The Pennsylvania State University
The J.Jeffrey and Ann Marie Fox Graduate School

QUANTUM ALGORITHMS FOR SOLVING POLYNOMIAL SYSTEMS AND PATHFINDING PROBLEMS WITH POTENTIAL SUPERPOLYNOMIAL SPEEDUPS

A Dissertation in
Computer Science and Engineering
by
Jianqiang Li

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2024

The dissertation of Jianqiang Li was reviewed and approved by the following:

Sean Hallgren
Professor of Computer Science and Engineerinng
Dissertation Advisor
Chair of Committee

Martin Fürer
Professor of Computer Science and Engineerinng

Antonio Blanca
Dorothy Quiggle Associate Professor of Computer Science and Engineering

Chunhao Wang
Assistant Professor of Computer Science and Engineering

Jason Morton
Associate Professor of Department of Mathematics

Chitaranjan Das
Program Head
Distinguished Professor of Computer Science and Engineering

Abstract

Finding problems in which quantum algorithms can provide superpolynomial speedup is one of the most challenging tasks in quantum computation. The challenges come from identifying problems that can only be exploited by quantum mechanics. In this thesis, we explore two such problems that have the potential to allow superpolynomial quantum speedup. One is to find a root of a multivariate polynomial system; the other is to find an s-t path in certain type graphs of exponential size. The security of two types of post-quantum cryptosystems, multivariate-based cryptography, and isogeny-based cryptography, depends on the difficulty of solving some instances of those two problems. These two problems are widely believed to be classically hard, while not enough quantum cryptoanalysis has been done.

For the problem of solving a system of polynomial equations, we have shown the limitations and potentials of a particular approach to show the superpolynomial quantum speed-up. The problem of finding the root of a multivariate polynomial system can be reduced to solving an exponentially large linear system. The Quantum Linear System (QLS) algorithm is well known for solving exponentially large quantum linear systems. Using the connection between the linear system and the QLS algorithm, Chen and Gao map the problem of finding a root of a polynomial system to a quantum algorithm, but the running time of their algorithm is unknown. We showed the limitations of the quantum algorithm by proving a lower bound on its running time and showed that the Grover search outperforms this algorithm in many cases. Then we improve the algorithm by running the QLS algorithm on a smaller linear system so that it has the potential to exhibit a superpolynomial speedup for some special polynomial systems.

For the problem of finding an s-t path in graphs of exponential size, we have shown exponential quantum speedups on three families of graphs in the adjacency list oracle, namely the welded tree path graph $\mathcal{G}_P$, the welded tree circuit graph $\mathcal{G}_C$ and the regular sunflower graph $\mathcal{G}_S$. The welded tree path graph $\mathcal{G}_P$ is a non-regular graph, and the quantum algorithm uses the degree information to efficiently find an s-t path. The welded tree circuit graph $\mathcal{G}_C$ is a regular graph, and a newly developed multidimensional electrical network framework is used to generate a quantum superposition state over the edges of the graph to efficiently find an s-t path. The regular sunflower graph $\mathcal{G}_S$ shares structures and expansion properties similar to supersingular isogeny graphs. The quantum algorithm utilizes the 0-eigenspace of the adjacency matrix of $\mathcal{G}_S$ to create a quantum superposition on the vertices of the graph, enabling efficient pathfinding.

These exponential speedups shed light on pathfinding in supersingular isogeny graphs, a cornerstone for isogeny-based cryptosystems, and welded tree graphs, a key challenge in quantum query complexity.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Sean Hallgren, for his unwavering support and guidance over the past six years. I am particularly grateful for his patience and for granting me the freedom to explore the problems that I was passionate about. He has profoundly influenced my research interests, reshaping them at a fundamental level, and I consider myself fortunate to have had him as my advisor. His mentorship has been instrumental in everything I have accomplished during my Ph.D. and will continue to shape my future career.

I would also like to acknowledge the funding sources that made this thesis possible: National Science Foundation awards CCF-1618287, CNS-1617802, CCF-1617710, and the Vannevar Bush Faculty Fellowship from the US Department of Defense. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation and the US Department of Defense.

I am also incredibly thankful to Sevag Gharibian, my previous advisor, who has been a constant source of support since I first arrived in the United States to pursue a Ph.D. I thoroughly enjoyed working with him, learning from him, and visiting him. His insights on Hamiltonian complexity and his balance between academia and family life have left a lasting impression on me.

During my Ph.D., I had the pleasure of collaborating with András Gilyén, Sebastian Zur, and Yu Tong in addition to my advisor. I am grateful for the opportunity to work alongside such talented researchers and I have learned a great deal from them, both in terms of algorithmic techniques and life experiences. I also want to thank Andrew Childs for referring me to the multidimensional quantum walk paper and for hosting me during a short visit to QuICS, which led to part of the results presented in this thesis.

I would also like to express my gratitude to the members of the theory group and friends at PSU for their constant help and support. A special thanks to Antonio Blanca for answering my many questions about random walks, and to Young Kun Ko for sharing invaluable life experiences and career advice. Conversations with Eunou Lee, Mahdi Belbasi, Chaowen Guan, Xusheng Zhang, Jeremy A. Huang, Heehyun Park, Medha Kumar, Zecheng Li, and Katiyar Akshit have been memorable and enriching. I am also thankful to Adrian Rublein for spending time improving my writing.

I also thank my committee members Martin Fürer and Jason Morton for their support and feedback.

In addition, I am deeply grateful to Kirsten Eisenträger, who not only shared her experiences with raising a baby, but also provided valuable insights on isogeny graph pathfinding. Her support encouraged both me and my wife during some of the most challenging times, and her guidance has been greatly appreciated.

I am especially thankful to Chunhao Wang, who has been supportive not only in my academic endeavors but also in helping with the many challenges of raising a baby. I also appreciate Nai-Hui Chia for his career advice and for being available whenever I needed help. My heartfelt thanks to Yuan Su, who not only taught me many quantum concepts and shared his unpublished notes, but also answered countless questions I had, while serving as a role model that I aspire to emulate.

Lastly, I want to express my deepest thanks to my wife, Mingming Chen, who has been unwaveringly supportive throughout this journey. To my two-year-old daughter, Chloe Amelia Li, you have brought immeasurable joy and color to my life, just as Sean Hallgren said you would. I am also grateful to my parents, Shengbao Li and Fengling Wang, and my mother-in-law, Jing Li, for traveling such a long distance to visit and support us during the most challenging times.

Chapter 1

Introduction

In 1994, Peter Shor [Sho97] introduced an efficient quantum algorithm for factoring integers and for the discrete logarithm problem, both of which form the basis of widely used public-key cryptosystems on the internet, and for which no efficient classical algorithms are known. Shor's algorithm uses quantum mechanics to run on a large-scale, fault-tolerant quantum computer. Since then, considerable efforts have been made to develop new quantum algorithms for problems that classical computers are unable to solve efficiently.

Since the invention of Shor's algorithm, there have been many advances in quantum algorithms. These include Grover search [Gro96], the adiabatic quantum computation algorithm [FGGS00], quantum walks [Sze04, CCD$^+$03, Bel13], the quantum approximate optimization algorithm [FGG14], the QLS algorithm [HHL09], and more recently, the quantum singular value transformation framework [GSLW18].

Despite these advances in quantum algorithms, only a handful of problems have been discovered that admit superpolynomial advantages in the past three decades [Aar22]. In addition to factoring and the discrete logarithmic problem, these problems also include Pell's equations [Hal07], the unit group of a number field [Hal05, EHKS14], the class group, the principal ideal problem [Hal05, BS16], computing Ray class groups, Hilbert class fields [EH10], approximate Jones polynomial [AJL06] and more recently an NP search problem [YZ24]. The quantum algorithms for those problems are mostly based on the quantum Fourier transform.

Identifying problems that can be leveraged by quantum algorithms to achieve superpolynomial speedup over classical algorithms remains one of the major challenges in the field of quantum computation. These challenges arise from two aspects. First, the problem needs to be believed to be hard classically. Second, the problem also needs to have an

efficient quantum algorithm. Several previous attempts to identify such problems have failed due to these constraints. For example, hopes of achieving exponential speedup with quantum algorithms in various machine learning problems, including recommendation systems [KP17], and principal component analysis [LMR14], were later refuted by Tang's breakthrough results [Tan19] and subsequent works [Tan21, CGL$^+$20]. Specifically, leveraging the assumptions used to prepare the input quantum states, [Tan19] presents a classical algorithm for the recommendation system problem that achieves a time complexity comparable to that of the quantum algorithm proposed by [KP17].

In addition to the long-standing Graph Isomorphism (GI) problem, one potential area to find such problems is related to post-quantum cryptography. Four types of classical post-quantum public-key cryptosystems, including lattice-based systems, code-based cryptography, multivariate-based cryptography, and isogeny-based cryptosystems, have been proposed as candidates to resist attacks based on quantum computers. Although the problems underlying these cryptosystems are believed to be classically hard, not much quantum cryptanalysis has been done. In particular, these problems are also not NP-complete for which quantum algorithms are believed also cannot provide efficient quantum algorithms. Therefore, an efficient quantum algorithm for any one of these problems will likely provide a superpolynomial quantum speedup. The complexity class of these problems can be summarized in Figure 1.1.

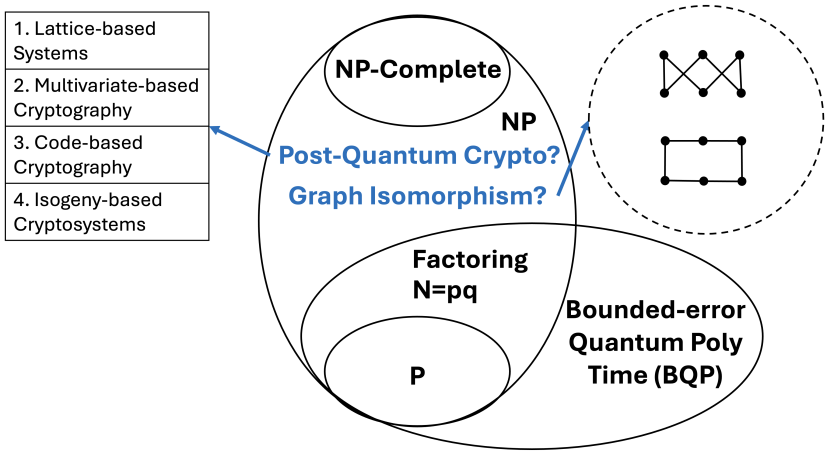

Figure 1.1. P: Problems can be solved in polynomial time; NP: Problems can be verified in polynomial time; NP-Complete: the hardest problem in NP; BQP: Problems can be solved in polynomial time in a quantum computer.

There are two main approaches to demonstrating quantum superpolynomial speedups. The first approach is to find an efficient quantum algorithm for a classically hard problem that involves both classical input and output. The second approach is to construct

an oracle problem and find a quantum algorithm that can provide a superpolynomial speedup relative to an oracle. This developed technique can then be extended to non-oracle problems. A prime example of the second approach is Simon's problem [Sim97], for which Simon's algorithm provides an exponential quantum speedup relative to an oracle. In particular, the quantum speedup demonstrated by Simon's algorithm for the oracle problem played a crucial role in the development of Shor's algorithm for the factoring problem [Sho97].

In this thesis, our aim is to explore the possibilities of quantum superpolynomial speedups for the problem of finding a Boolean solution of polynomial systems and the problem of finding an s-t path in graphs of exponential size. In particular, the security of multivariate-based cryptography and isogeny-based cryptography relies on the hardness of solving some polynomial system instances and finding an s-t path in certain types of isogeny graphs, respectively. The two problems considered in this thesis are summarized in Fig. 1.2.

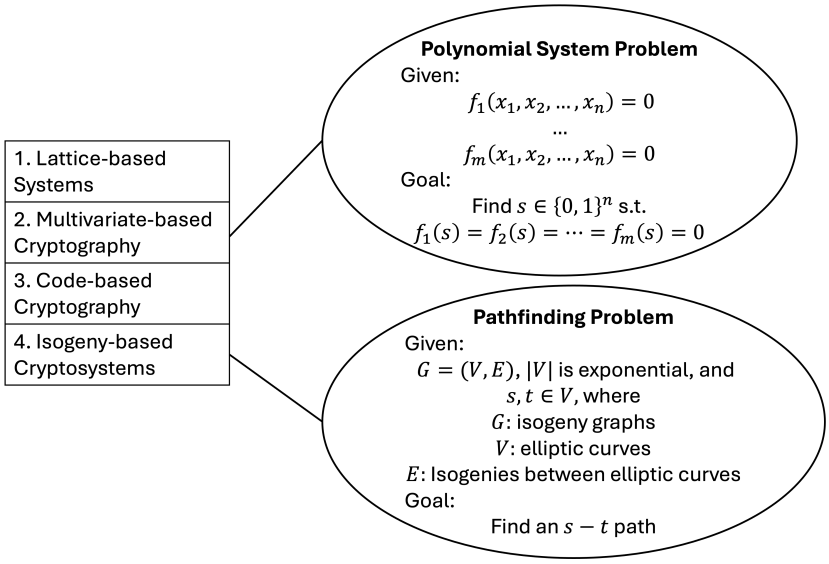

Figure 1.2. Problem of Solving Polynomial Systems and the Pathfinding Problem in Supersingular Isogeny Graphs

For the problem of finding a root of a multivariate polynomial system, we explore the potential for superpolynomial quantum speedup in the non-oracle setting in [DGG+23]. Chen and Gao [CG22] mapped the problem of finding a root of a polynomial system to a quantum algorithm based on the QLS algorithm. Importantly, both the input and output of their quantum algorithm involve classical information that can be processed in polynomial time. However, the running time of their algorithm depends on the condition

number of a related linear system, which was previously unknown. We addressed this by establishing a lower bound on the condition number, revealing the limitations of their quantum algorithm. Additionally, we enhanced the algorithm to have the potential for superpolynomial speedup in certain special polynomial systems.

For the problem of finding an s-t path in supersingular isogeny graphs, which is a class of expander graphs, we adopt the second approach by investigating quantum speedups on constructed graphs in [Li23, LZ23, LT24] and the graphs are given through adjacency list oracle. In particular, we constructed three types of graphs, namely the welded tree path graph $\mathcal{G}_P$, the welded tree circuit Graph $\mathcal{G}_P$, and the regular sunflower graph $\mathcal{G}_S$, and developed three types of new quantum algorithms for which exponential quantum speedup can be achieved to find an s-t path in these graphs. We summarize the graphs having exponential speedups in the following Fig. 1.3:

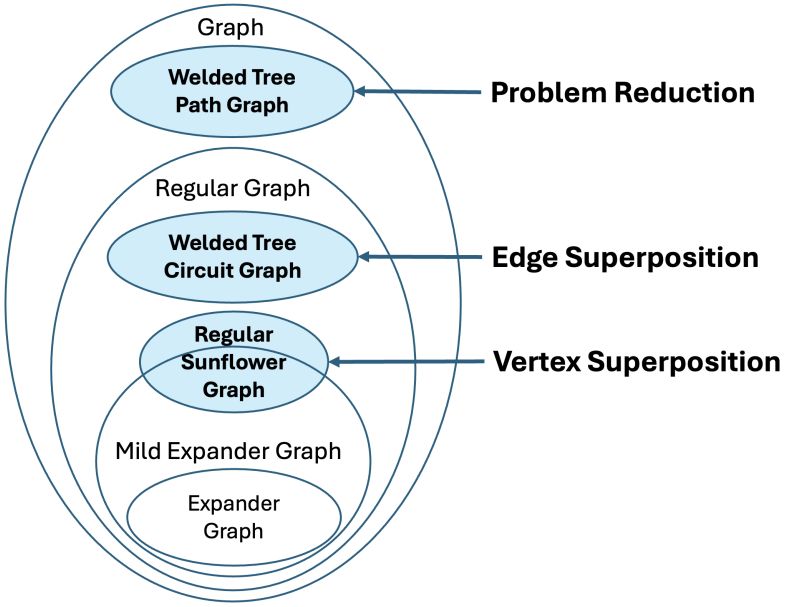

Figure 1.3. Exponential Separations Landscape for Pathfinding Problems

- Problem Reduction [Li23]: Reduce the s-t pathfinding problem in the welded tree path graph $\mathcal{G}_P$ to vertex finding problem in the welded tree graph.
- Edge Superposition [LZ23]: Use the multidimensional electrical network framework to generate a quantum superposition state over edges. This quantum state has significant overlap with edges along an s-t path in the welded tree circuit graph $\mathcal{G}_C$.

- Vertex Superposition [LT24]: Use quantum eigenstate filtering technique to prepare a 0-eigenstate of the adjacency matrix as a quantum superposition state over vertices. This quantum state has a large overlap with the vertices of an s-t path in the regular sunflower graph $\mathcal{G}_S$.

Thesis Organization: In Chapter 2, we introduce several quantum algorithmic primitives that will be useful for solving polynomial systems and addressing the pathfinding problem. In Chapter 3, we discuss the limitations and potentials of using the QLS algorithm for solving multivariate polynomial systems. In Chapter 4, we demonstrate how quantum algorithms can achieve exponential quantum-classical separations for pathfinding problems in three types of graphs. Finally, in Chapter 5, we conclude by discussing potential future research directions.

Chapter 2

Preliminaries

In this chapter, we first introduce the high-level idea and results of several quantum algorithms within the quantum singular value transformation (QSVT) framework [GSLW18], including the QLS algorithm [HHL09, CKS17, GSLW18], the continuous quantum walk [CCD$^+$03] and the quantum eigenstate filtering algorithm [LT19a]. Additionally, we cover the background of the discrete quantum walk based on electrical networks [Bel13, Pid19, AP22]. These quantum algorithms serve as foundational building blocks for quantum algorithms for polynomial system problems and pathfinding problems in graphs, both of which are central to this thesis.

2.1 Block Encoding

In this section, following [Gil19, GSLW18] and given sparse access to the Hermitian matrix A, we introduce the result of constructing a unitary U that is a block encoding of the Hermitian matrix A. Furthermore, by simulation of the sparse access oracle of the adjacency matrix A of a given graph using the adjacency list oracle, we also show the result of constructing a unitary U that is a block encoding of the adjacency matrix A of a given graph.

Definition 2.1.1 (Block encoding [GSLW18, Definition 43])**.** *Suppose that A is an s-qubit operator and let $\alpha, \epsilon \in \mathbb{R}_+$ and $a \in \mathbb{N}$. We say that the $(s+a)$-qubit unitary U is an (α, a, ϵ) block encoding of A on registers β_1 and β_2, if*

$$\|A - \alpha \left(\langle 0|_{\beta_1}^{\otimes a} \otimes I_{\beta_2}\right) U \left(|0\rangle_{\beta_1}^{\otimes a} \otimes I_{\beta_2}\right)\| \leq \epsilon.$$

The above definition means that we can construct a unitary circuit in which a certain

sub-matrix corresponds to the matrix A that we want to process in our algorithm. In particular, when $\epsilon = 0$, we have

$$U = \begin{pmatrix} \frac{A}{\alpha} & \cdot \\ \cdot & \cdot \end{pmatrix}$$

and

$$U |0\rangle^{\otimes a} |\psi\rangle = |0\rangle^{\otimes a} \frac{A}{\alpha} |\psi\rangle + |\perp\rangle ,$$

where the reduced state in the first a qubits of $|\perp\rangle$ is orthogonal to $|0\rangle^{\otimes a}$.

We will construct such a block encoding by treating the matrix A as a sparse matrix and thus applying [GSLW18, Lemma 48]. The sparse matrix block encoding in this lemma requires access to the *sparse access oracle*, which we define below:

Definition 2.1.2 (Sparse access oracle [GSLW18, Lemma 48])**.** *Let $A \in C^{2^n \times 2^n}$ be a Hermitian matrix that is d sparse, i.e., each row and each column have at most d nonzero entries.*

Let A_{ij} be a b-bit binary description of the ij-matrix element of A. The sparse-access oracle O_{val} is defined as follows.

$$O_{\text{val}} : |v\rangle |v'\rangle |0\rangle^{\otimes b} \longrightarrow |v\rangle |v'\rangle |A_{ij}\rangle \qquad \forall v, v' \in [2^n] - 1.$$

Let r_{vk} be the index for the k-th non-zero entry of the v-th row of A. If there are fewer than k nonzero entries, then $r_{vk} = k + 2^n$. The sparse access oracle O_{loc} is defined as follows.

$$O_{\text{loc}} : |v\rangle |k\rangle \longrightarrow |v\rangle |r_{vk}\rangle \qquad \forall v \in [2^n] - 1, k \in [s_r].$$

In the above, we assume only one oracle to access the location of non-zero entries in each row, rather than two separate oracles for rows and columns, respectively, as in [GSLW18, Lemma 48]. This is because we focus on a Hermitian matrix, and therefore the row and column oracles are identical.

In the classical setting, we will access a graph (multigraph) through an adjacency list oracle. We note that the sparse access oracle defined above can be simulated by the adjacency list oracle, as stated in the following Lemma 2.1.1.

