn}(\Omega(n)) = n$ for p a positive integer. This raises the question of what is the value of $\chi_d(\Omega(n))$ for $d \neq pn$? In particular does $\chi_{n+1}(\Omega(n)) = n$?

Questions Related to PMLG

An immediate question raised by this work is whether the same lower-bounds on quantum time complexity can be established for PMLG-exact or any version of PMLG on DAGs. This is interesting in light of improvements in quantum algorithms being possible for the problem of identifying if a string is in a regular language when the regular language is represented by an acyclic monoid [2]. Transferring these results from acyclic monoids to acyclic NFAs, would suggest the difficulty of finding a reduction from LCS to PMLG-exact on DAGs.

Another problem that deserves consideration is whether a reduction is possible from an LCS instance with strings X and Y to a dense graph $G = (V, E)$ where $|V| = o(\min(|X|, |Y|)$. The reductions used in this work (and in other works proving hardness results on PMLG in the classical case) construct sparse graphs where $|E| = \Theta(|V|)$. It is unknown whether an algorithm more efficient than $O(|E|m)$ exists in the classical case where $|E| = \Omega(|V|^{1+\varepsilon})$. Based on our algorithm in Chapter 4, a reduction from LCS to PMLG that constructs a graph where $|E| = |X|$, $m = |Y|$, and $|V| = o(|E|)$ would imply that there exists a quantum algorithm for LCS running in time $O(|X||Y|^{1-\varepsilon} + |X|^{1-\varepsilon}|Y|)$.

Finally, an interesting direction is to develop a strongly subquadratic reduction from PMLG to LCS, proving subquadratic equivalence. LCS is arguably more studied, and such a reduction could have the benefit of transferring new results for LCS to PMLG.

LIST OF REFERENCES

[1] Pangaia, Nov 2020. URL: https://www.pangenome.eu/.

[2] Scott Aaronson, Daniel Grier, and Luke Schaeffer. A quantum query complexity trichotomy for regular languages. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 942–965. IEEE, 2019.

[3] Amir Abboud, Arturs Backurs, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Or Zamir. Subtree isomorphism revisited. *ACM Trans. Algorithms*, 14(3):27:1–27:23, 2018. doi:10.1145/3093239.

[4] Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 59–78. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.14.

[5] Amir Abboud and Karl Bringmann. Tighter connections between formula-sat and shaving logs. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 8:1–8:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.8.

[6] Amir Abboud, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Ryan Williams. Simulating branching programs with edit distance and friends: or: a polylog shaved is a lower bound made. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing,*

STOC 2016, Cambridge, MA, USA, June 18-21, 2016, pages 375–388. ACM, 2016. doi:10.1145/2897518.2897653.

[7] Shyan Akmal and Ce Jin. Near-optimal quantum algorithms for string problems. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 2791–2832. SIAM, 2022. doi:10.1137/1.9781611977073.109.

[8] Amihood Amir, Moshe Lewenstein, and Noa Lewenstein. Pattern matching in hypertext. *J. Algorithms*, 35(1):82–99, 2000. doi:10.1006/jagm.1999.1063.

[9] David Avis, Jun Hasegawa, Yosuke Kikuchi, and Yuuya Sasaki. A quantum protocol to win the graph colouring game on all hadamard graphs. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 89(5):1378–1381, 2006.

[10] Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). *SIAM J. Comput.*, 47(3):1087–1097, 2018. doi:10.1137/15M1053128.

[11] Arturs Backurs, Liam Roditty, Gilad Segal, Virginia Vassilevska Williams, and Nicole Wein. Towards tight approximation bounds for graph diameter and eccentricities. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 267–280. ACM, 2018. doi:10.1145/3188745.3188950.

[12] David A Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc1. *Journal of Computer and System Sciences*, 38(1):150–164, 1989.

[13] Gilles Brassard, Anne Broadbent, and Alain Tapp. Quantum pseudo-telepathy. *Foundations of Physics*, 35(11):1877–1907, 2005.

[14] Gilles Brassard, Richard Cleve, and Alain Tapp. Cost of exactly simulating quantum entanglement with classical communication. *Physical Review Letters*, 83(9):1874, 1999.