Definition 2.1.3 (The adjacency list oracle)**.** *Let $G = (V, E)$ be an undirected multigraph of maximum degree $d = O(1)$, $V = \{0, 1, 2 \cdots, \mathsf{N} - 1\}$, $n = \lceil \log_2(\mathsf{N}) \rceil$. For any $v \in V$ and $k \in \{1, 2, \cdots, d\}$, if the k-th neighbor (using the natural ordering of bit-strings) of the vertex $v \in \mathcal{V}$ (not counting multiplicity) exists, then $O_{G,1}(v, k)$ returns*

the this neighbor. If the kth neighbor does not exist, then $O_{G,1}(v,k) = k + 2^n$*. For any* $v, v' \in V$*,* $O_{G,2}(v,v')$ *returns the multiplicity of the edge* $\{v,v'\}$*. For simplicity, we also use* O *or* O_G *to represent the adjacency list oracle* $O_{G,1}$ *when the graph is a simple graph.*

There are several other oracle models that can be considered. For example, we can let $O_G(v,k)$ return the k-th neighbor counting multiplicity, or let O_G return a list of all neighbors at the same time. These alternative models are equivalent to the one we are considering up to the $\mathrm{poly}(d) = O(1)$ overhead.

Lemma 2.1.1. *Let* A *be the adjacency matrix of a given graph* G *of maximum degree* d*. The sparse access oracle* O_{val}*,* O_{loc} *of the adjacency matrix* A *of* G *can be simulated by* $O(d)$ *queries of the adjacency list oracle* $O_{G,1}$*,* $O_{G,2}$*.*

Proof. Given two vertices $i, j \in V$, the sparse access oracle O_{val} returns the matrix element $A_{i,j}$. This is equal to the edge multiplicity between the vertices i, j returned by the oracle $O_{G,2}$. Therefore, a single use of $O_{G,2}$ can simulate the oracle O_{val}.

For any $(i,k) \in V \times \{0,1,2,\ldots,d-1\}$, the sparse access oracle O_{loc} returns the index r_{ik} of the $(k+1)$-th nonzero entry of the ith row of the adjacency matrix A. If there are less than $k+1$ non-zero entries, O_{loc} returns the index $k + 2^n$.

The adjacency list oracle $O_{G,1}$ returns the $(k+1)$-th neighbor i_k of the vertex $i \in V$ if it exists. Using at most d queries of $O_{G,1}$, we can obtain all neighbors of u and reorder them as $v_0, v_1, \ldots, v_{h-1} \in V$ using natural ordering of the bitstring. Therefore, using at most d queries of $O_{G,1}$, we can obtain the index of $(k+1)$-th nonzero entries of the i-th row of the adjacency matrix of A, that is, $r_{ik} = i_k$. If the $(k+1)$-th neighbor i_k of the vertex $i \in V$ does not exist, we have $O_{G,1}(v,k) = k + 2^n$, which is equal to the element returned by O_{loc}. Therefore, we can simulate the sparse access oracle O_{loc} using at most d queries of $O_{G,1}$.

Hence, using at most $O(d)$ queries of the adjacency list oracles $O_{G,1}, O_{G,2}$ defined in Definition 2.1.3, we can simulate the sparse access oracle O_{local}, O_A as defined in Definition 2.1.2. □

Lemma 2.1.2 (Block encoding of sparse-access matrices [GSLW18, Lemma 48])**.** *Let* $A \in \mathbb{C}^{2^n \times 2^n}$ *be a matrix that is* s_r*-row sparse and* s_c*-column-sparse, and each element* a_{ij} *of* A *has absolute value at most 1. We also assume that the* b*-bit binary description of* a_{ij} *is exact. We can implement a unitary* U_A*, which is a* $(\sqrt{s_r s_c}, n+3, \epsilon)$*-block-encoding of* A *with a single use of* O_{local}*, two uses of* O_{val} *and additionally using* $O(n + \log^{2.5}(\frac{s_r s_c}{\epsilon}))$ *one and two qubit gates while using* $O(b, \log^{2.5}(\frac{s_r s_c}{\epsilon}))$ *ancilla qubits.*

For simplicity, we assume that the block encoding error in Definition 2.1.1 is $\epsilon = 0$. This is a standard assumption for two reasons. First, the complexity dependence on ϵ is logarithmic when constructing the unitary U_A as indicated in Lemma 2.1.2. Second, for QSVT-based quantum algorithms in this thesis that use U_A as a subroutine, the block encoding error is insignificant because of the robustness of QSVT [GSLW18][Section 3.3].

2.2 Quantum Linear System Algorithm

In this section, we first introduce the Quantum Linear System Problem (QLSP) and sketch the basic idea of a quantum algorithm for solving the QLSP in the QSVT framework. We also discuss various QLS algorithms for QLSP and state the resulting time complexity of the QLS algorithm to solve QLSP.

Problem 2.2.1 (LSP)**.** *Given a Hermitian $A \in \mathbb{R}^{N\times N}$, the Linear System Problem (LSP) is to find $x \in \mathbb{R}^N$ such that*

$$Ax = b.$$

Problem 2.2.2 (QLSP)**.** *Given the sparse access oracle of the Hermitian $A \in \mathbb{R}^{N\times N}$ as Definition 2.1.2, a quantum state encoding the vector $b = [b_0, b_1, \ldots, b_{N-1}]^T \in \mathbb{R}^N$ as*

$$|b\rangle = \frac{1}{\|b\|_2} \sum_{i=0}^{N-1} b_i |i\rangle .$$

Let $|x\rangle = \frac{1}{\|x\|_2} \sum_{i=0}^{N-1} x_i |i\rangle$ be the quantum state such that $Ax = b$. The QLSP is to find a quantum state $|\tilde{x}\rangle$ so that

$$\| |\tilde{x}\rangle - |x\rangle \| \leq \epsilon.$$

The assumption that the matrix A is a Hermitian matrix is without loss of generality since we can construct a new linear system when A is not Hermitian

$$\begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix} \begin{pmatrix} 0 \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}.$$

The time complexity of classical algorithms for solving LSP has a polynomial dependence on the dimension of the matrix A [S$^+$94]. For QLSP, [HHL09] proposed the first QLS algorithm that takes time $\tilde{O}(\mathsf{s}^2 \log N\kappa^2/\epsilon)$, where s is the sparsity of the matrix A

and κ is the condition number of the matrix A. Later, using the variable-amplitude amplification technique, [Amb10] improved the dependence on κ from quadratic to linear as $\tilde{O}(\mathsf{s}^2\kappa/\epsilon^3)$ at the cost of precision. Using linear combinations of unitaries and the Chebyshev polynomial approximation of the inverse function $1/x$, the dependence on $1/\epsilon$ was then further improved from linear to $\mathrm{poly}\log(1/\epsilon)$ with the time complexity of the QLS algorithm $\tilde{O}(\mathsf{s}\kappa\mathrm{poly}\log(1/\epsilon))$[CKS17]. The QLS algorithm in [CKS17] can be evaluated within the QSVT framework [GSLW18] and has a time complexity $\tilde{O}(\mathsf{s}\kappa\mathrm{poly}\log(1/\epsilon))$. In particular, the Chebyshev polynomial that is used for the approximation of function $1/x$ can be stated as follows:

Definition 2.2.1 (Chebyshev Polynomials of the First Kind)**.**

$$\mathcal{T}_0 = 1, \qquad \mathcal{T}_1(x) = x \qquad \textit{and} \qquad \mathcal{T}_{n+1}(x) = 2x\mathcal{T}_n(x) - \mathcal{T}_{n-1}(x).$$

We also have $\mathcal{T}_n(\cos\theta) = \cos(n\theta)$.

Lemma 2.2.3 (Polynomial approximations of $1/x$ [GSLW18, CKS17])**.** *Let* $\kappa > 1$ *and* $\epsilon \in (0, 1/2)$*. For* $b = \lceil \kappa^2 \log(\kappa/\epsilon) \rceil$ *the odd function*

$$f(x) := \frac{1 - (1 - x^2)^b}{x} \tag{2.1}$$

is ϵ*-close to* $1/x$ *on the domain* $[-1, 1]\backslash(- - \frac{1}{\kappa}, \frac{1}{\kappa})$*. Let* $J := \lceil \sqrt{b \log(4b/\epsilon)} \rceil$*, then the* $O(\kappa \log(\frac{\kappa}{\epsilon}))$*-degree odd real polynomial*

$$P(x) := 4 \sum_{j=0}^{J} (-1)^j \frac{\sum_{i=j+1}^{b} \binom{2b}{b+i}}{2^{2b}} T_{2j+1}(x) \tag{2.2}$$

is ϵ*-close to* $f(x)$ *on the interval* $[-1, 1]$*, moreover* $|P(x)| \leq 4J = O(\kappa \log(\frac{\kappa}{\epsilon}))$ *on this interval.*

Given the sparse access of the Hermitian matrix A, we can construct a unitary U_A that block encodes the matrix A as a submatrix using the result in Section 2.1. With U_A, QSVT [GSLW18] or LCU [CKS17] allows us to implement a unitary V such that

$$V \left|0^{\otimes a}\right\rangle \left|\psi\right\rangle = \left|0^{\otimes a}\right\rangle P(A) \left|\psi\right\rangle + \left|\perp\right\rangle .$$

Measurement of the first register on the computational basis to obtain all $0^{\otimes a}$ with probability $\| P(A) \left|\psi\right\rangle \|_2^2$, then the result state is $\frac{P(A)|\psi\rangle}{\|P(A)|\psi\rangle\|} \approx \frac{A^{-1}|\psi\rangle}{\|A^{-1}|\psi\rangle\|}$. Combining this with

variable-time amplitude amplification technique [Amb10], we can obtain the resulting time complexity of the QLS algorithm as $\tilde{O}(\mathsf{s}\kappa\mathrm{poly}\log(1/\epsilon))$. We summarize the result as follows.

Theorem 2.2.4 ([GSLW18, CKS17])**.** *The QLSP can be solved by gate efficient quantum algorithm in time* $\tilde{O}(\mathsf{s}\kappa\mathrm{poly}\log(1/\epsilon))$.

In addition to the approach of approximating the inverse function A^{-1}, another approach to solving QLSP is based on the adiabatic quantum approach [SSO19, LT19b, CAS+22]. The difference between the two approaches arises when the matrix A is singular and QLSP has an infinite number of solutions. In this case, the first approach implements the Moore pseudoinverse A^{+}, and the solution x returned by the QLS algorithm of this type has the minimum ℓ_2 norm. For the second adiabatic quantum approach, it is unclear what properties the returned solution x could have.

The state-of-the-art QLS algorithm is based on the adiabatic quantum approach [CAS+22], which solves QLSP in time $O(\kappa\log(1/\epsilon))$ under the condition that U_A is given. Since we investigate problems with potential superpolynomial speedup and the QLS algorithm based on polynomials is easy to explain; therefore, we use the less optimal result for QLSP and this does not affect our result for problems considered in this thesis.

2.3 Continuous Time Quantum Walk

In this section, we introduce the result of the implementation of the unitary evolution $e^{-iAt}|\psi\rangle$ with $t\in\mathbb{R}$, where $|\psi\rangle$ is the initial state and A is the adjacency matrix of a graph G. In particular, we assume that the adjacency matrix A of graph G is given through a unitary U_A, which is a block encoding the adjacency matrix A as defined in Section 2.1. The idea is to use polynomials to approximate the function e^{-ix} and then to implement the polynomial transformation within QSVT.

First, since $e^{-ix}=\cos x - i\sin x$, the following Lemma 2.3.1 states the polynomials that provide a good approximation of $\cos x$ and $\sin x$, therefore also providing a good approximation of e^{-ix}.

Lemma 2.3.1 (Polynomial approximation of e^{-ix} (Lemma 57 in [GSLW18]))**.** *Let* $t\in\mathbb{R}\backslash\{0\}, \epsilon\in(0,\frac{1}{e})$*, and let* $R=\lfloor r\left(\frac{e|t|}{2},\frac{5}{4}\epsilon\right)/2\rfloor$*, then the following* $2R$ *and* $2R+1$

degree polynomials satisfying

$$\left\| \cos(tx) - J_0(t) + \sum_{k=1}^{R} (-1)^k J_{2k}(t) \mathcal{T}_{2k}(x) \right\|_{[-1,1]} \leq \epsilon$$

and

$$\left\| \cos(tx) - 2 \sum_{k=0}^{R} (-1)^k J_{2k+1}(t) \mathcal{T}_{2k+1}(x) \right\|_{[-1,1]} \leq \epsilon,$$

where $J_m(t) : m \in \mathbb{N}$ denote Bessel functions of the first kind.

With this polynomial approximation of e^{-ix} and using QSVT, the complexity of implementing e^{-iAt} can be state as follows:

Theorem 2.3.2 (Complexity of block-Hamiltonian simulation (Corollary 60 in [GSLW18]))**.** *Let $\epsilon \in (0, 1/2)$, $t \in \mathbb{R}$ and $\alpha \in \mathbb{R}_+$. Let U_A be an $(\alpha, a, 0)$ block encoding of the adjacency matrix A. In order to implement an ϵ-precise Hamiltonian simulation unitary V which is an $(1, a+2, \epsilon)$ block encoding of e^{-iAt}, that is*

$$\| e^{-iAt} - \left(\langle 0|_{\beta_1}^{\otimes a+2} \otimes I_{\beta_2} \right) V \left(|0\rangle_{\beta_1}^{\otimes a+2} \otimes I_{\beta_2} \right) \| \leq \epsilon,$$

it is necessary and sufficient to use U_A a total number of times

$$\Theta \left(\alpha |t| + \frac{\log(1/\epsilon)}{\log(e + \log(1/\epsilon)/(\alpha |t|)} \right).$$

2.4 Quantum Eigenstate Filtering Algorithm

In this section, we introduce the complexity of implementing the projection operator Π_0 into the 0-eigenspace of the adjacency matrix A of a graph by using the quantum singular value transformation (QSVT) technique [GSLW19, LC17], and the minimax filtering polynomial proposed in [LT19a].

The filtering polynomial takes the form

$$R_\ell(x; \Delta) = \frac{\mathcal{T}_\ell \left(-1 + 2 \frac{x^2 - \Delta^2}{1 - \Delta^2} \right)}{\mathcal{T}_\ell \left(-1 + 2 \frac{-\Delta^2}{1 - \Delta^2} \right)}, \tag{2.3}$$

where $\mathcal{T}_\ell(x)$ is the ℓ-th Chebyshev polynomial of the first kind (Definition 2.2.1), and the degree of the polynomial is therefore ℓ. This polynomial peaks at $x = 0$ and is close to

0 for $\Delta \leq |x| \leq 1$. It is minimax in the sense that the deviation from 0 for $\Delta \leq |x| \leq 1$ is minimal among all polynomials of degree ℓ that take the value 1 at $x = 0$.

The properties of $R_\ell(x;\Delta)$ are stated in [LT19a, Lemma 2], which we restate here:

Lemma 2.4.1 (Lemma 2 of [LT19a]). *Let $R_\ell(x;\Delta)$ be as defined in (2.3). It satisfies the following:*

(i) $R_\ell(x;\Delta)$ solves the minimax problem

$$\underset{p(x)\in\mathbb{P}_{2\ell}[x],p(0)=1}{\text{minimize}} \max_{x\in\mathcal{D}_\Delta} |p(x)|.$$

(ii) $|R_\ell(x;\Delta)| \leq 2e^{-\sqrt{2}\ell\Delta}$ for all $x \in \mathcal{D}_\Delta$ and $0 < \Delta \leq 1/\sqrt{12}$. Also $R_\ell(0;\Delta) = 1$.

(iii) $|R_\ell(x;\Delta)| \leq 1$ for all $|x| \leq 1$.

With the unitary U_A and the minimax filtering polynomial $R_\ell(x;\Delta)$, the complexity of constructing a unitary V such that

$$V\,|0^{a+1}\rangle\,|\psi\rangle = |0^{a+1}\rangle\,R_\ell(A/\alpha;\Delta)\,|\psi\rangle + |\perp\rangle$$

can be stated as follows:

Theorem 2.4.2 (Theorem 3 of [LT19a]). *Let A be a adjacency matrix and U_A is an $(\alpha, a, 0)$-block-encoding of A. If 0 is an eigenvalue of A that is separated from the rest of the spectrum by a gap Δ and Π_0 is the 0-eigenspace projection operator of A, then we can construct a unitary V, which is a $(1, a+1, \epsilon)$-block-encoding of Π_0, by $\mathcal{O}((\alpha/\Delta)\log(1/\epsilon))$ applications of (controlled-) U_A and $U_A^\dagger$, and $\mathcal{O}((a\alpha/\Delta)\log(1/\epsilon))$ other primitive quantum gates.*

2.5 Quantum Walk and Electrical Networks

In this section, we introduce the complexity of using quantum walk to generate a quantum state that encodes an s-t electrical flow state based on the recent work [Pid19, AP22]. We first define graph-theoretic concepts and basic knowledge of electrical networks following [Vis13, JZ23].

Definition 2.5.1 (Network). *A network is a connected weighted graph $G = (V, E, \mathsf{w})$ with a vertex set V, an (undirected) edge set E and some weight function $\mathsf{w} : E \to \mathbb{R}_{>0}$.*

Since edges are undirected, we can equivalently describe the edges by some set $\overrightarrow{E}$ *such that for all* $(u,v) \in E$*, exactly one of* (u,v) *or* (v,u) *is in* $\overrightarrow{E}$*. The choice of edge directions is arbitrary. Then we can view the weights as a function* $\mathsf{w} : \overrightarrow{E} \to \mathbb{R}_{>0}$*, and for all* $(u,v) \in \overrightarrow{E}$*, define* $\mathsf{w}_{v,u} = \mathsf{w}_{u,v}$*. For convenience, we will define* $\mathsf{w}_{u,v} = 0$ *for every pair of vertices such that* $(u,v) \notin E$*. For an implicit network* G*, and* $u \in V$*, we will let* $\Gamma(u)$ *denote the* neighborhood *of* u*:*

$$\Gamma(u) := \{v \in V : (u,v) \in E\}.$$

We use the following notation for the out- and in-neighborhood *of* $u \in V$*:*

$$\begin{aligned} \Gamma^+(u) &:= \{v \in \Gamma(u) : (u,v) \in \overrightarrow{E}\} \\ \Gamma^-(u) &:= \{v \in \Gamma(u) : (v,u) \in \overrightarrow{E}\}, \end{aligned} \tag{2.4}$$

Definition 2.5.2 (Flow, Circulation)**.** *A* flow *on a network* $G = (V, E, \mathsf{w})$ *is a real-valued function* $\theta : \overrightarrow{E} \to \mathbb{R}$*, extended to edges in both directions by* $\theta_{u,v} = -\theta_{v,u}$ *for all* $(u,v) \in \overrightarrow{E}$*. For any flow* θ *on* G*, vertex* $u \in V$*, and subset* $A \subseteq V$ *we define* $\theta_u = \sum_{v \in \Gamma(u)} \theta_{u,v}$ *as the flow coming out of* u*. If* $\theta_u = 0$*, we say the flow is conserved at* u*. If the flow is conserved at every vertex, we call* θ *a* circulation*. If* $\theta_u > 0$*, we call* u *a* source*, and if* $\theta_u < 0$ *we call* u *a* sink*. A flow with a unique source* s *and unique sink* t *(satisfying* $\theta_s = -\theta_t = -1$*) is called an* (unit) s-t flow*. The* energy *of any flow* θ *is*

$$\mathcal{E}(\theta) := \sum_{(u,v) \in \overrightarrow{E}} \frac{\theta_{u,v}^2}{\mathsf{w}_{u,v}}.$$

The effective resistance $\mathcal{R}_{s,t}$ *is given by the minimal energy* $\mathcal{E}(\theta)$ *over all unit flows* θ *from* s *to* t*. The* s-t electrical flow *is the unique unit* s-t *flow that achieves this minimal energy.*

Definition 2.5.3 (Electrical Network)**.** *Given a network* $G = (V, E, \mathsf{w})$ *with a weight function* w*, we can interpret every edge* $(u,v) \in E$ *as a resistor with resistance* $1/\mathsf{w}_{u,v}$*. This allows* G *to be modeled as an* electrical network*.*

Two fundamental laws related to electrical networks are Kirchhoff's Law (also known as Kirchhoff's Node Law) and Ohm's Law. The former states the definition of a s-t flow, as in Definition 2.5.2:

Definition 2.5.4 (Kirchhoff's Law)**.** *For any* s-t *flow on an electrical network* $G =$

(V, E, w) *with* $s, t \in V$*, the amount of electrical flow that enters any* $u \in V \backslash \{s, t\}$ *is equal to the amount of flow that exits* u*, that is,* $\sum_{v \in \Gamma(u)} \theta_{u,v} = 0$*.*

The latter states that if we inject a unit of current into s and extract it from t in the electrical network G, then there is an induced potential vector p which relates to the s-t electrical flow θ:

Definition 2.5.5 (Ohm's Law)**.** *Let* θ *be the* s-t *electrical flow in an electrical network* $G = (V, E, \mathsf{w})$ *with* $s, t \in V$*. Then there exists a potential vector* p *such that the potential difference between the two endpoints of any edge* $(u, v) \in E$ *is equal to the amount of electrical flow* $\theta_{u,v}$ *along this edge multiplied with the resistance* $1/\mathsf{w}_{u,v}$*, that is,* $p_u - p_v = \theta_{u,v}/\mathsf{w}_{u,v}$*.*

The potential p induced by an s-t electrical flow θ in Ohm's Law is not unique and it is therefore convention to consider the potential p that assigns $p_t = 0$, in which case $p_s = \mathcal{R}_{s,t}$.

There is a direct relationship between the analysis of random walks and electrical networks, see for example [LP16]. The relationship between quantum walks and electrical networks was built for the first time by [Bel13], where electrical network theory was used to construct and analyze a phase estimation algorithm to detect whether a given graph contained a marked element. Recently, [Pid19, AP22] have shown that the resulting state after running this phase estimation is a quantum state representing the electrical flow between a starting vertex s and the marked vertices t.

For a network $G = (V, E, \mathsf{w})$ and vertices $s, t \in V$, let

$$\mathcal{H} = \operatorname{span}\{|u, v\rangle \,|(u, v) \in E\}$$

be the associated vector space of its edges. We emphasize here, especially for the readers familiar with [JZ23], that each edge (u, v) appears twice in $\mathcal{H}$: both as $|u, v\rangle$ and $|v, u\rangle$, which are orthogonal states in $\mathcal{H}$. For each vertex $u \in V$, we let $\mathsf{w}_u = \sum_{v \in \Gamma(u)} \mathsf{w}_{u,v}$ be the weighted degree of u. We use it to define the (normalized) *star state* of u as

$$|\psi_u\rangle = \frac{1}{\sqrt{\mathsf{w}_u}} \sum_{v \in \Gamma^+(u)} \sqrt{\mathsf{w}_{u,v}}\, |u, v\rangle - \sum_{v \in \Gamma^-(u)} \sqrt{\mathsf{w}_{u,v}}\, |u, v\rangle = \frac{1}{\sqrt{\mathsf{w}_u}} \sum_{v \in \Gamma(u)} (-1)^{\Delta_{u,v}} \sqrt{\mathsf{w}_{u,v}}\, |u, v\rangle .$$

Here for any $(u, v) \in E$, the quantity $\Delta_{u,v}$ is equal to 0 if $(u, v) \in \overrightarrow{E}$ and 1 if $(v, u) \in \overrightarrow{E}$.

This definition of a star state is slightly different from most of the literature, where there is usually no sign difference depending on whether (u, v) is part of the directed

edge set, but this will be necessary later on when working with the multidimensional quantum walk framework from [JZ23]. Now consider the following two subspaces of $\mathcal{H}$. Let

$$\mathcal{A} := \operatorname{span}\{|\psi\rangle \in \mathcal{H} : \langle u,v|\psi\rangle = -\langle v,u|\psi\rangle \ \ \forall\, |u,v\rangle \in \mathcal{H}\}$$

be the *antisymmetric subspace* of $\mathcal{H}$. Moreover, let $\mathcal{B} := \operatorname{span}\{|\psi_u\rangle : u \in V\backslash\{s,t\}\}$ be the *star space* of $\mathcal{H}$. Then the *quantum walk operator* $U_{\mathcal{AB}}$ is defined as

$$U_{\mathcal{AB}} := (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I), \tag{2.5}$$

where $\Pi_{\mathcal{A}}$ and $\Pi_{\mathcal{B}}$ are orthogonal projectors onto $\mathcal{A}$ and $\mathcal{B}$ respectively. Note that

$$2\Pi_{\mathcal{A}} - I = -\mathsf{SWAP}, \qquad\qquad 2\Pi_{\mathcal{B}} - I = 2\sum_{u\in V\backslash\{s,t\}} |\psi_u\rangle\langle\psi_u| - I,$$

where SWAP acts as $\mathsf{SWAP}\,|u,v\rangle = |v,u\rangle$ for any $|u,v\rangle \in \mathcal{H}$. For any star state $|\psi_u\rangle$, we write

$$|\psi_u^+\rangle := \sqrt{2}(I - \Pi_{\mathcal{A}})\,|\psi_u\rangle = \frac{I + \mathsf{SWAP}}{\sqrt{2}}\,|\psi_u\rangle$$

for its normalized projection onto $\mathcal{A}^\perp$, which is also known as the *symmetric subspace* of $\mathcal{H}$. For any flow θ, we define its associated (normalized) flow state in $\mathcal{H}$ as

$$|\theta\rangle := \frac{1}{\sqrt{2\mathcal{E}(\theta)}} \sum_{(u,v)\in\overrightarrow{E}} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}} \left(|u,v\rangle + |v,u\rangle\right). \tag{2.6}$$

In the case where θ is the s-t electrical flow, we define the (unnormalized) state associated with the induced potential vector p (with the convention that $p_t = 0$) as

$$|p\rangle = \sqrt{\frac{2}{\mathcal{R}_{s,t}}} \sum_{u\in V\backslash\{s\}} p_u\sqrt{\mathsf{w}_u}\,|\psi_u\rangle\,. \tag{2.7}$$

In [Pid19, AP22], this potential state $|p\rangle$ is used to show that by running phase estimation on the quantum walk operator $U_{\mathcal{AB}}$, we can obtain a close approximation to the flow state $|\theta\rangle$. The precision required in this phase estimation algorithm scales with a quantity in [AP22] is defined as the escape time ET_s:

$$\mathsf{ET}_s := \frac{1}{2}\,\||p\rangle\|^2 = \frac{1}{\mathcal{R}_{s,t}} \sum_{u\in V} p_u^2 \mathsf{w}_u.$$

Since we will not be using the operational meaning of ET_s, we will omit ET_s in the rest of this thesis and instead work with $\||p\rangle\|$.

Lemma 2.5.1 (Modified Lemma 8 in [Pid19] and Lemma 10 in [AP22])**.** *Define the unitary $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - 1)(2\Pi_{\mathcal{B}} - 1)$ acting on a Hilbert space $\mathcal{H}$ for projectors $\Pi_{\mathcal{A}}, \Pi_{\mathcal{B}}$ onto some subspaces $\mathcal{A}$ and $\mathcal{B}$ of $\mathcal{H}$ respectively. Let $|\psi\rangle = \sqrt{p}\,|\varphi\rangle + (I - \Pi_{\mathcal{A}})\,|\phi\rangle$ be a normalized quantum state such that $U_{\mathcal{AB}}\,|\varphi\rangle = |\varphi\rangle$ and $|\phi\rangle$ is a (unnormalized) vector satisfying $\Pi_{\mathcal{B}}\,|\phi\rangle = |\phi\rangle$. Then performing phase estimation on the state $|\psi\rangle$ with operator $U_{\mathcal{AB}}$ and precision δ outputs "0" with probability $p' \in [\frac{4}{\pi^2}p, p + \frac{17\pi^2\||\phi\rangle\|}{16T}]$ and $T = \frac{1}{\delta}$, leaving a state $|\psi'\rangle$ satisfying*

$$\frac{1}{2}\,\||\psi'\rangle\langle\psi'| - |\varphi\rangle\langle\varphi|\|_1 \leq \sqrt{\frac{17\pi^4\delta\,\||\phi\rangle\|}{64p}}.$$

Consequently, when the precision is $O\left(\frac{p\epsilon^2}{\||\phi\rangle\|}\right)$, the resulting state $|\psi'\rangle$ satisfies

$$\frac{1}{2}\,\||\psi'\rangle\langle\psi'| - |\varphi\rangle\langle\varphi|\|_1 \leq \epsilon.$$

Proof. See Appendix B.1 . □

The input to the phase estimation algorithm is the state $|\psi\rangle$ and an arbitrary unitary operator U. When the measured phase value is 0, the phase estimation algorithm projects the initial state $|\psi\rangle$ onto the 1-eigenspace of the unitary U. Notably, the result of Lemma 2.12 applies to a broader context beyond network flows.

This lemma is almost equivalent to Lemma 8 in [Pid19] and Lemma 10 in [AP22], but we have modified it slightly as we were unable to verify the constants in [Pid19, AP22] and the scaling with the precision in [AP22]. The theory of electrical networks tells us that if we consider the s-t electrical flow θ, then we can apply Lemma 2.5.1 to approximate the s-t electrical flow state $|\theta\rangle$.

Theorem 2.5.2. *Let $U_{\mathcal{AB}}$ be the quantum walk operator as defined in Eq. (2.5). Then by performing phase estimation on the initial state $|\psi_s^+\rangle$ with the operator $U_{\mathcal{AB}}$ and precision $O\left(\frac{\epsilon^2}{\sqrt{\mathcal{R}_{s,t}\mathsf{w}_s}\|p\|}\right)$, the phase estimation algorithm outputs "0" with probability $\Theta\left(\frac{1}{\mathcal{R}_{s,t}\mathsf{w}_s}\right)$, leaving a state $|\theta'\rangle$ satisfying*

$$\frac{1}{2}\,\||\theta'\rangle\langle\theta'| - |\theta\rangle\langle\theta|\|_1 \leq \epsilon.$$

For completeness, we proved the Theorem 2.5.2 using our unusual definition of a star state. The proof of Theorem 2.5.2 can also be found in [Pid19, AP22]. We remark that it is possible to modify the network G to ensure that $\mathcal{R}_{s,t}\mathsf{w}_s = \Theta(1)$, which is a standard tool used in quantum electrical networks [Bel13].

Proof. Firstly, by Kirchhoff's Law (see Definition 2.5.4), we know the s-t electrical flow θ is conserved at each vertex $u \in V \setminus \{s,t\}$, which shows that $\Pi_{\mathcal{B}} |\theta\rangle = 0$:

$$\begin{aligned}\langle\psi_u|\theta\rangle &= \frac{1}{\sqrt{2\mathcal{R}_{s,t}}} \sum_{v\in\Gamma(u)} (-1)^{\Delta_{u,v}} \sqrt{\mathsf{w}_{u,v}} \langle u,v| \sum_{(u,v)\in\overrightarrow{E}} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}(|u,v\rangle + |v,u\rangle) \\ &= \frac{1}{\sqrt{2\mathcal{R}_{s,t}}} \left(\sum_{v\in\Gamma^+(u)} \theta_{u,v} + \sum_{v\in\Gamma^-(u)} -\theta_{v,u} \right) = \frac{1}{\sqrt{2\mathcal{R}_{s,t}}} \sum_{v\in\Gamma(u)} \theta_{u,v} = 0.\end{aligned} \tag{2.8}$$

By Ohm's Law (see Definition 2.5.5), we know that there exists a potential p, with $p_t = 0$, such that for each edge $(u,v) \in E$ we have $p_u - p_v = \frac{\theta_{u,v}}{\mathsf{w}_{u,v}}$. This shows that $\Pi_{\mathcal{A}} |\theta\rangle = 0$, which combined with the fact that $\Pi_{\mathcal{B}} |\theta\rangle = 0$ shows that $|\theta\rangle$ is indeed a normalized $+1$-eigenvector of $U_{\mathcal{AB}}$:

$$\begin{aligned}|\theta\rangle &= \frac{1}{\sqrt{2\mathcal{R}_{s,t}}} \sum_{(u,v)\in\overrightarrow{E}} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}(|u,v\rangle + |v,u\rangle) \\ &= \frac{1}{\sqrt{2\mathcal{R}_{s,t}}} \sum_{(u,v)\in\overrightarrow{E}} \left(\sqrt{\mathsf{w}_{u,v}}(p_u - p_v)\,|u,v\rangle + (p_u - p_v)\sqrt{\mathsf{w}_{u,v}}\,|v,u\rangle\right) \\ &= \frac{1}{\sqrt{2\mathcal{R}_{s,t}}} \left(\sum_{u\in V} p_u \sum_{v\in\Gamma(u)} (-1)^{\Delta_{u,v}} \sqrt{\mathsf{w}_{u,v}}\,|u,v\rangle + \mathsf{SWAP} \sum_{u\in V} p_u \sum_{v\in\Gamma(u)} (-1)^{\Delta_{u,v}} \sqrt{\mathsf{w}_{u,v}}\,|u,v\rangle \right) \\ &= (I - \Pi_{\mathcal{A}}) \sqrt{\frac{2}{\mathcal{R}_{s,t}}} \sum_{u\in V} p_u \sqrt{\mathsf{w}_u}\,|\psi_u\rangle\,.\end{aligned} \tag{2.9}$$

Not only does Eq. (2.9) tells us that $\Pi_{\mathcal{A}} |\theta\rangle = 0$, it also immediately shows us how to decompose $|\theta\rangle$ to obtain the factor $|p\rangle$, where we make use of the fact that $p_s = \mathcal{R}_{s,t}$:

$$\begin{aligned}|\theta\rangle &= (I - \Pi_{\mathcal{A}}) \sqrt{\frac{2}{\mathcal{R}_{s,t}}} \Big(\sum_{u\in V} p_u \sqrt{\mathsf{w}_u}\,|\psi_u\rangle = (I - \Pi_{\mathcal{A}})\,|p\rangle + (I - \Pi_{\mathcal{A}}) \sqrt{\frac{2}{\mathcal{R}_{s,t}}} p_s \sqrt{\mathsf{w}_s}\,|\psi_u\rangle \\ &= (I - \Pi_{\mathcal{A}})\,|p\rangle + \sqrt{\mathcal{R}_{s,t}\mathsf{w}_s}\,|\psi_s^+\rangle\,,\end{aligned}$$

which we can rewrite to

$$|\psi_s^+\rangle = \frac{1}{\sqrt{\mathcal{R}_{s,t}\mathsf{w}_s}}|\theta\rangle - (I - \Pi_{\mathcal{A}})\frac{1}{\sqrt{\mathcal{R}_{s,t}\mathsf{w}_s}}|p\rangle\,. \tag{2.10}$$

Lastly, since $p_t = 0$, we immediately have $|p\rangle \in \mathcal{B}$ by its definition in Eq. (2.7), meaning $\Pi_{\mathcal{B}}|p\rangle = |p\rangle$. Hence by applying Lemma 2.5.1 with the parameters $|\psi\rangle = |\psi_s^+\rangle$, $|\varphi\rangle = |\theta\rangle$, $|\phi\rangle = -\frac{1}{\sqrt{\mathcal{R}_{s,t}\mathsf{w}_s}}|p\rangle$ and $p = \frac{1}{\mathcal{R}_{s,t}\mathsf{w}_s}$, we find that the resulting state after running phase estimation on the quantum walk operator $U_{\mathcal{AB}}$ with the initial state $|\psi_s^+\rangle$ is approximately the s-t electrical flow state. □

2.6 Conclusion

We have discussed the complexity of four types of quantum algorithms: the QLS algorithm to solve QLSP (Section 2.2), continuous quantum walk (Section 2.3), quantum eigenstate filtering (Section 2.4), and discrete quantum walk based on an electrical network (Section 2.5). In particular, the QLS algorithm serves as the key subroutine for the quantum algorithm in Chapter 3 for solving polynomial systems, while the remaining algorithms form the foundation of quantum algorithms aimed at finding an s-t path in graphs of exponential size, as detailed in Chapter 4.

An interesting problem to explore is how the result of generating a quantum electrical flow state using the phase estimation algorithm can be incorporated into the QSVT framework with the minimax filtering polynomial. This investigation could potentially yield deeper insights and more efficient algorithms for quantum algorithms that rely on electrical network models.

Chapter 3

Quantum Algorithms and Multivariate Polynomial Systems

In this chapter, we investigate the limitations and potentials of a particular approach using the QLS algorithm for finding a Boolean solution of polynomial systems in the non-oracle setting. Chen and Gao [CG22] proposed a new quantum algorithm for Boolean polynomial system solving, motivated by the cryptanalysis of some post-quantum cryptosystems. The key idea of their approach is to apply a QLS algorithm to a Macaulay linear system over $\mathbb{C}$, which is derived from the Boolean polynomial system. The efficiency of their algorithm depends on the condition number of the Macaulay matrix. We give a strong lower bound on the condition number as a function of the Hamming weight of the Boolean solution and show that in many (if not all) cases a Grover-based exhaustive search algorithm outperforms their algorithm. Then, we improve upon Chen and Gao's algorithm by introducing the Boolean Macaulay linear system over $\mathbb{C}$ by reducing the original Macaulay linear system. This improved algorithm could potentially significantly outperform the brute-force algorithm when the Hamming weight of the solution is logarithmic in the number of Boolean variables.

Furthermore, we provide a simple and more elementary proof of correctness for our improved algorithm using a reduction employing the Valiant-Vazirani affine hashing method, and also extend the result to polynomial systems over $\mathbb{F}_q$ improving on subsequent work by Chen, Gao and Yuan [CGY18]. We also suggest a new approach for extracting the solution of the Boolean polynomial system via a generalization of the quantum coupon collector problem [ABC$^+$20a].

3.1 Introduction

Solving systems of multivariate polynomial equations is a fundamental problem that is NP-complete even when the polynomials are restricted over $\mathbb{F}_2$. The problem can be reduced to solving an exponential number of linear equations via the so-called Macaulay matrix, which holds coefficients of linear equations that come from the input polynomials, and multiples of them (multiplying each polynomial by each monomial up to a certain degree). Each monomial is represented by a new variable, recasting the polynomial equations and their multiples as linear equations. The usual classical approach to solve a polynomial system is based on computing the Gröbner basis of the corresponding polynomial ideal by triangularizing the Macaulay matrix. There is a vast literature on characterizing and improving the complexity of solving various types of polynomial systems using the Macaulay matrix [AFI+04, CG17, CKPS, DS13, Die04, Per16, WW15].

In quantum computing, the HHL [HHL09] QLS algorithm outputs a quantum state $|x\rangle$ such that $\mu A |x\rangle = |b\rangle$ for an exponentially large matrix A with certain properties, and a quantum state $|b\rangle$, in time $\widetilde{\mathcal{O}}(\kappa^2 \mathsf{s}^2)$,[1] where κ is the *condition number* of A, μ is a normalization factor, and s is the sparsity of the matrix A, while state-of-the-art QLS algorithms [Amb12, CKS17, CGJ19, GSLW19, SSO19, LT19a] have complexity $\widetilde{\mathcal{O}}(\kappa \mathsf{s})$. Although, the QLS algorithm is BQP-complete [HHL09], meaning that it captures all essential features of quantum computing, a natural "killer-application" is still to be discovered – showing the difficulty of finding a practically interesting problem instance that satisfies all stringent conditions. For example, to efficiently solve the classical equation $Ax = b$ using the original QLS algorithm, where implicit access is given to an exponentially large matrix A and b, the following must be satisfied: the state $|b\rangle$ can be efficiently prepared, the sought data can be efficiently extracted from the output state $|x\rangle$, and the matrix A should be sparse and well-conditioned [Aar15].

Chen and Gao [CG22] made an interesting connection between the exponential size Macaulay matrix and the QLS algorithm. While they use Gröbner bases in their proof of correctness, they do not explicitly compute the Gröbner basis and instead use the QLS algorithm to solve the exponentially large system of linear equations resulting from the Macaulay matrix. They show that the access requirements that usually cause so much trouble, can all be resolved for this application, namely: they can efficiently compute the entries of an appropriate sparse matrix A, prepare $|b\rangle$, and extract the answer from $|x\rangle$. However, a major question was left open: what is the condition number of the matrices,

[1] We denote $\mathcal{O}(T \cdot \operatorname{poly}\log(T) \cdot \operatorname{poly}(1/\varepsilon))$ by $\widetilde{\mathcal{O}}(T)$, where ε is the required precision of the solution.

driving the running time? Intuitively, for worst case instances of polynomial systems, the condition number of the resulting matrix should be large because the approach can solve NP-complete problems. This being said, the analysis of the condition number was left open, both in general and for special cases such as breaking cryptosystems, which have distributions over specific problem instances that might be easier than the worst case. Therefore, the algorithm of Chen and Gao [CG22] together with the follow-up work of Chen, Gao, and Yuan [CGY18] presented a potential quantum threat on multivariate cryptosystems. However, there was no consensus on the strength of this potential quantum attack, as its cryptanalysis was wide open.

We prove an exponential lower bound on the condition number κ of the matrix A related to the Boolean polynomial system, which shows that the quantum algorithm in [CG22] takes exponential time in the worst case. We also give a Grover-based brute-force search algorithm that outperforms their quantum algorithm for solving Boolean polynomial systems when there is a unique solution or all solutions have the same Hamming-weight. Specifically, in the unique solution case we give a simple proof that the condition number κ is $\Omega((3n)^{h/2})$, where h is the Hamming weight of the solution to the original n-variable Boolean polynomial system. Meanwhile, a simple Grover-based brute-force search algorithm over the possible assignments to the variables takes time $\mathcal{O}\left(\sqrt{\binom{n}{h}}\right)$, where $\sqrt{\binom{n}{h}} \leq \left(\frac{en}{h}\right)^{h/2} \leq (3n)^{h/2}$.