[15] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum counting. In *International Colloquium on Automata, Languages, and Programming*, pages 820–831. Springer, 1998.

[16] Andries E Brouwer and Willem H Haemers. *Spectra of graphs*. Springer Science & Business Media, 2011.

[17] Harry Buhrman, Richard Cleve, John Watrous, and Ronald De Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16):167902, 2001.

[18] Harry Buhrman, Richard Cleve, and Avi Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 63–68, 1998.

[19] Harry Buhrman, Bruno Loff, Subhasree Patro, and Florian Speelman. Memory compression with quantum random-access gates. *CoRR*, abs/2203.05599, 2022. `arXiv:2203.05599`, `doi:10.48550/arXiv.2203.05599`.

[20] Harry Buhrman, Subhasree Patro, and Florian Speelman. The quantum strong exponential-time hypothesis. *arXiv preprint arXiv:1911.05686*, 2019.

[21] Harry Buhrman, Subhasree Patro, and Florian Speelman. A framework of quantum strong exponential-time hypotheses. In Markus Bläser and Benjamin Monmege, editors,

38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference), volume 187 of *LIPIcs*, pages 19:1–19:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.STACS.2021.19.

[22] Peter J Cameron, Ashley Montanaro, Michael W Newman, Simone Severini, and Andreas Winter. On the quantum chromatic number of a graph. *the electronic journal of combinatorics*, 14(R81):1, 2007.

[23] Niccolo' Cascarano, Pierluigi Rolando, Fulvio Risso, and Riccardo Sisto. infant: Nfa pattern matching on gpgpu devices. *ACM SIGCOMM Computer Communication Review*, 40(5):20–26, 2010.

[24] Gregory J Chaitin. Register allocation & spilling via graph coloring. *ACM Sigplan Notices*, 17(6):98–101, 1982.

[25] Sai Chen, Peter Krusche, Egor Dolzhenko, Rachel M Sherman, Roman Petrovski, Felix Schlesinger, Melanie Kirsche, David R Bentley, Michael C Schatz, Fritz J Sedlazeck, et al. Paragraph: a graph-based structural variant genotyper for short-read sequence data. *Genome biology*, 20(1):1–13, 2019.

[26] Computational Pan-Genomics Consortium. Computational pan-genomics: status, promises and challenges. *Briefings in bioinformatics*, 19(1):118–135, 2018.

[27] Gabriel Coutinho, Rafael Grandsire, and Célio Passos. Colouring the normalized laplacian. *Electronic Notes in Theoretical Computer Science*, 346:345–354, 2019.

[28] Andrew Cross. The ibm q experience and qiskit open-source quantum computing software. In *APS March meeting abstracts*, volume 2018, pages L58–003, 2018.

[29] Ronald De Wolf et al. *Quantum computing and communication complexity*. Citeseer, 2001.

[30] Christoph Dürr, Mark Heiligman, Peter Høyer, and Mehdi Mhalla. Quantum query complexity of some graph problems. *SIAM J. Comput.*, 35(6):1310–1328, 2006. doi: 10.1137/050644719.

[31] Christoph Durr and Peter Hoyer. A quantum algorithm for finding the minimum. *arXiv preprint quant-ph/9607014*, 1996.

[32] Hannes P Eggertsson, Snaedis Kristmundsdottir, Doruk Beyter, Hakon Jonsson, Astros Skuladottir, Marteinn T Hardarson, Daniel F Gudbjartsson, Kari Stefansson, Bjarni V Halldorsson, and Pall Melsted. Graphtyper2 enables population-scale genotyping of structural variation using pangenome graphs. *Nature communications*, 10(1):1–8, 2019.

[33] Jordan M Eizenga, Adam M Novak, Jonas A Sibbesen, Simon Heumos, Ali Ghaffaari, Glenn Hickey, Xian Chang, Josiah D Seaman, Robin Rounthwaite, Jana Ebler, et al. Pangenome graphs. *Annual Review of Genomics and Human Genetics*, 21, 2020.