In fact, we give "robust" lower bounds on the condition number by also considering "truncated" QLS algorithms [HHL09, GSLW19]. Namely, if the singular-values of A are only inverted on a *well-conditioned* subspace and the overlap of the solution x with such a subspace is large enough, then a "truncated" QLS algorithm can provide a sufficiently accurate solution $\tilde{x}$. In order to give a bound on the performance of such "truncated" versions of the QLS algorithm, we define the concept of the *truncated QLS condition number* $\kappa_b(A) := \|A\| \, \|A^+ b\| \, / \, \|b\|$,[2] which is also a lower bound on $\kappa = \|A\| \, \|A^+\|$. All of our lower bounds also apply to the truncated QLS condition number, ruling out further improvement by truncated QLS algorithms. These results provide strong evidence that the quantum algorithm of [CG22] (at least in its original form) does not present a fatal cryptanalytic threat, and give generic tools for analyzing the strength of individual cryptosystems against this type of quantum attack.

Finally, we refine Chen and Gao's algorithm to the point that our lower bound does not always rule out the possibility of a superpolynomial quantum speedup even

[2] A^+ stands for the (Moore-Penrose) pseudoinverse of A, and $\|\cdot\|$ for the ℓ_2 norm of vectors and for the corresponding induced operator norm, i.e., the spectral norm.

for unique solutions. In particular, the lower bound changes from $(3n)^{h/2}$ to $2^{h/2}$ on our refined algorithm, so for $h = \Theta(\log n)$ the lower bound is only a polynomial, while the brute-force algorithm takes quasi-polynomial time. Thus, it is conceivable that the condition number is upper bounded by poly(n) for some set of interesting input equations, potentially yielding a superpolynomial speedup. We leave it open to find a problem instance whose associated Macaulay matrix has a small enough condition number so that the running time of our refined quantum algorithm gives a speedup over the best classical or Grover-based algorithm. Such an example could result in a new type of quantum speedup and one that uses the QLS algorithm in a novel way.

The core ingredient of our refined algorithm is to show that the Macaulay matrix can be simplified to what we call the Boolean Macaulay matrix over $\mathbb{C}$ by exploiting that the input consists of quadratic polynomials over $\mathbb{C}$, but restricted to 0/1 solutions. The Boolean Macaulay matrix is a submatrix of the original Macaulay matrix that can be obtained via Gaussian elimination over $\mathbb{C}$. This construction of the Boolean Macaulay matrix over $\mathbb{C}$ is different from the Boolean Macaulay matrix over $\mathbb{F}_2$ as defined in [BFSS13] since they are over different fields. This matrix preserves the solution set while its size is much smaller compared to the original Macaulay matrix – ultimately leading to a smaller lower bound $\Omega(2^{h/2})$ on the condition number.

The correctness of our refined algorithm can be shown via the equivalence between the Boolean Macaulay linear system and the Macaulay linear system, where the correctness of the latter has been proven in [CG22]. For completeness, we also provide a simple self-contained proof of correctness for our improved algorithm in Appendix A.1, which is more elementary than that of the original algorithm proposed by Chen and Gao [CG22], since our proof does not require Gröbner bases. Instead, our proof combines a special case of the reduction in [CGY18] with the Valiant-Vazirani affine hashing method, reducing any Boolean polynomial system with more than one solution to one that has a unique solution. For a Boolean Macaulay linear system that has a unique solution, we also provide an alternative approach for extracting the Boolean solution of the corresponding Boolean polynomial system from the output quantum state, i.e., the normalized monomial solution vector of the Boolean Macaulay linear system. Specifically, we reformulate this problem as a generalization of the quantum coupon collector problem, and prove that $O(\log n)$ iterations suffice for extracting a solution, whereas Chen and Gao's algorithm uses $O(n)$ iterations. On the other hand, the affine hashing reduction introduces $\mathcal{O}(n)$ extra rounds, so the total number of iterations in our algorithm can be bounded as $\mathcal{O}(n \log n)$.

3.1.1 Quantum algorithms for solving polynomial systems

Chen and Gao [CG22] proposed using the QLS algorithm to solve a Boolean polynomial system via solving an exponentially large linear system of equations. Now we discuss the two main parameters that appear in the complexity of the QLS algorithm and their relevance in our case.

$\kappa_b(A)$: The *truncated QLS condition number* (tQLScn) $\kappa_b(A)$ of the QLSP $Ax = b$ is an important parameter related to the time-complexity of "truncated" QLS algorithms.[3] (For simplicity let us assume without loss of generality that $\|A\| \leq 1$ and $\|b\| = 1$.) We use a simple Markov-type inequality showing that inverting A via a truncated variant of the QLS algorithm, with (condition number) truncation much below $\kappa_b(A)$, must give a highly inaccurate solution. Indeed, let S be a subspace, which is spanned by right singular vectors of A that have singular value at least $c/\kappa_b(A)$ for some $c > 1$. Let Π_S be the orthogonal projector on S, then $\|\Pi_S x\| = \|\Pi_S A^+ b\| \leq \|\Pi_S A^+\| \, \|b\| \leq \kappa_b(A)/c$. On the other hand $\|x\| = \|A^+ b\| \geq \|A\| \, \|A^+ b\| \, / \, \|b\| = \kappa_b(A)$, thereby the overlap of x with the subspace S must be relatively small for large c. This argument shows, that giving a lower bound on the truncated QLS condition number makes our results stronger, as it does not only bound the complexity of standard QLSP solvers, but also lower-bounds the complexity of fine-tuned truncated variants.

s : In order to efficiently solve the Quantum Linear System Problem (QLSP), we need a succinct representation of the vector b and the matrix A. In our case both b and A are s-sparse for some polynomially large s, i.e., they have at most s nonzero entries in every column and row. In order to utilize their sparsity, we also need to be able to efficiently compute the locations and the values of the nonzero entries; this is easy to do in our case since the vector b is sparse and the (Boolean) Macaulay matrix A has a quasi-Toeplitz structure.

More precisely, it suffices to have efficient (quantum) circuits computing the locations and values of the nonzero elements of A, allowing us to perform the transformations

$$|i, k\rangle \longrightarrow |i, \bar{c}(i, k)\rangle \tag{3.1}$$

[3] Note that the truncated QLS condition number gives a lower bound on the performance of truncated QLS alorithms, but it does not characterize their complexity, i.e., there might not exist any truncated QLS algorithm with complexity matching the truncated QLS condition number.

$$|i,j,0\rangle \longrightarrow |i,j,\bar{A}_{ij}\rangle\,, \tag{3.2}$$

where i labels the row indices of the matrix $\bar{A}$ standing for A and A^T, k labels the nonzero entries of $\bar{A}$ (which is assumed to be s-sparse), $\bar{c}(i,k)$ represents the column index of the k-th nonzero entry of the matrix $\bar{A}$ in row i, and $|0\rangle$ in Eq. (3.2) represents a (large enough) ancillary system in which the matrix element $\bar{A}_{ij}$ can be stored (as a bit-string with sufficient precision so that errors can be neglected). Note that the transformations Eq. (3.1)-Eq. (3.2) are essentially only used to build an efficient quantum circuit U_A [GSLW18, Lemma 48], which implements a unitary U, such that the top left $N \times M$ corner of U equals A – such a unitary is called a block-encoding of A [CGJ19, GSLW19] (also see Section 2.1). Similarly, the sparsity assumption for b is only used in order to build an efficient quantum circuit C_b for preparing the quantum state $|b\rangle := \left(\sum_{i=1}^{N} b_i\,|i\rangle\right) / \left\|\sum_{i=1}^{N} b_i\,|i\rangle\right\|$.

For completeness, let us mention two additional types of quantum algorithms that have been proposed for solving polynomial systems:

1. QAOA: In 2002, Burges [Bur02] reformulated RSA factoring problem as a polynomial system solving that has a unique solution, which is a special case of Problem 3.2.2. Later, Anschuetz, Olson, Aspuru-Guzik and Cao [AOAGC18] reformulated the polynomial system solving problem as a Local Hamiltonian Problem (LHP) that has a corresponding unique ground state, and they applied the QAOA algorithm for finding this ground state; the exact complexity of this algorithm is unknown.

2. Grover: In 2017, Faugère, Horan, Kahrobaei, Kaplan, Kashefi, and Perret [FHK$^+$17] presented a quantum version of the classical algorithm in [BFSS13], that applies Grover search on both the exhaustive search and the consistency check subroutines. Under certain assumptions, the running time of this quantum algorithm is slightly better than $\mathcal{O}\left(2^{n/2}\right)$, which is the running time of the trivial Grover search algorithm. Similarly, Bernstein and Yang [BY18] gave a quantum algorithm (GroverXL) for random polynomial systems over a finite field $\mathbb{F}_q$.[4]

[4]The generalization of the Boolean Macaulay matrix method in [BFSS13] is equivalent to the reduced XL approach that first appeared in [CKPS] and then defined in [Die04].

3.2 Reducing polynomial system solving over a finite field $\mathbb{F}_q$ to polynomial system solving over $\mathbb{C}$

First, we present the $\mathbb{F}_q = \mathbb{F}_2$ special case of Chen, Gao and Yuan's approach [CGY18]. In this special case, there is a bijection between the solution sets of the corresponding polynomial systems over $\mathbb{F}_2$ and $\mathbb{C}$.

Problem 3.2.1. *Solve a system of n-variate quadratic polynomials with Boolean variables over $\mathbb{F}_2$.*

Input : *$\mathcal{F} = \{f_1, \ldots, f_m\} \subseteq \mathbb{F}_2[x_1, \ldots, x_n]$ with $\deg(f_i) \leq 2$ for $i = 1, 2, \ldots, m$.*

Output : *a solution $s \in \mathbb{F}_2^n$ such that $f_1(s) = \cdots = f_m(s) = 0$, when one exists.*

Problem 3.2.2. *Solve a system of n-variate quadratic polynomials over $\mathbb{C}$, together with the $\mathbb{F}_2$ field equations that force variables to be Boolean.*

Input : *$\mathcal{F} \subseteq \mathbb{C}[x_1, \ldots, x_n]$ where $\mathcal{F} = \{f_1, \ldots, f_m\}$ with $\deg(f_i) = 2$ for $i = 1, \ldots, m$.*

Output : *an $s \in \{0,1\}^n$ such that $f_1(s) = \cdots = f_m(s) = 0$ over $\mathbb{C}$, when one exists.*

Let $\#\mathcal{F}$ denote the maximum number of nonzero terms in any polynomial in $\mathcal{F}$. Also let $\#f$ be a shorthand for $\#\{f\}$.

Lemma 3.2.3. *There is a polynomial-time reduction from Problem 3.2.1 on n variables and a set of m equations $\mathcal{F}$ to Problem 3.2.2 on $n + m \cdot \lfloor \log_2 \#\mathcal{F} \rfloor$ variables and $n + m \cdot (\lfloor \log_2 \#\mathcal{F} \rfloor + 1)$ equations.*

Proof. Solving Problem 3.2.1 is equivalent to solving the following polynomial system in variables $x_1, \ldots, x_n$, $z_1, \ldots, z_m$ over $\mathbb{C}$:

$$\forall i \in [m]: \qquad f_i(x_1, \ldots, x_n) - z_i = 0, \tag{3.3}$$

$$\forall i \in [m]: \qquad z_i/2 \in \mathbb{Z}, \tag{3.4}$$

$$\forall j \in [n]: \qquad x_j^2 - x_j = 0. \tag{3.5}$$

The field Eq. (3.5) forces each x_j to be 0 or 1, and therefore each f_i evaluates to an even or odd integer. In addition, it is easy to see that each integer z_i in Eq. (3.4) is in $[0, \#f_i]$ for $i = 1, \ldots m.$ and can be treated as a polynomial. Eq. (3.3)-Eq. (3.4) force f_i to evaluate to an even integer, so it is 0 mod 2.

Chen, Gao and Yuan [CGY18] then represent each $z_1, \ldots, z_m$ by the bits in its binary expansion. For each variable $z_i \in [0, \#f_i]$ a polynomial is introduced with Boolean variables y_{ib} to represent its value in binary, i.e., $z_i = \sum_{b=1}^{\lfloor \log_2 \#f_i \rfloor} 2^b y_{ib}$, and $y_{ib}^2 - y_{ib} = 0$ for $b = 1, \ldots, \lfloor \log_2 \#f_i \rfloor$.

Substituting the polynomials and Boolean constraints corresponding to each z_i into the polynomial Eq. (3.3)-Eq. (3.5), we get a following polynomial system $\mathcal{F}$ over $\mathbb{C}$:

$$\forall i \in [m] \colon f_i(x_1, \ldots, x_n) - \sum_{b=1}^{\lfloor \log_2 \#\mathcal{F} \rfloor} 2^b y_{ib} = 0, \tag{3.6}$$

$$\forall i \in [m] \forall b \in [\lfloor \log_2 \#\mathcal{F} \rfloor] \colon y_{ib}^2 - y_{ib} = 0, \tag{3.7}$$

$$\forall j \in [n] \colon x_j^2 - x_j = 0. \tag{3.8}$$

It is easy to see that there is a bijection between the set of solutions of $\mathcal{F} \subseteq \mathbb{F}_2[x_1, x_2 \ldots, x_n]$ and the set of solutions of Eq. (3.6)-Eq. (3.8) over $\mathbb{C}$. On one hand, given a solution $(s_1, \ldots, s_n)$ of $\mathcal{F} \subseteq \mathbb{F}_2[x_1, x_2 \ldots, x_n]$, evaluating $f_i(s_1, \ldots, s_n)$ over $\mathbb{C}$ gives an even number z_i, and its binary expansion gives the values of the y_{ib} variables. On the other hand, let $(s_j), (t_{ib})$ be a solution to Eq. (3.6)-Eq. (3.8). For each j, $s_j \in \{0, 1\}$ by Eq. (3.8), and for each i, $f_i(s_1, \ldots, s_n) = \sum_{b=1}^{\lfloor \log_2 \#\mathcal{F} \rfloor} 2^b y_{ib}$ by Eq. (3.6). Due to Eq. (3.7) this implies that $f_i(s_1, \ldots, s_n) \equiv 0 \mod 2$, so (s_j) is a solution of $\mathcal{F}$. □

Given a set of polynomials $\mathcal{F} \subseteq \mathbb{F}_q[X]$, Chen, Gao and Yuan [CGY18] propose an analogous approach for reducing the polynomial system $\mathcal{F}$ to a polynomial system $\mathcal{F}_\mathbb{C} = 0$, where $\mathcal{F}_\mathbb{C} \subseteq \mathbb{C}[X, Y]$, in the form of Problem 3.2.2. However, in general even if $\mathcal{F}$ has a unique solution, the constructed polynomial systems $\mathcal{F}_\mathbb{C}$ may have multiple solutions.

Now we present two additional reduction steps that make the polynomial systems in Problem 3.2.2 easier to handle:

Red1: In order to ensure that the polynomial system $\mathcal{F} \subseteq \mathbb{C}[X]$ in Problem 3.2.2 has no more than one solution, we employ the Valiant-Vazirani affine hashing method [VV86]. Suppose that the polynomial system $\mathcal{F}$ has $S \in [2^n]$ different solutions. The main idea of the affine hashing method is the following [BKW19]: if one introduces $\lfloor \log_2(S) \rfloor + 2$ random linear equations $\mathcal{F}_R$ with $\mathcal{F}_R \subseteq \mathbb{F}_2[X]$, then they isolate a unique solution with probability at least $\frac{1}{8}$. Even if we don't know the number of solutions a priori, we can loop over all possible values of $\lfloor \log_2(S) \rfloor \in \{0, 1, \ldots, n\}$; making $\mathcal{O}(\ln(1/\varepsilon))$ trials for all possible choices of $\lfloor \log_2(S) \rfloor$ gives at least one system $\mathcal{F}_R = 0$ with a unique solution with probability at least $1 - \varepsilon$.

This amounts to $\mathcal{O}(n\log(1/\varepsilon))$ different polynomial systems to check. Remember that $\mathcal{F}_R \subseteq \mathbb{F}_2[X]$ whereas $\mathcal{F} \subseteq \mathbb{C}[X]$. By Lemma 3.2.3, we can reduce the new polynomials $\mathcal{F}_R$ to polynomials $\mathcal{F}_{R\mathbb{C}} \subseteq \mathbb{C}[X, Y]$, where Y is the set of new variables introduced during the reduction. Finally, if $\mathcal{F}_R$ isolated a unique solution of $\mathcal{F}$, then the polynomial system $\mathcal{F}_{R\mathbb{C}} \cup \mathcal{F}$ has a unique solution. Thus, without loss of generality, we can always assume that Problem 3.2.2 has a unique solution.

Red2: Any polynomial system $\mathcal{F} = \{f_1, f_2, \ldots, f_m\} \subseteq \mathbb{C}[X]$ can be rewritten as $\mathcal{F}' = \{f_1', f_2', \ldots, f_m'\}$, where f_1' has constant term -1, while $f_2', \ldots, f_m'$ have no constant terms. In case no polynomial in $\mathcal{F}$ has a constant term, the all-zero vector is a trivial solution. Otherwise, let c_i denote the constant term of f_i, and let us assume without loss of generality that $c_1 \neq 0$. Then we can simply set $f_1' := -f_1/c_1$, and $f_i' := f_i + c_i f_1'$ for all $i \in \{2, 3, \ldots, m\}$.

The above two reductions increase the parameters considered in this paper only moderately. Indeed, Red1 introduces at most $\mathcal{O}(n\log(n))$ new equations and variables, while Red2 only affects the number of nonzero terms in the polynomial system. Moreover, Red2 increases $\#\mathcal{F}$ and the total number of nonzero terms $\sum \#f_i$ by at most a factor of 2 (for the latter we shall choose f_1 to be the polynomial with a nonzero constant term that also has the fewest nonzero coefficients).

3.3 Macaulay linear systems and their tQLScn

In this section, we define the Macaulay linear system of a set of polynomials $\mathcal{F} \subseteq \mathbb{C}[x_1, \ldots, x_n]$ and show that when $\mathcal{F}$ has a unique solution, the condition number of the matrix is $\Omega(\sqrt{\binom{n}{h}})$, where h is the Hamming weight of the solution. We show that the lower bound also holds when using max degree instead of total degree in the definition, as was done in [CG22], showing that their proposed quantum algorithm for solving polynomial equations by using the QLS algorithm to solve a Macaulay linear system in general takes time $\Omega((3n)^{h/2})$. We also show that if there are t different solutions, but they have the same Hamming weight h, then the above lower bound can reduce by at most a factor of $\sqrt{t}$. Finally, we give a formula that can be used for giving a lower bound on the condition number for any number of solutions and present computational evidence that this analytical lower bound is exponentially large in terms of the smallest Hamming weight among the solutions.

3.3.1 Macaulay linear systems

There is a well-known approach for solving polynomial systems by linearizing them with the help of introducing new latent auxiliary variables. The advantage of this approach is that the problem becomes linear, but the downside is that the new problem is exponentially large. The matrix of the resulting linear system is called the Macaulay matrix.

Definition 3.3.1. *The* Macaulay matrix $\hat{\mathcal{M}}$ of degree d *of* $\mathcal{F} = \{f_1, \ldots, f_m\} \subseteq \mathbb{C}[X]$ *is the matrix where each row is labeled by a pair of polynomials* $(\hat{m}, f)$ *and contains the corresponding coefficient vector of the polynomial* $\hat{m}f$*. The rows ranges over all* $f \in \mathcal{F}$ *and monomials* $\hat{m}$ *such that* $\hat{m}f$ *has degree at most* d*. The columns are labeled by the set of monomials in* $x_1, \ldots, x_n$ *of degree at most* d *and are ordered with respect to a specified monomial ordering. The element in the row corresponding to* $(\hat{m}, f)$ *and the column corresponding to the monomial* $\hat{m}'$ *is the coefficient of* $\hat{m}'$ *in the polynomial* $\hat{m}f$*.*

In the above definition, one can interpret the degree as either the total degree or the max degree (the maximum degree of any variable) of multivariate polynomials resulting in different notions of the Macaulay matrix. When it is necessary we will always clarify which definition is being used. For example, [CG22] uses the max degree, so all references to that paper refer to the max degree version of this definition.

In the classical setting, the goal is to compute the Gröbner basis from the Macaulay matrix, where the Gröbner basis is a set of polynomials $\mathbb{G} = \{g_1, g_2, \ldots, g_r\}$ such that for the leading term of any polynomial f in the ideal $I = (\mathcal{F})$, there exists a polynomial $g_k \in \mathbb{G}$ such that $LM(g_k)|LM(f)$ [5]. Note that the size of the Macaulay matrix depends on the selected degree. For a set of m quadratic polynomials with n variables, the degree is approximately lower bounded by $\frac{n}{\sqrt{m}}$ [CKPS]. When $m = \alpha n$, the degree is upper bounded by $c_\alpha n$ for some constant c_α [BFSS13]. In the quantum setting, the goal is to compute the monomials up to a certain degree. In this paper, we only provide the upper bound of the degree that applies to any quadratic polynomials. It might be interesting to check the degree of some special polynomial systems.