[34] Clive Elphick and Pawel Wocjan. Spectral lower bounds for the quantum chromatic number of a graph. *Journal of Combinatorial Theory, Series A*, 168:338–347, 2019.

[35] Massimo Equi, Roberto Grossi, Veli Mäkinen, Alexandru Tomescu, et al. On the complexity of string matching for graphs. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[36] Nick C Fiala and Willem H Haemers. 5-chromatic strongly regular graphs. *Discrete mathematics*, 306(23):3083–3096, 2006.

[37] Travis Gagie, Giovanni Manzini, and Jouni Sirén. Wheeler graphs: A framework for bwt-based data structures. *Theoretical computer science*, 698:67–78, 2017.

[38] Anka Gajentaan and Mark H. Overmars. On a class of o(n2) problems in computational geometry. *Comput. Geom.*, 5:165–185, 1995. doi:10.1016/0925-7721(95)00022-2.

[39] Viktor Galliard, Alain Tapp, and Stefan Wolf. The impossibility of pseudotelepathy without quantum entanglement. In *IEEE International Symposium on Information Theory, 2003. Proceedings.*, page 457. IEEE, 2003.

[40] Viktor Galliard and Stefan Wolf. Pseudo-telepathy, entanglement, and graph colorings. In *Proceedings IEEE International Symposium on Information Theory,*, page 101. IEEE, 2002.

[41] Erik Garrison, Jouni Sirén, Adam M Novak, Glenn Hickey, Jordan M Eizenga, Eric T Dawson, William Jones, Shilpa Garg, Charles Markello, Michael F Lin, et al. Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature biotechnology*, 36(9):875–879, 2018.

[42] Daniel Gibney, Gary Hoppenworth, and Sharma V. Thankachan. Simple reductions from formula-sat to pattern matching on labeled graphs and subtree isomorphism. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 232–242. SIAM, 2021. doi:10.1137/1.9781611976496.26.

[43] Chris Godsil. Interesting graphs and their colourings. *Unpublished notes*, 2003.

[44] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[45] Wilhelmus Hubertus Haemers. Eigenvalue techniques in design and graph theory. 1980.

[46] Willem H Haemers. Interlacing eigenvalues and graphs. *Linear Algebra and its applications*, 226:593–616, 1995.

[47] Willem H. Haemers and Vladimir D. Tonchev. Spreads in strongly regular graphs. *Des. Codes Cryptogr.*, 8(1-2):145–157, 1996. doi:10.1023/A:1018037025910.

[48] Ramesh Hariharan and V. Vinay. String matching in õ(sqrt(n)+sqrt(m)) quantum time. *J. Discrete Algorithms*, 1(1):103–110, 2003. doi:10.1016/S1570-8667(03)00010-8.

[49] Glenn Hickey, David Heller, Jean Monlong, Jonas A Sibbesen, Jouni Sirén, Jordan Eizenga, Eric T Dawson, Erik Garrison, Adam M Novak, and Benedict Paten. Genotyping structural variants in pangenome graphs using the vg toolkit. *Genome biology*, 21(1):1–17, 2020.

[50] Alan J Hoffman. On eigenvalues and colorings of graphs. In *Selected Papers Of Alan J Hoffman: With Commentary*, pages 407–419. World Scientific, 2003.

[51] Jan Holub. Bit parallelism-nfa simulation. In *International Conference on Implementation and Application of Automata*, pages 149–160. Springer, 2001.

[52] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

[53] Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Mip*= re. *Communications of the ACM*, 64(11):131–138, 2021.

[54] David Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM (JACM)*, 45(2):246–265, 1998.

[55] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[56] KR Khadiev and LI Safina. Quantum algorithm for shortest path search in directed acyclic graph. *Moscow University Computational Mathematics and Cybernetics*, 43(1):47–51, 2019.

[57] David Avis1 Jun Hasegawa2 Yosuke Kikuchi and Yuuya Sasaki. A quantum protocol to win the graph colouring game on all hadamard graphs. *arXiv preprint quant-ph/0509047*.