Row operations on the matrix $\hat{\mathcal{M}}$ correspond to polynomial addition, subtraction, and scalar multiplication in the polynomial ideal $\langle \mathcal{F} \rangle$ [Bat13, Buc18], and these operations preserve the common roots of the system. Classically, Gaussian elimination can be performed on this matrix, and the entries can be read out from the row-reduced matrix [Die04, CKPS]. However, in the quantum case, we cannot directly do Gaussian elimination on this matrix and look at the row-reduced matrix. Instead, Chen and Gao

[5] Here $LM(f)$ is the leading term of the polynomial f

showed that the QLS algorithm can be used for sampling from nonzero solutions of the following related linear systems.

Definition 3.3.2. *Let $\hat{\mathcal{M}}$ be the Macaulay matrix of a given polynomial system, with the last column $-\vec{b}$ corresponding to the constant terms of the polynomials. Let $\hat{\mathcal{M}} = [\mathcal{M}| \ -\vec{b}\,]$. Then the equation $\mathcal{M}\vec{y} = \vec{b}$ is called the* Macaulay linear system.

In other words, the Macaulay matrix is the augmented matrix from the Macaulay linear system $\mathcal{M}\vec{y} = \vec{b}$. Due to the reduction Red2 in the previous section, it can be assumed that exactly one of the input polynomials has a nonzero constant term. So we may assume without loss of generality that $\vec{b} = [1 \quad \mathbf{0}]^T$, where the vector $\vec{b}$ corresponds to the column vector indexed by the degree 0 monomial 1.

Chen and Gao's algorithm applies the QLS algorithm to output the quantum state $|\hat{y}\rangle$ that can be measured in order to sample from monomials with nonzero values in a valid assignment corresponding to a solution. We will lower bound the condition number of $\mathcal{M}$, which will in turn lower bound the running time of the proposed algorithm.

Chen and Gao proposed to set the max degree to $3n$ in the Macaulay linear system, and they showed with this choice if a set of polynomials $\mathcal{F}$ has a unique solution, then the linear system also has a unique solution [CG22, Lemma 4.1]. The output state $|\hat{y}\rangle$ of the QLS algorithm then corresponds to this unique solution of the linear system, and the solution of $\mathcal{F}$ can be efficiently obtained from measuring the state $|\hat{y}\rangle$.

Classically, the solving degree of $\mathcal{F}$ is used, which is at most $n + 2$ as shown by Caminata and Gorla [CG17, Theorem 3.26]. This allows computing the Gröbner basis of the polynomial ideal $\langle\mathcal{F}\rangle$ via Gaussian elimination of the linear system, and the solution of $\mathcal{F}$ can be obtained from the Gröbner basis. However, for some polynomial systems, the affine subspace of all solutions of the linear system has no well-understood structure even though $\mathcal{F}$ has a unique solution. In this case, the QLS algorithm outputs a state $|\hat{y}\rangle$ that corresponds to the smallest ℓ_2-norm solution of the linear system. In general, we don't know how to extract the solution of $\mathcal{F}$ from such states $|\hat{y}\rangle$.

When $\mathcal{F}$ has more than one solution and the max degree is set to be $3n$, the dimension of the affine subspace of all solutions of the linear system equals the number of solutions of $\mathcal{F}$ minus 1. For each solution $a \in \{0,1\}^n$ of $\mathcal{F}$, there is a corresponding solution $\hat{y}_a$ of the linear system. Those solutions $\hat{y}_a$ are linearly independent and any solution of the linear system is an affine combination of the solutions $\hat{y}_a$ [CG22, Lemma 3.18]. Again, the QLS algorithm outputs a state $|\hat{y}\rangle$ corresponding to the smallest ℓ_2 norm vector $\hat{y}$ in this affine subspace.

Chen and Gao [CG22] showed that $\mathcal{M}$ is an $\mathcal{O}(m \cdot \#\mathcal{F})$-sparse, row computable matrix and $\vec{b}$ can be efficiently prepared as a quantum state. In particular, assuming $|\mathcal{F}| = \mathcal{O}(\text{poly}(n))$, they show that the QLS algorithm can be run in time $\widetilde{\mathcal{O}}(\text{poly}(n)\kappa(\mathcal{M}))$. There is strong complexity theoretic evidence that in general running the QLS algorithm requires time $\widetilde{\Omega}(\kappa(\mathcal{M}))$ [HHL09], so a lower bound on the condition number also lower bounds the running time.

3.3.2 Lower bound on the truncated QLS condition number $\kappa_{\vec{b}}(\mathcal{M})$

In this section, we give a lower bound on the tQLScn $\kappa_{\vec{b}}(\mathcal{M})$. Since $\kappa_{\vec{b}}(\mathcal{M}) \leq \kappa(\mathcal{M})$, this also implies a lower bound on the time complexity of Chen and Gao's [CG22] algorithm.

In order to prove a lower bound on $\kappa_{\vec{b}}(\mathcal{M})$, it suffices to lower bound the length of the solution vector $\vec{y} = \mathcal{M}^+\vec{b}$, since

$$\kappa_{\vec{b}}(\mathcal{M}) = \|\mathcal{M}\| \frac{\left\|\mathcal{M}^+\vec{b}\right\|}{\left\|\vec{b}\right\|} \geq \left\|\mathcal{M}^+\vec{b}\right\| = \|\vec{y}\|. \tag{3.9}$$

Here, the first equality is the definition of $\kappa_{\vec{b}}$, and the inequality follows from $\left\|\vec{b}\right\| = 1$, and because $\|\mathcal{M}\| \geq 1$, as $\mathcal{M}$ has at least one matrix element which has the absolute value at least 1.[6]

In order to understand the length $\|\vec{y}\|$ of the solution vector $\vec{y} = \mathcal{M}^+\vec{b}$, let us first study the monomial solution vector $\vec{y}^{(a)}$ corresponding to a binary solution a. For a degree-d Macaulay linear system, a monomial exponent $0 \neq e \in \mathbb{N}^n$ is a "valid" coordinate of $\vec{y}^{(a)}$ if $e \in \{0, 1, \ldots, d\}^n$ (and $\sum e_i \leq d$ for total degree), moreover the solution vector satisfies $\vec{y}_e^{(a)} = \Pi_i a_i^{e_i}$, which is 1 if and only if $a_i = 1$ for all variables x_i in the monomial $\Pi_i x_i^{e_i}$ indexed by e. If the Hamming weight of a is h and $\mathcal{M}$ is constructed with max degree, then the number of such non-zero coordinates (monomials) is $(d+1)^h - 1$, thus

$$\left\|\vec{y}^{(a)}\right\|^2 = (d+1)^h - 1. \tag{3.10}$$

When $\mathcal{M}$ is constructed with the total degree, the number of such non-zero coordinates

[6]For the Macaulay matrix construction, let f be $x^2 - x$, the row $(1, f)$ will have a matrix element of magnitude 1.

(monomials) is $\binom{d+h}{h} - 1$, so

$$\left\|\vec{y}^{(a)}\right\|^2 = \binom{d+h}{h} - 1. \tag{3.11}$$

Suppose $a_1, a_2, \cdots, a_t \in \{0,1\}^n$ are the t solutions of $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$, where $\mathcal{F}_1 = \{f_1, \ldots, f_m\}$ and $\mathcal{F}_2 = \{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$. The t solutions $a_1, a_2, \cdots, a_t$ of $\mathcal{F}$ must be nonzero because the first equation has constant term $b_1 = 1$. Let $\vec{y}_1, \vec{y}_2, \cdots, \vec{y}_t$ [7] be the 0/1 solution vectors of the linear system $\mathcal{M}\vec{y} = \vec{b}$ under the assignments $a_1, a_2, \cdots, a_t$ respectively. When the max degree of the linear system is set to be $3n$, the affine subspace of all solutions of the linear system is spanned by the monomial solution vectors $\vec{y}_1, \vec{y}_2, \cdots, \vec{y}_t$ [CG22, Theorem 3.21 and Lemma 4.1], but this property might also hold for lower degrees. From now on we assume that degree d is such that the linear system has this property, then $\vec{y} = \mathcal{M}^+\vec{b}$ has the minimum ℓ_2 norm in the affine subspace spanned by $\vec{y}_1, \vec{y}_2, \cdots, \vec{y}_t$.

If all the t solutions $a_1, a_2, \cdots, a_t \in \{0,1\}^n$ have the same Hamming-weight, we can lower bound the length $\|\vec{y}\|$ of the solution vector $\vec{y} = \mathcal{M}^+\vec{b}$ by the following lemma.

Lemma 3.3.1. *Suppose $\vec{y}_1, \vec{y}_2, \cdots, \vec{y}_t$ are vectors with 0 and 1 entries such that their Hamming weights are equal. Then every vector in their (complex) affine hull A has length (ℓ_2 norm) at least $\|\vec{y}_1\| / \sqrt{t}$.*

Proof. Every entry of the vectors $\vec{y}_i$ is either 0 or 1, therefore $\langle \vec{y}_i, \mathbf{1} \rangle = \|\vec{y}_i\|_1$ (we denote by $\mathbf{1}$ the all-1 vector). Since the vectors have the same Hamming weight we also have $\|\vec{y}_1\|_1 = \|\vec{y}_i\|_1$ for all $i \in [t]$. Let $\vec{y}$ be any vector in A, then $\langle \vec{y}, \mathbf{1} \rangle = \langle \vec{y}_1, \mathbf{1} \rangle$, and in particular $\|\vec{y}\|_1 \geq \langle \vec{y}, \mathbf{1} \rangle = \|\vec{y}_1\|_1 = \|\vec{y}_1\|_2^2$. Let $\|\vec{y}\|_0$ denote the support size of $\vec{y}$. Then $\|\vec{y}\|_0 \leq \sum_{i=1}^t \|\vec{y}_i\|_0 = t\,\|\vec{y}_1\|_0 = t\,\|\vec{y}_1\|_1 = t\,\|\vec{y}_1\|_2^2$. By the Cauchy-Schwarz inequality, we have that $\|\vec{y}\|_1 \leq \|\vec{y}\|_2 \sqrt{\|\vec{y}\|_0} \Longrightarrow \|\vec{y}\|_2 \geq \frac{\|\vec{y}\|_1}{\sqrt{\|\vec{y}\|_0}} \geq \frac{\|\vec{y}_1\|_2^2}{\sqrt{t}\|\vec{y}_1\|_2} = \frac{\|\vec{y}_1\|_2}{\sqrt{t}}$. □

If the minimum ℓ_2-norm solution $\vec{y} = \hat{\mathcal{M}}^+\vec{b}$ happens to be a convex combination $\vec{y} = \sum_{i=1}^t w_i\vec{y}_i$ of the (possibly differing Hamming weight) solution vectors $\vec{y}_1, \vec{y}_2, \cdots, \vec{y}_t$, then we can similarly lower bound the length $\|\vec{y}\|$.

Lemma 3.3.2. *Suppose $\vec{y}_1, \vec{y}_2, \cdots, \vec{y}_t$ are 0/1 vectors and $\vec{y}_1$ has the minimum Hamming weight. Then every vector in their convex hull A has length (ℓ_2-norm) at least $\|\vec{y}_1\| / \sqrt{t}$.*

[7] Note that the monomial solution vector corresponding to a binary solution a_i is $\vec{y}^{a_i}$. Here and in the following discussion of multiple solutions case, we write $\vec{y}^{a_i}$ as $\vec{y}_i$ for simplicity.

Proof. Let $\vec{y} = \sum_{i=1}^{t} w_i \vec{y}_i$ be an arbitrary vector in the convex hull A generated by $\vec{y}_1, \vec{y}_2, \cdots, \vec{y}_t$, where $\sum_{i=1}^{t} w_i = 1$ and $w_i \geq 0$. Then $\|\vec{y}\|^2 = \sum_{i=1}^{t} \sum_{j=1}^{t} w_i w_j \langle \vec{y}_i, \vec{y}_j \rangle \geq \sum_{i=1}^{t} w_i^2 \langle \vec{y}_i, \vec{y}_i \rangle \geq \sum_{i=1}^{t} w_i^2 \langle \vec{y}_1, \vec{y}_1 \rangle \geq \langle \vec{y}_1, \vec{y}_1 \rangle / t$. The first equality is by the definition of $\|\vec{y}\|^2 = \langle \vec{y}, \vec{y} \rangle$. The first inequality is because $w_i w_j \langle \vec{y}_i, \vec{y}_j \rangle \geq 0$ for any pair $i, j \in [t]$. The second inequality is true because $\langle \vec{y}_i, \vec{y}_i \rangle \geq \langle \vec{y}_1, \vec{y}_1 \rangle$ for any $i \in [t]$ as $\vec{y}_i$ are a 0/1 vectors and $\vec{y}_1$ has the minimum Hamming weight. The third inequality follows from Cauchy-Schwarz. □

Combining Eq. (3.9) with Eq. (3.10)-Eq. (3.11), Lemma 3.3.1 and Lemma 3.3.2, we get our first lower bound result.

Theorem 3.3.3. *Suppose $a_1, a_2, \cdots, a_t \in \{0,1\}^n$ are the t solutions of $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$, where $\mathcal{F}_1 = \{f_1, \ldots, f_m\}$ and $\mathcal{F}_2 = \{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$, and let h be the minimum Hamming weight of the t solutions $a_1, a_2, \cdots, a_t$. Let d be the selected degree on constructing the Macaulay linear system $\mathcal{M}\vec{y} = \vec{b}$ and let $\vec{y}_1, \vec{y}_2, \cdots, \vec{y}_t$ be the corresponding solution vectors of the Macaulay linear system $\mathcal{M}\vec{y} = \vec{b}$ under the assignments $a_1, a_2, \cdots, a_t$ respectively.*

If all the t solutions $a_1, a_2, \cdots, a_t$ have the same Hamming weight h or the minimum ℓ_2-norm solution vector $\vec{y} = \mathcal{M}^+\vec{b}$ is in the convex hull of $\vec{y}_1, \vec{y}_2, \cdots, \vec{y}_t$, then the tQLScn of $\mathcal{M}$ of $\mathcal{F}$ in the Macaulay linear system

- *using max degree is $\kappa_{\vec{b}}(\mathcal{M}) \geq \sqrt{((d+1)^h - 1)/t}$, and*
- *using total degree is $\kappa_{\vec{b}}(\mathcal{M}) \geq \sqrt{\left(\binom{d+h}{h} - 1\right)/t}$.*

In particular in the setup in [CG22], using max degree $d = 3n$, we have $\kappa_{\vec{b}}(\mathcal{M}) \geq \sqrt{(3n)^h/t}$.

Now we give a lower bound in terms of the smallest Hamming weight in the binary solution set. For this, we use the following purely geometrical lemma.

Lemma 3.3.4 (Shortest vector within an affine subspace)**.** *Let $V \in \mathbb{C}^{n \times k}$ be a matrix with columns $v_1, v_2, \ldots, v_k$, and let A be their (complex) affine hull $A := \{Vx : x \in \mathbb{C}^k \text{ s.t. } \langle x, \mathbf{1} \rangle = 1\}$. Then A contains the origin if and only if the column space of the Gram matrix $G = V^\dagger V$ does not contain $\mathbf{1}$. Moreover, the length-square γ^* of the shortest vector (with respect to the ℓ_2-norm) in A is*

$$\gamma^* = \max\{\gamma \colon G - \gamma \mathbf{1} \cdot \mathbf{1}^T \succeq 0\}. \tag{3.12}$$

Furthermore, if A does not contain the origin, then $\gamma^* = 1/\langle \mathbf{1}, G^+\mathbf{1}\rangle$ and the shortest vector in A is $V \cdot w$ for $w = G^+\mathbf{1}/\langle \mathbf{1}, G^+\mathbf{1}\rangle$.

Note that the problem of finding the length of the shortest vector in an affine subspace can also be reformulated as the following SDP:

$$\gamma^* = \min_{\rho \succeq 0} \operatorname{Tr}(G\rho) \qquad \text{subject to: } \operatorname{Tr}(\mathbf{1} \cdot \mathbf{1}^T \rho) = 1,$$

where without loss of generality we can assume that the above optimizer ρ has rank 1.

By the weak duality of SDPs, and utilizing the dual of the above problem, we get:

$$\gamma^* \geq \max_{\gamma \in \mathbb{R}} \gamma \qquad \text{subject to: } G - \gamma \mathbf{1} \cdot \mathbf{1}^T \succeq 0.$$

In fact, the following proof of Lemma 3.3.4 shows that the above inequality is tight, i.e., strong duality always holds for this SDP.

Proof. The shortest vector s in the affine subspace is orthogonal to all vectors of the form $v_i - v_j$, therefore we must have that $\langle s, v_i\rangle$ is constant for all $i \in [k]$, i.e., $V^\dagger s \propto \mathbf{1}$. Since s is in the column space of V, we can write it in the form $s = V \cdot w$ so we get $V^\dagger s = V^\dagger V \cdot w = Gw \propto \mathbf{1}$. If $s \neq 0$ then this implies that $\mathbf{1}$ is in the column space of G, consequently $0 \neq \langle \mathbf{1}, G^+\mathbf{1}\rangle$ and thereby $s = V \cdot w$ for $w = G^+\mathbf{1}/\langle \mathbf{1}, G^+\mathbf{1}\rangle$.[8] From this we can conclude $\gamma^* = \|s\|^2 = \langle w, Gw\rangle = 1/\langle \mathbf{1}, G^+\mathbf{1}\rangle$.

Conversely, if $\mathbf{1}$ is in the column space of G then $s = V \cdot w$ for $w = G^+\mathbf{1}/\langle \mathbf{1}, G^+\mathbf{1}\rangle$ is a non-zero vector, which is the shortest vector in A due to the fact that it is orthogonal to all vectors of the form $v_i - v_j$. We can conclude that $\gamma^* \neq 0$ iff $\mathbf{1}$ is in the column space of G.

If $\gamma^* > 0$, we can formulate a "dual" optimization problem the following way: $\gamma^* = \max\{\gamma \colon 1 - \gamma/\gamma^* \geq 0\} = \max\{\gamma \colon 1 - \gamma\langle \mathbf{1}, G^+\mathbf{1}\rangle \geq 0\} = \max\{\gamma \colon I - \gamma\sqrt{G^+}\mathbf{1} \cdot \mathbf{1}^T\sqrt{G^+} \succeq 0\}$. By multiplying the matrices with $\sqrt{G}$ from both sides we get the following equivalent maximization formulation

$$\gamma^* = \max\{\gamma \colon G - \gamma\sqrt{G}\sqrt{G^+}\mathbf{1} \cdot \mathbf{1}^T\sqrt{G^+}\sqrt{G} \succeq 0\}. \tag{3.13}$$

[8] For the last implication note that we already showed $w = \beta G^+\mathbf{1} + v$, where $v \in \ker(G)$. As $\ker(G) = \ker(V)$ we can assume without loss of generality that $v = 0$. It is easy to see that $VG^+\mathbf{1}/\langle \mathbf{1}, G^+\mathbf{1}\rangle \in A$, and since A is an affine subspace not containing the origin, it can only contain one vector of the form $\beta VG^+\mathbf{1} \colon \beta \in \mathbb{C}$, thus $s = VG^+\mathbf{1}/\langle \mathbf{1}, G^+\mathbf{1}\rangle$. (Indeed, if two distinct vectors x, y are in A and $y = \lambda x$, then $\frac{1}{1-\lambda}y - \frac{\lambda}{1-\lambda}x = 0$ is also in A.)

Note that $\sqrt{G^+}\sqrt{G} = \sqrt{G}\sqrt{G^+} = (G^+G)$ is the orthogonal projector to the column space of G. Since $\mathbf{1}$ is in the image of G the above Eq. (3.13) is equivalent to Eq. (3.12).

On the other hand, if $\gamma^* = 0$, then $\mathbf{1}$ is not in the column space of G and so $(I - G^+G)\mathbf{1} \neq 0$. Observe that $G - \gamma \mathbf{1} \cdot \mathbf{1}^T \succeq 0$ implies $(I - G^+G)G(I - G^+G) - \gamma(I - G^+G)\mathbf{1} \cdot \mathbf{1}^T(I - G^+G) \succeq 0$ or equivalently $-\gamma(I - G^+G)\mathbf{1} \cdot \mathbf{1}^T(I - G^+G) \succeq 0$, which then only holds for $\gamma \leq 0 = \gamma^*$. Consequently, Eq. (3.12) holds even in the case $\gamma^* = 0$. $\square$

Suppose that the Boolean solution set of the polynomial system is $S = \{a_1, a_2, \cdots, a_t\}$. Let A_S be the affine subspace corresponding to the solution set S, spanned by the monomial solution vectors $y_1, y_2, \cdots, y_t$ of the linear system $\mathcal{M}\vec{y} = \vec{b}$ corresponding to the Boolean solutions $a_1, a_2, \cdots, a_t$ respectively. We wish to lower bound the length of the shortest vector in A_S; for this, it suffices to find the length of the shortest vector in an enlarged affine subspace $A_{S'} \supseteq A_S$.

Let S' be the symmetrized solution set of S by applying all possible permutations of the variables of a_i's and taking their union. Let $A_{S'}$ be the affine subspace corresponding to the solution set S' and let v be the shortest vector in $A_{S'}$. For each Hamming weight h that appears in S', there is a symmetrized monomial solution vector $\mathbf{v}_h$. This $\mathbf{v}_h$ equals the average over all the monomial solution vectors that are associated with the Boolean solutions of Hamming weight h.

Next, we will argue that the minimum ℓ_2-norm vector $v \in A_{S'}$ is an affine combination of the symmetrized monomial solution vectors $\mathbf{v}_h$. If we apply an induced permutation on the coordinates of v according to a permutation of the Boolean variables, then the ℓ_2-norm of the resulting vector $u \in A_{S'}$ is equal to the ℓ_2-norm of v. Because v has the minimum ℓ_2-norm in $A_{S'}$ we have $\left\|\frac{u+v}{2}\right\|_2 \geq \|v\|$, and due to $\|u\| = \|v\|$ by the triangle inequality $\left\|\frac{u+v}{2}\right\|_2 \leq \|v\|$ (equality holds if and only if $u = v$), the resulting vector u is equal to v. Therefore, the shortest vector v is invariant under all the possible induced permutations, therefore we can conclude that v is an affine combination of the symmetrized monomial solution vectors $\mathbf{v}_h$.

Now, we can lower bound $\left\|\mathcal{M}_{\mathcal{F}}^{+} b\right\|$ by finding the lowest ℓ_2 norm of a vector in the affine subspace spanned by the symmetrized vectors $\mathbf{v}_h$ corresponding to the Hamming weights h that appear in S. This can be achieved by considering the Gram matrix as explained in Lemma 3.3.4.

In order to compute this Gram matrix, we need to understand the symmetrized vectors $\mathbf{v}_h$. For this, let us introduce the following orthonormal vector system $(\mathbf{b}_s)$, corresponding to the set of monomials $\mathfrak{m}_s^d$ that contain exactly s variables with a non-zero exponent, with the degree of the monomials being at most d, then $\mathbf{b}_s := \frac{1}{\sqrt{|\mathfrak{m}_s^d|}} \sum_{m \in \mathfrak{m}_s^d} e_m$.

Also let $\Pi_{\mathfrak{m}_s^d} = \sum_{m \in \mathfrak{m}_s^d} e_m \cdot e_m^T$ be the projector to coordinates in $\mathfrak{m}_s^d$. Finally, let c_s^d be the number of monomials that contain s specific variables with a non-zero exponent and have a degree at most d, so that $|\mathfrak{m}_s^d| = \binom{n}{s} c_s^d$.

One can see that for any $a \in \{0,1\}^n$ of Hamming weight h we have $c_s^d \binom{h}{s} = \left\langle \mathbf{1}, \Pi_{\mathfrak{m}_s^d} \vec{y}^{(a)} \right\rangle = \left\langle \mathbf{1}, \Pi_{\mathfrak{m}_s^d} \mathbf{v}_h \right\rangle$. Since $\mathbf{v}_h$ has uniform coordinates over $\mathfrak{m}_s^d$ we have $\left\| \Pi_{\mathfrak{m}_s^d} \mathbf{v}_h \right\|^2 = |\mathfrak{m}_s^d| \left(\frac{\left\langle \mathbf{1}, \Pi_{\mathfrak{m}_s^d} \mathbf{v}_h \right\rangle}{\|\mathfrak{m}_s^d\|} \right)^2 = c_s^d \binom{n}{s} \left(\frac{c_s^d \binom{h}{s}}{c_s^d \binom{n}{s}} \right)^2 = c_s^d \binom{h}{s}^2 / \binom{n}{s}$, consequently

$$\mathbf{v}_h = \sum_{s=1}^{h} \sqrt{c_s^d \binom{h}{s}^2 / \binom{n}{s}} \mathbf{b}_s, \tag{3.14}$$

and the Gram matrix G of the symmetrized vectors has matrix elements

$$G_{ij} = \langle \mathbf{v}_i, \mathbf{v}_j \rangle = \sum_{s=1}^{n} c_s^d \binom{i}{s} \binom{j}{s} / \binom{n}{s}.$$

Together with Lemma 3.3.4 this enables us to give a lower bound on the smallest ℓ_2-norm solution in terms of the minimal Hamming weight appearing in the solution set as follows.

Theorem 3.3.5. *Suppose that $\mathcal{F}$ is a Boolean polynomial system with n Boolean variables where each solution has Hamming weight at least h, and $d \geq n$. Recall that tQLScn, defined as $\kappa_{\vec{b}}(\mathcal{M}) = \|\mathcal{M}\| \frac{\left\|\mathcal{M}^+ \vec{b}\right\|}{\left\|\vec{b}\right\|}$, is also lower bounded by the smallest ℓ_2-norm solution by Eq. (3.9). Then the degree-d Macaulay linear system's tQLScn is lower bounded by*

$$\frac{1}{\langle \mathbf{1}, (G^{(h)})^{-1} \mathbf{1} \rangle} = \max\{\gamma \colon G^{(h)} - \gamma \mathbf{1} \cdot \mathbf{1}^T \succeq 0\}, \tag{3.15}$$

where $G \in \mathbb{R}^{n \times n}$ is the Gram matrix whose (i,j) matrix element is

$$G_{ij} = \sum_{s=1}^{n} c_s^d \binom{i}{s} \binom{j}{s} / \binom{n}{s}, \tag{3.16}$$

$c_s^d = d^s$ for max degree, while $c_s^d = \binom{d}{s}$ for total degree, and finally $G^{(h)}$ is the bottom-right $(n-h+1)$ by $(n-h+1)$ minor of G.[9]

The expression in Eq. (3.15) is difficult to bound analytically, but it appears to be

[9]When $d \geq n$, due to the triangular shape of the non-zero coefficients of the vectors in Eq. (3.14) it is easy to see that G has full rank, i.e., it is positive definite. It follows, that all principal submatrices of G are also positive definite, i.e., have full rank, therefore $G^{(h)}$ is invertible.

exponentially large in terms of h for large enough d. In particular, we could verify[10] that for max degree $d = 3n$ Eq. (3.15) is lower bounded by $h^h/2$ for every $h \in [n]$ up to $n = 300$.

3.3.3 Comparison to brute-force search

In case there is a unique Boolean solution we showed that the lower bound of the running time of the quantum algorithm using the QLS algorithm is exponential in the Hamming weight of the unique Boolean solution, and we provided strong evidence that this is also true when there are multiple solutions.

It is useful to compare the QLS-based approach to classical brute-force search and also to Grover's algorithm. In case we know that the unique solution has Hamming weight h, we can simply classically search through all the $\binom{n}{h}$ different Hamming-weight-h assignments of the original polynomial system. We can also use Grover search to find such an assignment with $\mathcal{O}\left(\sqrt{\binom{n}{h}}\right)$ evaluations of the polynomials. Even if we do not a priory know the Hamming weight, we can classically iterate over increasing Hamming weights w of n-bit strings which requires at most $\mathcal{O}\left(\sum_{w=0}^{h}\binom{n}{w}\right)$ different possible assignments to be checked before finding the solution, which in the case $h \leq n/2$ can be bounded by $\mathcal{O}\left(\sqrt{h}\binom{n}{h}\right)$ as we show in Appendix A.2. For the $h > n/2$ case, a similar complexity can be achieved by searching through decreasing Hamming weights. In the quantum case, naively iterating through increasing Hamming weights and using Grover's algorithm for each weight gives a complexity bound of $\mathcal{O}\left(h\sqrt{\binom{n}{h}}\right)$.

Moreover, we can use a slight variant of Grover's algorithm for searching through an unknown size search space[11] which requires only $\mathcal{O}\left(\sqrt[4]{h}\sqrt{\binom{n}{h}}\right)$ evaluations of the polynomials. By comparing this to the lower bounds of Theorem 3.3.3 one can see that in case $d + h \geq n$ Grover's algorithm performs at least as well as the QLS-based algorithm (up to some potential lower order correction $\sqrt[4]{h}$), and the algorithm of Chen and Gao [CG22] where the max degree $d = 3n$ is definitely outperformed by Grover search.

In case there are multiple solutions, but all their Hamming weights are the same, Theorem 3.3.3 ensures that we do not get a bigger reduction in the condition number

[10] Aided by symbolic computations executed by Mathematica 12.3 on Linux. See [Src22] for the code.

[11] One can use an algorithm analogous to the "exponential Grover search" [BBHT98] in order to check for a unique solution in subsequently enlarged search spaces corresponding to larger and larger Hamming-weights. By carefully choosing the sequence of upper bounds on the Hamming weights such that the search space expands in each consecutive iteration by a bounded multiplicative factor in $[c, C] \subset (1, \infty)$ the claimed running time bound follows.

than the analogous speedup we can already achieve by plain Grover search. So the above Grover-based algorithm still performs just as competitively.

In the general case of having multiple solutions with different Hamming weights, the situation is harder to analyze, but we could still obtain an exponential lower bound on tQLScn in terms of the smallest Hamming weight solution up to $n = 300$, providing strong evidence for Chen and Gao's algorithm [CG22] having a best case complexity that is exponentially large in terms of the minimal Hamming weight of a solution, making it unlikely that their algorithm would give a substantial improvement over brute-force Grover search.

3.4 The Boolean Macaulay linear system and its tQLScn

In this section, we give an equivalent but more efficient way to represent the Macaulay matrix using the fact that we are only searching for 0/1 solutions in $\mathbb{C}$. This results in a smaller lower bound on the tQLScn of size $\Omega(2^{h/2})$. While the quantum algorithm's running time is still exponentially large for larger Hamming weight solutions, for Hamming weight $h = \Theta(\log n)$ the smaller lower bound leaves open the possibility of a quasipolynomial speedup compared to the classical brute-force search algorithm having running time $\mathcal{O}\left(\binom{n}{h}\right)$.

In this section, we again include the field polynomials $\mathcal{F}_2 = \{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$ for the field $\mathbb{F}_2$ together with the input polynomials $\mathcal{F}_1$. Solving the system $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$ forces the roots to be effectively Boolean even though the underlying field is $\mathbb{C}$. This allows all monomials in an equation to be replaced with equivalent multilinear versions and a reduced Macaulay matrix will be defined that has a more compact form. This was done in [BFSS13] for finite fields, where the extra equations prevented solutions from being in field extensions. We derive the analogous matrix when the solutions are forced to be Boolean but the arithmetic is over $\mathbb{C}$. Additionally, in our case (similarly to Chen and Gao's original construction [CG22]) the structure of the Boolean solutions makes it possible to extract the Boolean solutions from measuring the quantum state corresponding the the solution vectors (over $\mathbb{C}$).

Let $\psi : R \to R$ map a monomial to its multilinear image as $\psi(\prod_{i=1}^n x_i^{a_i}) = \prod_{i=1}^n x_i^{\min\{a_i,1\}}$, and extend it to $R = \mathbb{C}[x_1, \ldots, x_n]$ by linearity. For example, $\psi(3x_1^3x_2 - 1) = 3x_1x_2 - 1 = \psi(x_1^3x_2 - 2x_1^2x_2^2 + 4x_1x_2 - 1) \neq x_1 - x_1x_2 - 1 = \psi(x_1^3 - 2x_1^2x_2^2 + x_1x_2 - 1)$.

Lemma 3.4.2 will show that having max degree higher than 1 becomes redundant, so the following definition only has rows up to max degree 1, and in this section we will set the total degree $d = n$, the number of variables. For notation, let $\hat{m}$ and $\hat{m}'$ denote monomials, and let m, m', and m'' denote multilinear monomials (i.e., monomials with max degree at most 1).

Definition 3.4.1. *The* Boolean Macaulay matrix $\hat{\mathcal{B}}$ of degree d *of* $\mathcal{F}_1 = \{f_1, \ldots, f_m\} \subseteq \mathbb{C}[X]$ *is the matrix where each row is labeled by a pair of polynomials* (m, f) *and contains the corresponding coefficient vector of the polynomial* $\psi(mf)$*. The rows ranges over all* $f \in \mathcal{F}_1$ *and multilinear monomials* m *such that* $\psi(mf)$ *has degree at most* d*. The columns are labeled by the set of multilinear monomials in* $x_1, \ldots, x_n$ *of degree at most* d *and are ordered with respect to a specified monomial ordering. The matrix element in the row corresponding to* (m, f) *and the column corresponding to the monomial* m' *is the coefficient of* m' *in the polynomial* $\psi(mf)$*.*

Note that compared to the Macaulay matrix, in addition to forcing answers to be Boolean, the Boolean Macaulay matrix is reduced with a certain way, by eliminating polynomials with the max degree at least 2. Next, we will show that the Boolean Macaulay matrix can be obtained as a submatrix of the Macaulay matrix $\hat{\mathcal{M}}$ of max degree d corresponding to the set of polynomials $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$ after Gaussian reduction on the rows.

First, we consider the special case when $\mathcal{F}_1 = \emptyset$ and perform the row reduction on the Macaulay matrix of $\mathcal{F}_2$ to show that the field equations $\mathcal{F}_2$ take a special form.

Lemma 3.4.1. *Let the Macaulay matrix* $\hat{\mathcal{M}}_2$ *of max degree* d *of* $\mathcal{F}_2$ *have its columns ordered such that they are partitioned into two parts the following way: the labels on the right side are multilinear monomials (including the degree 0 monomial 1) and ordered in ascending order with respect to the integer represented by the exponent vector of the multilinear monomial, and let the left side columns be labeled by non-multilinear monomials and ordered under any monomial order.*

Then using row operations, $\hat{\mathcal{M}}_2 = [L_2 \;\; R_2]$ *can be reduced to* $\hat{\mathcal{M}}'_2 = [I_2 \;\; B_2]$ *where* I_2 *is the identity matrix of dimension* $(d+1)^n - 2^n$ *with rows and columns labeled by non-multilinear monomials, and rows with zeros are removed.*

Proof. The rows of $\hat{\mathcal{M}}_2$ are indexed by pairs of polynomials $(\hat{m}, x_j^2 - x_j)$, where $\hat{m}$ is a monomial and $\operatorname{maxdeg} \hat{m}(x_j^2 - x_j) \leq d$. The approach is to first change each row, which starts with coefficients for a polynomial $\Pi_i x_i^{a_i} - x_j^{-1}\Pi_i x_i^{a_i}$ for some j in $\hat{\mathcal{M}}_2$, to

the coefficients of $\Pi_i x_i^{a_i} - \Pi_i x_i^{\min\{a_i,1\}}$. At this point, the left side of the matrix has at most one 1 in each row. The second step is to zero out the bottom rows.

For the first step, work in descending the total degree of the polynomials, starting at degree nd. Let the current row have the coefficients of $\Pi_i x_i^{a_i} - \Pi_i x_i^{b_i}$ during the algorithm. Let $b_j \geq 2$ for some j, or else this row is reduced. Because $\deg \Pi_i x_i^{b_i} < \deg \Pi_i x_i^{a_i}$, the rows where $\Pi_i x_i^{b_i}$ is the highest degree term have not changed yet, and therefore, one of the rows has the coefficients of $\Pi x_i^{b_i} - x_j^{-1} \Pi x_i^{b_i}$. Adding this row changes the current row to $\Pi_i x_i^{a_i} - \Pi_i x_i^{b_i} + (\Pi_i x_i^{b_i} - x_j^{-1} \Pi_i x_i^{b_i}) = \Pi_i x_i^{a_i} - x_j^{-1} \Pi_i x_i^{b_i}$, which has decreased the total degree of the second term by one while keeping the set of variables the same. This is repeated until the row has the coefficients of $\Pi_i x_i^{a_i} - \Pi_i x_i^{\min\{a_i,1\}}$.

At the end of the first step, each row at the left side (i.e., columns indexed by non-multilinear monomials) has at most one 1. This is in fact a constructive argument showing that

$$\prod_{i=1}^{n} x_i^{a_i} - \prod_{i=1}^{n} x_i^{\min\{a_i,1\}} \in \langle \mathcal{F}_2 \rangle .$$

Consider any two rows indexed by $\hat{m}(x_i^2 - x_i)$ and $\hat{m}'(x_j^2 - x_j)$. If $\hat{m} x_i^2 = \hat{m}' x_j^2$, they have the same set of variables, and therefore $\psi(\hat{m} x_i) = \psi(\hat{m}' x_j)$, so the rows are equal and one can be eliminated (zeroed out). Keep doing this until for every leading non-multilinear monomial there is only one row where the corresponding coefficient is nonzero.

Because for every column in the left part, there is a unique nonzero row with the corresponding leading monomial, the matrix can be written (up to permutation of the rows) as $\begin{bmatrix} I_2 & B_2 \\ 0 & 0 \end{bmatrix}$. □

Lemma 3.4.2. *Let* $\hat{\mathcal{M}} = \begin{bmatrix} L_1 & R_1 \\ L_2 & R_2 \end{bmatrix}$ *be the Macaulay matrix for* $\mathcal{F}_1 \cup \mathcal{F}_2$ *with the row and column ordering from Lemma 3.4.1. Using row operations (and then removing some zero rows),* $\hat{\mathcal{M}}$ *can be reduced to* $\hat{\mathcal{M}}' = \begin{bmatrix} 0 & \hat{\mathcal{B}} \\ I_2 & B_2 \end{bmatrix}$ *where* $\hat{\mathcal{B}}$ *is the Boolean Macaulay matrix of* $\mathcal{F}_1$*, and* I_2, B_2 *are as in Lemma 3.4.1.*

Proof. By Lemma 3.4.1, using row operations on the $\mathcal{F}_2$ submatrix of $\hat{\mathcal{M}}$ we get a matrix $\begin{bmatrix} L_1 & R_1 \\ I_2 & B_2 \end{bmatrix}$, where zero rows from $\mathcal{F}_2$ are removed.

Row operations utilizing I_2 can then be used to zero out the top left, resulting in

$\begin{bmatrix} 0 & R_1' \\ I_2 & B_2 \end{bmatrix}$. From the polynomial perspective, this maps all the non-multilinear polynomials to their corresponding multilinear polynomials under ψ, i.e., for each monomial $\prod_{i=1}^{n} x_i^{a_i}$, the map encodes the coefficient vector of $\psi((\prod_{i=1}^{n} x_i^{a_i}) f_j), 1 \leq j \leq m$ into the Macaulay matrix as a row vector.

Recall that the Macaulay matrix has rows labeled by pairs; observe that rows $(\hat{m}, f_i)$ and $(\hat{m}', f_i)$ will be equal when $\psi(\hat{m}) = \psi(\hat{m}')$. In particular for any non-multilinear monomial $\hat{m}$ the rows $(\hat{m}, f_i)$ and $(\psi(\hat{m}), f_i)$ will be equal at this point, so we can eliminate (zero out and then remove) any row indexed by non-multilinear monomials. To be compatible with Definition 3.4.1 we choose not to further reduce/remove rows despite the fact $\hat{\mathcal{B}}$ might have zero rows, for example, rows with $\psi(\hat{m}f) = \psi(\hat{m}'f')$, but $f \neq f'$.

As claimed, in matrix notation we get $\hat{\mathcal{M}}' = \begin{bmatrix} 0 & \hat{\mathcal{B}} \\ I_2 & B_2 \end{bmatrix}$. □

As in the general case let $\hat{\mathcal{B}} = [M \quad -\vec{b}]$ define the Boolean Macaulay linear system as $M\vec{y} = \vec{b}$, where the entries of $\vec{y}$ are labeled by the nontrivial multilinear monomials and $\vec{b} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}$.

Recall that a matrix is s-sparse if it has at most s entries in any row or column.

Lemma 3.4.3. *The Boolean Macaulay matrix $\hat{\mathcal{B}}$ of total degree d of $\mathcal{F}_1$ is an $\mathcal{O}(m \cdot \#\mathcal{F}_1)$-sparse matrix.*

Proof. The Boolean Macaulay matrix $\hat{\mathcal{B}}$ is constructed by placing $\psi(tf)$ in a row for a multilinear monomial t and $f \in \mathcal{F}_1$, so the support of each row has size at most $\mathcal{O}(\#\mathcal{F}_1)$.

For the column sparsity first consider the Boolean Macaulay matrix of $\{t\}$, which has a 1 matrix element at column m' and row (m'', t) if and only if $m' = \psi(m'' \cdot t)$. This can only happen if t divides m', so we can define $\bar{t} := m'/t$. It is easy to see that $m' = \psi(m'' \cdot t)$ if and only if $m'' = \bar{t} \cdot m_d$ for some monomial m_d that divides t. This implies that the column sparsity of the Boolean Macaulay matrix of $\{t\}$ equals the number of divisors of t which is at most 4 if the multilinear monomial t has (total) degree at most 2.