[58] Heng Li, Xiaowen Feng, and Chong Chu. The design and construction of reference pangenome graphs with minigraph. *Genome biology*, 21(1):1–19, 2020.

[59] Antoine Limasset, Jean-François Flot, and Pierre Peterlongo. Toward perfect reads: self-correction of short reads via mapping on de bruijn graphs. *Bioinformatics*, 36(5):1374–1381, 2020.

[60] László Lovász. On the ratio of optimal integral and fractional covers. *Discrete mathematics*, 13(4):383–390, 1975.

[61] Udi Manber and Sun Wu. Approximate string matching with arbitrary costs for text and hypertext. In *Advances In Structural And Syntactic Pattern Recognition*, pages 22–33. World Scientific, 1992.

[62] Laura Mancinska and David E. Roberson. Graph homomorphisms for quantum players. In Steven T. Flammia and Aram W. Harrow, editors, *9th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2014, May 21-23, 2014, Singapore*, volume 27 of *LIPIcs*, pages 212–216. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014. doi:10.4230/LIPIcs.TQC.2014.212.

[63] Dániel Marx. Graph colouring problems and their applications in scheduling. *Periodica Polytechnica Electrical Engineering (Archives)*, 48(1-2):11–16, 2004.

[64] Pierre Morisse, Thierry Lecroq, and Arnaud Lefebvre. Hybrid correction of highly noisy long reads using a variable-order de bruijn graph. *Bioinformatics*, 34(24):4213–4222, 2018.

[65] Kunsoo Park and Dong Kyue Kim. String matching in hypertext. In *Annual Symposium on Combinatorial Pattern Matching*, pages 318–329. Springer, 1995.

[66] Benedict Paten, Adam M Novak, Jordan M Eizenga, and Erik Garrison. Genome graphs and the evolution of genome inference. *Genome research*, 27(5):665–676, 2017.

[67] Vern I Paulsen, Simone Severini, Daniel Stahlke, Ivan G Todorov, and Andreas Winter. Estimating quantum chromatic numbers. *Journal of Functional Analysis*, 270(6):2188–2222, 2016.

[68] Vern I Paulsen and Ivan G Todorov. Quantum chromatic numbers via operator systems. *The Quarterly Journal of Mathematics*, 66(2):677–692, 2015.

[69] Bala Ravikumar and Nicolae Santean. Deterministic simulation of a nfa with k–symbol lookahead. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 488–497. Springer, 2007.

[70] Aritra Sarkar, Zaid Al-Ars, Carmen G Almudever, and Koen Bertels. An algorithm for dna read alignment on quantum accelerators. *arXiv preprint arXiv:1909.05563*, 2019.

[71] Giannicola Scarpa and Simone Severini. The rank-1 quantum chromatic number.

[72] Giannicola Scarpa and Simone Severini. Kochen–specker sets and the rank-1 quantum chromatic number. *IEEE Transactions on Information Theory*, 58(4):2524–2529, 2011.

[73] Gilbert Strang. *Linear algebra and its applications.* Belmont, CA: Thomson, Brooks/-Cole, 2006.

[74] Krysta M. Svore, Alan Geller, Matthias Troyer, John Azariah, Christopher E. Granade, Bettina Heim, Vadym Kliuchnikov, Mariia Mykhailova, Andres Paz, and Martin Roetteler. Q#: Enabling scalable quantum computing and development with a high-level DSL. In *Proceedings of the Real World Domain Specific Languages Workshop, RWDSL@CGO 2018, Vienna, Austria, February 24-24, 2018*, pages 7:1–7:10. ACM, 2018. doi:10.1145/3183895.3183901.

[75] Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005. doi:10.1016/j.tcs.2005.09.023.

[76] Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 645–654. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.67.

[77] Pawel Wocjan and Clive Elphick. Spectral lower bounds for the orthogonal and projective ranks of a graph. *The Electronic Journal of Combinatorics*, pages P3–45, 2019.

[78] Pawel Wocjan, Clive Elphick, and Parisa Darbari. Spectral lower bounds for the quantum chromatic number of a graph–part ii. *The Electronic Journal of Combinatorics*, pages P4–47, 2020.