Now consider the Boolean Macaulay matrix of $\{f\}$ for some (at most) quadratic polynomial $f \in \mathcal{F}_1$. Observe that f is a linear combination of at most $\#\mathcal{F}_1$ monomials of degree at most 2 and the Boolean Macaulay matrix of $\{f\}$ is likewise the linear combination of the Boolean Macaulay matrix of these (at most) quadratic monomials. So

the column sparsity of the Boolean Macaulay matrix of $\{f\}$ is at most $4 \cdot \#\mathcal{F}_1$. Finally, the entire Boolean Macaulay matrix of $\mathcal{F}_1$ is simply given by stacking the Boolean Macaulay matrices of $\{f\}$ for $f \in \mathcal{F}_1$, so the total column sparsity is at most $4m \cdot \#\mathcal{F}_1$. □

Note that this also implies that M, a submatrix of $\hat{\mathcal{B}}$, is also sparse. Moreover, the location and value of the nonzero entries of each column/row of $\hat{\mathcal{B}}$ can be efficiently computed.

Now we show that the Boolean Macaulay linear system is equivalent to the Macaulay linear system. It follows that solving the Boolean Macaulay linear system returns a correct solution of the Boolean polynomial system. [12]

Lemma 3.4.4. *Let $M_1\vec{y}_1 = \vec{b}_1$ be the Macaulay linear system of a polynomial system $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$ and let $M_2\vec{y}_2 = \vec{b}_2$ be the corresponding Boolean Macaulay linear system, where the Macaulay matrix is $\hat{\mathcal{M}} = [M_1 \quad -\vec{b}_1]$, the Boolean Macaulay matrix is $\hat{\mathcal{B}} = [M_2 \quad -\vec{b}_2]$. Then a solution $\hat{y}_2$ of the Boolean Macaulay linear system $M_2\vec{y}_2 = \vec{b}_2$ corresponds to a solution $\hat{y}_1$ of the Macaulay linear system $M_1\vec{y}_1 = \vec{b}_1$.*

Proof. By Lemma 3.4.2, $\hat{\mathcal{M}} = [M_1 \quad -\vec{b}_1]$ can be reduced to $\begin{bmatrix} 0 & \hat{\mathcal{B}} \\ I_2 & B_2 \end{bmatrix}$ by row operations, where $\hat{\mathcal{B}} = [M_2 \quad -\vec{b}_2]$ is the Boolean Macaulay matrix. Also, $B_2 = [B_2' \quad 0]$ because the last column of the reduced Macaulay matrix $\begin{bmatrix} 0 & \hat{\mathcal{B}} \\ I_2 & B_2 \end{bmatrix}$ is indexed by the degree 0 monomial 1 and the polynomials generated from $\mathcal{F}_2$ have no constant terms. Therefore $\begin{bmatrix} 0 & \hat{\mathcal{B}} \\ I_2 & B_2 \end{bmatrix} = \begin{bmatrix} 0 & M_2 & -\vec{b}_2 \\ I_2 & B_2' & 0 \end{bmatrix}$.

Since performing row operations on the augmented matrix of a linear system does not change the set of solutions, solving the Macaulay linear system $M_1\vec{y}_1 = \vec{b}_1$ is equivalent to solving the linear system $\begin{bmatrix} 0 & M_2 \\ I_2 & B_2' \end{bmatrix} \begin{bmatrix} \vec{z}_1 \\ \vec{y}_2 \end{bmatrix} = \begin{bmatrix} \vec{b}_2 \\ 0 \end{bmatrix}$, where the entries of $\vec{y}_2$ and $\vec{z}_1$ are indexed by nontrivial multilinear monomials and non-multilinear monomials respectively.

For the linear system

$$\begin{bmatrix} 0 & M_2 \\ I_2 & B_2' \end{bmatrix} \begin{bmatrix} \vec{z}_1 \\ \vec{y}_2 \end{bmatrix} = \begin{bmatrix} \vec{b}_2 \\ 0 \end{bmatrix}$$

we have $M_2\vec{y}_2 = \vec{b}_2$, which is the Boolean Macaulay linear system, and $\vec{z}_1 + B_2'\vec{y}_2 = 0$.

If $\hat{y}_2$ is a solution of the Boolean Macaulay linear system $M_2\vec{y}_2 = \vec{b}_2$, set $\hat{z}_1$ to be

[12] This observation also implies that the complete solving degree in Chen and Gao's original approach is always at most $n + 2$, tightening their upper bound $3n$.

$-B_2'\hat{y}_2$, then $\begin{bmatrix}\hat{z}_1\\ \hat{y}_2\end{bmatrix}$ is a solution of the linear system $\begin{bmatrix}0 & M_2\\ I_2 & B_2'\end{bmatrix}\begin{bmatrix}\vec{z}_1\\ \vec{y}_2\end{bmatrix} = \begin{bmatrix}\vec{b}_2\\ 0\end{bmatrix}$. Because the Macaulay linear system is equivalent to the linear system $\begin{bmatrix}0 & M_2\\ I_2 & B_2'\end{bmatrix}\begin{bmatrix}\vec{z}_1\\ \vec{y}_2\end{bmatrix} = \begin{bmatrix}\vec{b}_2\\ 0\end{bmatrix}$, therefore, a solution $\hat{y}_2$ of the Boolean Macaulay linear system $M_2\vec{y}_2 = \vec{b}_2$ corresponds to a solution $\hat{y}_1$ of the Macaulay linear system $M_1\vec{y}_1 = \vec{b}_1$. □

As M is a $\mathcal{O}(m \cdot \#\mathcal{F})$-sparse row/column computable matrix and we can efficiently prepare the sparse vector $\vec{b}$ as a quantum state $|b\rangle$, we can apply a QLS algorithm to "solve" the Boolean Macaulay linear system $M\vec{y} = \vec{b}$, which takes time $\widetilde{\mathcal{O}}(\text{poly}(n)\kappa(M)\log(1/\epsilon))$ [CKS17].

The key parameter in the running time is the condition number of the matrix M. Next, we will provide a lower bound of the tQLScn of M and thus also a lower bound on known QLS algorithms.

3.4.1 Lower bound on the tQLScn $\kappa_{\vec{b}}(M)$

Suppose $a_1, a_2, \cdots, a_t \in \{0,1\}^n$ are the t solutions of $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$, where $\mathcal{F}_1 = \{f_1, \ldots, f_m\}$ and $\mathcal{F}_2 = \{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$, and let h be the minimum Hamming weight of the t solutions $a_1, a_2, \cdots, a_t$. Let $\vec{y}_1, \vec{y}_2, \cdots, \vec{y}_t$ be the corresponding solution vectors of the Boolean Macaulay linear system $M\vec{y} = \vec{b}$ under the assignments $a_1, a_2, \cdots, a_t$ respectively.

In this case, we have $\|M\| \geq 1/2$ as M has at least one matrix element which has an absolute value at least $1/2$ [13]. Analogously to Theorem 3.3.3 we get:

Corollary 3.4.5. *Let $\hat{\mathcal{B}} = [M \quad -\vec{b}]$ be the Boolean Macaulay matrix of $\mathcal{F}$ with columns labeled by multilinear monomials. Let h be the minimum Hamming weight of the t solutions $a_1, a_2, \cdots, a_t$. If all the t solutions $a_1, a_2, \cdots, a_t$ have the same Hamming weight h or the minimum ℓ_2-norm solution vector $\vec{y} = M^+\vec{b}$ is in the convex hull of $\vec{y}_1, \vec{y}_2, \cdots, \vec{y}_t$, then the tQLScn $\kappa_{\vec{b}}(M) \geq \frac{1}{2}\sqrt{(2^h - 1)/t}$ of M of $\mathcal{F}$.*

For $h = \Theta(\log n)$, this lower bound does not rule out the possibility that the Macaulay matrix has a polynomial condition number, which would result in the quantum algorithm beating the brute-force classical algorithm that runs in time $\widetilde{\mathcal{O}}\left(\binom{n}{\log n}\right)$.

[13] After applying Red2 there is at least one polynomial f with a constant term of magnitude 1. If f does not have a degree-1 monomial x_i, then $x_i \cdot f$ has a magnitude 1 degree-1 monomial x_i, so the row (x_i, f) will have a matrix element of magnitude 1. Otherwise, suppose the coefficient of x_1 in f is c_1, then the rows $(1, f)$ and (x_1, f) will have a matrix element of magnitude c_1 and $c_1 - 1$ respectively. Therefore, at least one of them has a magnitude of at least $1/2$.

3.4.2 Details comparing running times

As we have discussed in Section 3.3.3 the classical brute-force algorithm tries all $\binom{n}{j}$ choices for the locations of the 1's in the solution a for each $j \leq h$, and its running time can be bounded $\mathcal{O}\left(\sqrt{h}\binom{n}{h}\right)$, where

$$\forall 1 \leq h \leq n : \left(\frac{n}{h}\right)^h \leq \binom{n}{h} \leq \left(\frac{en}{h}\right)^h.$$

Comparing the above expression with our tQLScn lower bound, we saw that the Gorver-enhanced brute-force search always outperforms the Macaulay matrix approach in case there is a unique solution and $d = n$ (or even $d + h \geq n$). This in particular shows that the quantum algorithm achieves at most a quadratic speed-up compared to classical brute-force search. Moreover, if one chooses $d = 3n$ and works with the max degree as Chen and Gao suggested [CG22], then $\kappa_{\vec{b}}(\mathcal{M}) \geq (3n)^{h/2}$ and so

- For $h = \Omega(\sqrt{n})$, the classical brute force algorithm is faster than the quantum algorithm.
- For $h = \mathcal{O}(\sqrt{n})$, it is unknown which is faster.

On the other hand in the Boolean case, we have only the lower bound $\kappa_{\vec{b}}(M) \geq \frac{1}{2}\sqrt{(2^h - 1)}$, so:

- For $h = pn$, where $p \in (0, \frac{1}{2}]$, the lower bound of $\kappa_{\vec{b}}(M) \geq \frac{1}{2}(2^h - 1)^{1/2}$ is exponentially large and exhaustive search takes time $\mathcal{O}\left(2^{H(p)n}\right)$ where $H(p) = -p\log p - (1-p)\log(1-p)$ is the binary entropy function, as shown in Appendix A.2.
- For $h = \mathcal{O}(1)$, $\exists$ classical algorithm that takes time $\mathcal{O}\left(\binom{n}{h}\right)$ to solve the problem efficiently by exhaustive search whereas the lower bound of $\kappa(M)$ is a constant $(2^{\mathcal{O}(1)} - 1)^{1/2}$.
- For $h = \Theta(\log n)$, we only know that $\kappa_{\vec{b}}(M) \geq \text{poly}(n)$ whereas classical exhaustive search takes time $\mathcal{O}\left(\binom{n}{\log n}\right)$. Thus, we cannot exclude the possibility that the quantum algorithm might give a quasi-polynomial speedup in this case.

Without loss of generality, let $0 \leq h \leq \frac{n}{2}$ (otherwise we can flip all variables). Then the lower bound on the tQLScn $\kappa(M)$ is always smaller than the time required by brute-force search. Thus, there is a possibility that the quantum algorithm performs better than the exhaustive search approach.

3.5 Our new improved quantum algorithm

3.5.1 A Variant of the Quantum Coupon Collector Problem

By [CG22, Corollary 3.19] and Lemma 3.4.4, if a set of polynomials $\mathcal{F}$ over $\mathbb{C}[x_1, \ldots, x_n]$ has a unique solution $a = (a_1, a_2, \cdots, a_n) \in \{0,1\}^n$, then for some d less than or equal to n, the corresponding Boolean Macaulay linear system $M\vec{y} = \vec{b}$ of total degree d has a unique solution $\vec{y} = M^+\vec{b}$, where the entries of $\vec{y}$ are indexed by multilinear monomials in $x_1, \ldots, x_n$ with total degree at most d. Let $U = \{x_1, x_2, \ldots, x_n\}$. There is a one-to-one correspondence between the subsets of U of size at most d and multilinear monomials in $x_1, \ldots, x_n$ with total degree at most d. Let S be the largest subset of U such that all the variables $x_k \in S$ have assignment $a_k = 1$ and S_d be the set containing all nonempty subsets of S that have size at most d. There is a one-to-one correspondence between the elements of the set S_d and the nonzero entries of $\vec{y}$. Given implicit access to matrix M and sparse vector b, the QLS algorithm outputs the solution vector $\vec{y}$ as the quantum state $|\vec{y}\rangle$, which encodes the nonzero entries of $\vec{y}$. Because the unique solution $\vec{y} = M^+\vec{b}$ of the Boolean Macaulay linear system is a 0/1 vector, the quantum state $|\vec{y}\rangle$ can be represented by

$$|\vec{y}\rangle = \frac{1}{\sqrt{|S_d|}} \sum_{R \in S_d} |R\rangle .$$

If we measure the quantum state $|\vec{y}\rangle$, we will get a uniformly random subset $R \in S_d$, where all the variables $x_k \in R$ have assignment $a_k = 1$. Given copies of the quantum state $|\vec{y}\rangle$, the goal is to compute S.

Next, we will reformulate this problem as a variant of the quantum coupon collector problem.

Problem 3.5.1. *Let $S \subseteq U = \{x_1, x_2, \ldots, x_n\}$ be an unknown subset and S_d be the set containing all nonempty subsets of S that have size at most d. Given copies of the state*

$$|\vec{y}\rangle = \frac{1}{\sqrt{|S_d|}} \sum_{R \in S_d} |R\rangle ,$$

which is a superposition of subsets of S of size at most d. The goal is to compute S.

Specifically, when d equals 1, this is the quantum coupon collector problem defined in [ABC$^+$20b]. They proved that $\Theta(|S| \log(\min\{|S|, n - |S|\}))$ copies of the states $|\vec{y}\rangle$ are necessary to compute S.

Without loss of generality, we can assume d is at most $|S|$ because when d is greater

than $|S|$, the quantum state $|\vec{y}\rangle$ is the same as the case of d equals $|S|$. Then, we have the following result of Problem 3.5.1.

Theorem 3.5.2. *Let $r = \mathcal{O}\left((|S|/d)\log(|S|/\varepsilon)\right)$. Measuring r copies of the quantum superposition state in Problem 3.5.1, the set S can be computed with probability at least $1-\varepsilon$.*

Since the only quantum operation is a measurement in the computational basis, this is essentially a classical coupon collector problem, where we can sample a uniformly random subset.

Proof. For any $x \in S$, the number of sets $R \in S_d$ containing x is $\sum_{i=1}^{d}\binom{|S|-1}{i-1}$ out of a total number of sets $\sum_{i=1}^{d}\binom{|S|}{i}$ in S_d. Thus, the probability of seeing x equals $\sum_{i=1}^{d}\binom{|S|-1}{i-1}/\sum_{i=1}^{d}\binom{|S|}{i}$. If $0 \le d \le \lfloor\frac{|S|}{3}\rfloor$ and by Appendix A.2, we have $\binom{|S|}{d} \le \sum_{i=1}^{d}\binom{|S|}{i} \le \binom{|S|}{d}\frac{|S|-d+1}{|S|-2d+1}$, so $\sum_{i=1}^{d}\binom{|S|-1}{i-1}/\sum_{i=1}^{d}\binom{|S|}{i} \ge \frac{d}{|S|}\cdot\frac{|S|-2d+1}{|S|-d+1}$, where $\frac{|S|-2d+1}{|S|-d+1} > \frac{1}{2}$. Hence, when $0 \le d \le \lfloor\frac{|S|}{3}\rfloor$, the probability of not seeing x after r tries is at most $(1-\frac{d}{|S|}\cdot\frac{1}{2})^r = (1-\frac{d}{2|S|})^{-\frac{2|S|}{d}\frac{d}{2|S|}r} \le \exp(-\frac{d}{2|S|}r)$. Since $\frac{\binom{|S|-1}{0}}{\binom{|S|}{1}} < \frac{\binom{|S|-1}{1}}{\binom{|S|}{2}} < \cdots < \frac{\binom{|S|-1}{d-1}}{\binom{|S|}{d}}$, the probability function $\sum_{i=1}^{d}\binom{|S|-1}{i-1}/\sum_{i=1}^{d}\binom{|S|}{i}$ is an increasing function. If $\lfloor\frac{|S|}{3}\rfloor \le d \le |S|$ and $|S| = \Omega(1)$ [14], the probability function has the minimum value when $d = \lfloor\frac{|S|}{3}\rfloor$. For all three cases, we have

$$\frac{d}{|S|}\cdot\frac{|S|-2d+1}{|S|-d+1} = \begin{cases} \frac{d+1}{6d+3} & \text{when } |S| = 3d \\ \frac{d^2+2d}{6d^2+8d+2} & \text{when } |S| = 3d+1 \\ \frac{d^2+3d}{6d^2+13d+6} & \text{when } |S| = 3d+2 \end{cases}$$

the probability of seeing x is greater than $\frac{d}{|S|}\cdot\frac{|S|-2d+1}{|S|-d+1} \ge \frac{1}{6}$. Hence, the probability of not finding x after r tries is at most $(1-\frac{1}{6})^r$.

Let $r = \mathcal{O}\left((|S|/d)\log(|S|/\varepsilon)\right)$, and by the union bound, the probability of not collecting all the elements x in S is at most ε. That is, if $r = \mathcal{O}\left((|S|/d)\log(|S|/\varepsilon)\right)$, the entire set S can be recovered with probability at least $1-\varepsilon$. □

With respect to the choice of d, there is a trade-off between the number of samples and the memory space:

- For $d = O(1)$, $r = O(|S|\log|S|)$.
- For $d = O(\log|S|)$, $r = O(|S|)$.

[14] Note that when $|S| = O(1)$, the probability of seeing x is at least a constant.

- For $d = O(\frac{|S|}{\log |S|})$, $r = O(\log^2 |S|)$
- For $\frac{|S|}{c} \leq d \leq n$, where c is a positive integer, $r = O(\log |S|)$.

3.5.2 The algorithm

When a set of polynomials $\mathcal{F}$ has a unique solution, Algorithm 1 finds the solution. If a set of polynomials has more than one solution, we apply the Valiant-Vazirani reduction Red1 to get a set of polynomials $\mathcal{F}$ that have a unique solution.

Algorithm 1 Quantum linear system algorithm for $\mathcal{F}$ over $\mathbb{C}$

Input: $\mathcal{F} \subseteq \mathbb{C}[x_1, \ldots, x_n]$ where $\mathcal{F} = \{f_1, \ldots, f_m\}$ with $\deg(f_i) = 2$ for $i = 1, \ldots, m$.

Output: The solution $a \in \{0,1\}^n$ such that $f_1(a) = \cdots = f_m(a) = 0$ over $\mathbb{C}$ when one exists.

Step 1: Apply a quantum linear system algorithm to the Boolean Macaulay linear system $M\vec{y} = \vec{b}$ of total degree n and get the solution $\vec{y}$ in the quantum state

$$|\vec{y}\rangle = \frac{1}{\sqrt{|S_d|}} \sum_{R \in S_d} |R\rangle$$

.

Step 2: Perform measurement on the quantum state $|\vec{y}\rangle$ and get outcome $|R\rangle$, then let all the variables in the set R equal 1.

Step 3: Repeat Step 1 and Step 2 $O(\log n)$ times, and then set all the remaining variables $a_j = 0$.

Step 4: Return a.

Lemma 3.5.3. *With high probability Algorithm 1 solves Problem 3.2.2 in time* $\widetilde{\mathcal{O}}(\mathrm{poly}(n)\kappa(M))$.

Proof. For the Boolean Macaulay linear system $M\vec{y} = \vec{b}$, the matrix M is $\mathcal{O}(m \cdot \#\mathcal{F})$-sparse and the vector $\vec{b}$ can be prepared as $|\vec{b}\rangle = |0\rangle^{n\lceil \log m \rceil}$. Therefore, we can apply a QLS algorithm [CKS17] to the Boolean Macaulay linear system, which takes time $\widetilde{\mathcal{O}}(\mathrm{poly}(n)\kappa(M)\log(1/\varepsilon))$. The QLS algorithm outputs a quantum state $|\vec{y}^*\rangle$, which is an approximation of $|\vec{y}\rangle$ with $\||\vec{y}\rangle - |\vec{y}^*\rangle\| \leq \varepsilon$. If we repeat the process r-times, then we essentially prepare the state $|(\vec{y}^*)^{\otimes r}\rangle$ for which we have $\langle(\vec{y})^{\otimes r}|(\vec{y}^*)^{\otimes r}\rangle = (\langle\vec{y}|\vec{y}^*\rangle)^r = (1 - \Theta(\varepsilon^2))^r$. For $\varepsilon = \mathcal{O}(1/r)$ we have that this equals $(1 - \Theta(r\varepsilon^2))$ and so $\||(\vec{y})^{\otimes r}\rangle - |(\vec{y}^*)^{\otimes r}\rangle\| = \Theta(\sqrt{r}\varepsilon)$. Then the total variation distance between the two probability distributions of any measurements on the two states $|(\vec{y})^{\otimes r}\rangle$ and $|(\vec{y}^*)^{\otimes r}\rangle$ is at most

$\Theta(\sqrt{r}\varepsilon)$ [dW19, Exercise 4.3] (see also [BV97, Lemma 3.6]), so replacing the ideal state $|(\vec{y})^{\otimes r}\rangle$ by the approximate state $|(\vec{y^*})^{\otimes r}\rangle$ induces error probability at most $\mathcal{O}\left(\sqrt{r}\varepsilon\right)$. By Lemma 3.5.2, we can extract the solution of the polynomial system $\mathcal{F}$ from $|(\vec{y})^{\otimes r}\rangle$ with high probability by choosing r to be $O(\log n)$. Therefore, letting $\varepsilon = 1/\Theta(\log n)$, with high probability Algorithm 1 solves Problem 3.2.2 in time $\widetilde{\mathcal{O}}\left(\mathrm{poly}(n)\kappa(M)\right)$. □

Compared with Chen and Gao's [CG22] algorithm, there are two differences with Algorithm 1. First, the size of the Boolean Macaulay matrix in Algorithm 1 is $m2^n \times 2^n$, which leads to a smaller lower bound of the tQLScn and leaves a possibility of superpolynomial speedup using Algorithm 1. By contrast, the size of the Macaulay matrix in Chen and Gao's algorithm is $(m+n)(3n+1)^n \times (3n+1)^n$ [15], which leads to a larger lower bound of the tQLScn that prohibits a potential quantum speedup. Second, in Algorithm 1, the polynomial system have a unique solution, so in contrast to [CG22] the Boolean Macaulay linear system stays the same for every iteration and the number of iterations (measurements) required to obtain the solution of the polynomial system is $\mathcal{O}\left(\log n\right)$. However, the Valiant-Vazirani reduction needs $\mathcal{O}\left(n\right)$ iterations to generate a polynomial system that has a unique solution with high probability. This amounts to $\mathcal{O}\left(n \log n\right)$ iterations in total to find a solution. On the other hand, in Chen and Gao's algorithm, the polynomial system could have any finite number of solutions, so the Macaulay linear system needs to be updated after each iteration (measurement) and the number of iterations is $\mathcal{O}\left(n\right)$.

[15] The parameters m, n comes from Problem 3.2.2

Chapter 4

Quantum Algorithms for the Pathfinding Problem

In this chapter, we demonstrate exponential quantum speedups for the problem of finding an s-t path in three types of graph within the oracle setting: the welded tree path graph $\mathcal{G}_P$, the welded tree circuit graph $\mathcal{G}_C$, and the regular sunflower graph $\mathcal{G}_S$. These graphs are provided through an adjacency list oracle. To show these exponential speedups, in addition to using the standard technique of problem reduction, we introduce two new quantum approaches, namely edge superposition and vertex superposition, which could potentially be useful for designing new quantum algorithms for more graph problems.

4.1 Introduction

Finding problems that enable exponential quantum speedup remains one of the biggest challenges in the area of quantum computation. The key challenge is to identify specific problem structures that quantum mechanics can uniquely exploit. Graph theory is one of the most promising areas for investigating these structures, as exponential-size graphs exhibit a wide range of complex structures.

The most well-known graph structure that allows for exponential quantum-classical separations is the welded tree graph [CCD$^+$03]. It consists of two binary trees of height n connected by a cycle that alternates between the leaves of the two trees, making its size exponentially large as (the size of the graph is $2^{n+2} - 2$). In the welded tree exit finding problem, we are given an entrance vertex s, and access to the graph via an adjacency list oracle, with the goal of finding the exit vertex t. Both s and t are distinguished from the other vertices by having degree 2. The inherent structure of the welded tree graph

creates conditions in which quantum algorithms can provide an exponential speedup over classical algorithms, making it an ideal candidate for exploring quantum computational advantages.

The welded tree graph structure is also crucial for demonstrating exponential quantum speedups in various related problems. For example, the exponential advantage in adiabatic quantum computation without the sign problem has been achieved by efficiently finding a marked vertex in a graph that is similar to the welded tree graph [GHV21]. The exponential quantum speedup of the graph property testing problem is obtained in the welded tree candy graph [BDCG$^+$20]. More recently, the exponential quantum advantage of finding a marked vertex in the welded tree graph has been generalized to more general types of graphs, such as families of random hierarchy graphs [BBK$^+$23].

Despite much progress in showing exponential speedups for various graph problems on different graph structures, there is no exponential quantum-classical separation for the problem of finding an s-t path in graphs has been discovered. Furthermore, in addition to being one of the most fundamental problems in the field of computer science and having numerous applications, the problem of finding an s-t in certain type graphs, such as the welded tree graph [Aar21] and the supersingular isogeny graph [CLG09], has played an important role in the area of quantum query complexity and post-quantum cryptography, respectively.

The welded tree pathfinding problem in the adjacency list oracle is one of the top open problems in the field of quantum query complexity [Aar21]. Given the name of the two roots s and t, the goal of the welded tree pathfinding problem is to output the names of vertices of an s-t path. While there is an efficient quantum algorithm to solve the welded tree problem, that is, finding the name of the root t given the name of the root s, it has been shown that a natural class of quantum algorithms cannot solve the welded tree pathfinding problem [CCG22].

The pathfinding problem in exponentially large expander graphs is important because the security of isogeny-based cryptography is based on the hardness of finding an s-t path in supersingular isogeny graphs [CLG09, EHL$^+$18, Wes22], which is a class of expander graphs. An isogeny graph is constructed with vertices as isomorphism classes of elliptic curves and edges as isogenies (maps) between two elliptic curves. The size of the graph is exponentially large, making it difficult to find a path (map) between two vertices (elliptic curves) of polynomial length. Although a specific isogeny-based scheme, namely the supersingular isogeny Diffie–Hellman key exchange (SIDH) [JDF11], has been

broken recently [CD23], the general problem of finding an s-t path in the supersingular isogeny graph (which is an expander graph) remains unaffected [ACL+24]. Thus, many other isogeny-based cryptosystems such as CSIDH [CLM+18] and SQISign [DFKL+20] are immune to this attack [Gal22].

Known quantum algorithms for the problem of finding an s-t path can only provide at most a polynomial speedup for general graphs as well as special graphs. The amplitude amplification technique achieves polynomial speedup for various graph problems, including the pathfinding problem in a general graph with Oracle access [DHHM06]. Quantum walk-based algorithms demonstrate polynomial quantum advantages in special graphs such as regular tree graphs, chains of star graphs [RHK17, Hil21, KH18], and supersingular isogeny graphs [JS19, Tan09]. Recently, two distinct quantum algorithms were proposed that leverage the quantum electrical flow state to show quantum advantages for finding an s-t path in graphs with a unique s-t path [JKP23] and graphs composed of welded Bethe trees [HL]. [JKP23] also presents a quantum algorithm that uses a quantum subroutine for path detection to find an s-t path. This quantum algorithm is faster than the quantum algorithm in [DHHM06] for graphs with all s-t paths being short. The speedups resulting from these works are polynomial and therefore fall short of the exponential speedup that we aim for.

We investigate the problem of finding an s-t path in exponential-size graphs. Instead of directly solving the welded tree pathfinding problem and the isogeny graph pathfinding problem directly, the goal of our research is to find more graph structures and develop new algorithms tools to show exponential quantum-classical separations for the task of finding an s-t path. Hopefully, the new findings of the graph structure and algorithmic tools will provide insight into solving some open problems.

We first construct a family of welded tree path graphs $\mathcal{G}_P$ that associate n distinct welded trees with a path length n as in Figure 4.1. Given the name of two vertices s,t in $\mathcal{G}_P$, the goal of the pathfinding problem is to output the names of vertices of an s-t path. Similarly to the welded tree pathfinding problem, there is an exponential number of s-t paths in the graph $\mathcal{G}_P$. On the other hand, the shortest s-t path is unique in the graph $\mathcal{G}_P$, while there is an exponential number of shortest paths in the welded tree graph. Using the distinctness of the welded trees in the graph $\mathcal{G}_P$, we show that there is an efficient quantum algorithm that solves this pathfinding problem.

Following the standard technique of problem reduction, the quantum algorithm works by reducing the problem of finding an s-t path to the problem of finding a sequence of marked vertices in n welded tree graphs. The quantum algorithm finds the edges of the

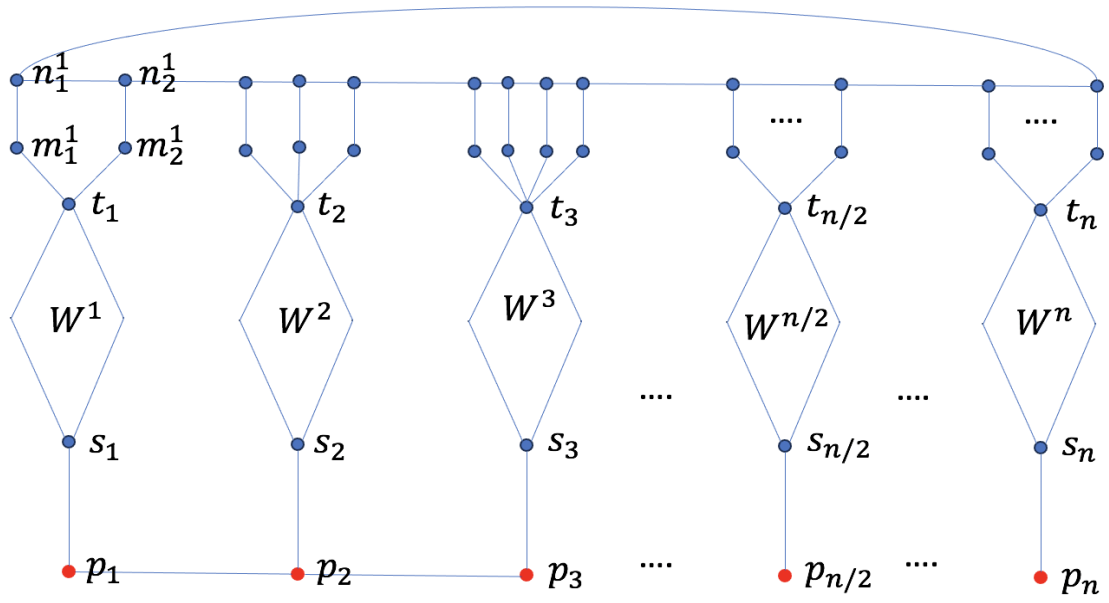

Figure 4.1. The Welded Tree Path Graph $\mathcal{G}_P$

s-t shortest path step by step. For each step, the polynomial-time continuous quantum walk algorithm [CCD$^+$03] for the welded tree problem is used to select one edge of the shortest s-t path. After n steps, the quantum algorithm outputs the shortest s-t path. The quantum algorithm for finding an s-t path in the welded tree path graph is heavily based on the fact that the graph is non-regular. The degree information is used to find an s-t path and show an exponential quantum-classical separation. This implies that this approach cannot be easily extended to regular graphs, where every vertex has the same degree.

We then construct a family of welded tree circuit graph $\mathcal{G}_C$ connecting the $3n$ welded tree graphs as in Fig. 4.2, where the graph is an almost regular graph with the exception of the starting vertex s and the ending vertex t. The formal definition is in Section 4.3.7. The welded tree circuit graph consists of n layers, where each layer consists of three welded trees. In addition, the welded tree circuit graph is a path graph if layers are viewed as vertices and the layers are connected through a critical edge with its neighboring layers. In order to show an exponential speedup on this type of regular graph, we develop a new multidimensional electrical network framework that is used to generate a quantum superposition state over edges in the graph, which we call an edge superposition approach.

For the edge superposition approach, we extend the classical electrical network into a multidimensional electrical network by generalizing Kirchhoff's law and Ohm's law. This

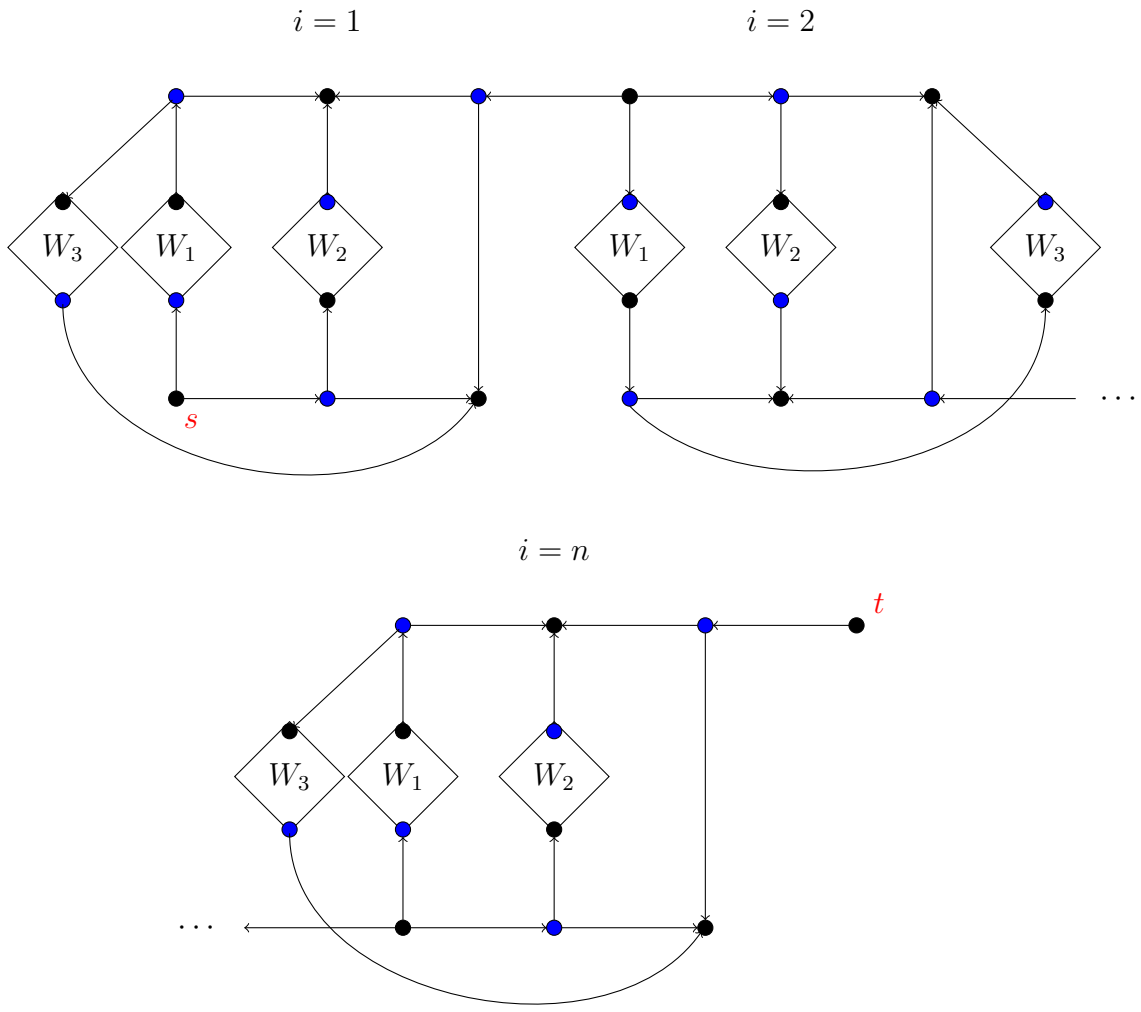

Figure 4.2. The Welded Tree Circuit Graph $\mathcal{G}_C$.

new framework allows our quantum algorithm to generate a quantum superposition state over edges that encodes an s-t flow, whereas the classical electrical network framework only encodes the s-t electrical flow. We utilize this quantum superposition state over edges to demonstrate an exponential quantum speedup for finding an s-t path in a welded tree circuit graph. In addition to showing an exponential speedup for a specific family of graphs with designated vertices s and t, this multidimensional electrical network framework could also be applied more broadly to develop new classical and quantum algorithms for various graph problems.

The quantum algorithms developed for the pathfinding problem in the welded tree path graph and the welded tree circuit graph do not provide much insight into finding

s-t paths in supersingular isogeny graphs or even expander graphs. The first quantum algorithm employing the problem reduction approach that demonstrated an exponential quantum speedup for the pathfinding problem on the welded tree path graph [Li23] relies heavily on the graph being non-regular because vertices with varying degrees are used as markers to help find the path. Therefore, it cannot be extended to find an s-t path in regular expander graphs. The second quantum algorithm uses a multidimensional electrical network framework to generate a quantum superposition state on the edges of the welded tree circuit graph [LZ23], which has a large overlap with an s-t path. Although this multidimensional electrical network framework has the potential to produce an exponential quantum speedup for the pathfinding problem in the welded tree graph or supersingular isogeny graph, analyzing the multidimensional electrical network in these graphs to demonstrate an exponential quantum advantage is challenging. In addition, the welded tree path graph and the welded tree circuit graph are far from being an expander graph. Specifically, the spectral gaps of the adjacency matrix of the welded tree path graph [Li23] and the welded tree circuit graph [LZ23] are $1/\operatorname{poly}(|V|)$. This is true because removing a constant number of edges would disconnect both graphs into subgraphs of size $\operatorname{poly}(|V|)$, implying that the conductance is at most $1/\operatorname{poly}(|V|)$. Here, the spectral gap is the difference between the largest eigenvalue and the second largest eigenvalue of the adjacency matrix, which is equal to the spectral gap around the 0 eigenvalue of the graph Laplacian.

We then find a new family of regular sunflower graph $\mathcal{G}_S$ that deviates from the welded tree structure and is close to being an expander graph. A regular sunflower graph consists of n trees of height m. For simplicity, we assume $m = \Theta(n)$. The roots $s_1, s_2, \ldots, s_n$ of the n trees $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$ are connected by the edges $\{s_1, s_2\}$, …, $\{s_{n-1}, s_n\}$, $\{s_n, s_1\}$ as a cycle. The leaves of trees $\mathcal{T}_i, \mathcal{T}_{i+1}$ for $i = 1, 2 \ldots, n-1$ and $\mathcal{T}_1, \mathcal{T}_n$ are connected by a random perfect matching such that the graph is regular. An example of the 3-regular graph is shown in Fig. 4.3. The formal definition of a regular sunflower graph is in Definition 4.4.5.

We show that, with high probability, the regular sunflower graph of degree at least 7 is a mild expander graph (defined in Definition 4.4.4). A mild expander graph is close to an expander graph in the sense that a random walk on a mild expander graph converges to a uniform distribution in $\operatorname{poly}\log(|\mathcal{V}|)$ time while the time needed for an expander graph is $O(\log|\mathcal{V}|)$. The spectral gap of the adjacency matrix of the mild expander graph is $1/\operatorname{poly}(\log|\mathcal{V}|)$ while the spectral gap of an expander graph is constant. We develop a new quantum approach for the pathfinding problem that generates a quantum

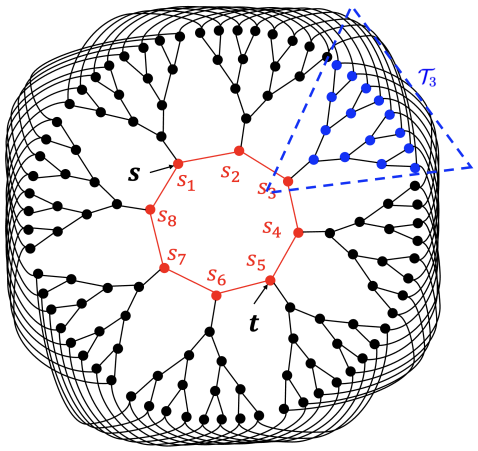

Figure 4.3. An example of the regular sunflower graph with $d = 3, m = 5, n = 8$. The s and t vertices are marked out. The tree within the dashed triangle is the subtree $\mathcal{T}_i$ (in this instance $i = 3$). The leaves of the trees $\mathcal{T}_i$ are connected via $(d-1)/2$ random perfect matchings.

superposition state over vertices to find an s-t path in the regular sunflower graph, which we call a vertex superposition approach.

For the vertex superposition approach, we apply the quantum eigenstate filtering algorithm within the QSVT framework to generate a quantum state that is an 0-eigenstate of the adjacency matrix of the graph. In addition to introducing a new family of graphs that enable exponential quantum speedups for pathfinding in the regular sunflower graph relative to an oracle, this approach presents two additional categories of contributions.

First, our quantum algorithm demonstrates an exponential separation for the pathfinding problem in a mild expander graph, raising the open question of whether this result can be extended for the pathfinding problem in the expander graphs, and even to certain classes of isogeny graphs. As a mild expander graph, the regular sunflower graph is close to being an expander graph since a random walk on both graphs can reach a uniform distribution in $\operatorname{poly}\log(|\mathcal{V}|)$ time. This connection provides the hope that this work might be useful for developing new algorithms for the non-oracle pathfinding problem in supersingular isogeny graphs, where the adjacency list oracle can be instantiated by computing isogenies between two elliptic curves.

Second, our new quantum algorithm, which leverages the 0-eigenspace of the adjacency matrix, provides insights to attack a wider range of problems, including the pathfinding problem in the welded tree graph. Specifically, our quantum algorithm lies outside the class of algorithms considered by Childs, Coudron, and Gilani to solve the welded tree pathfinding problem [CCG22], suggesting that the excluded algorithms may be more useful than previously anticipated. Furthermore, existing results on spectral

graph theory, including those for graphs related to the welded tree graph [GS21, Figure 2], may be combined with our algorithm for wider applications, including providing a potential approach to attack the welded tree pathfinding problem.

We also show that no classical algorithm can solve these pathfinding problems in welded tree path graphs, welded tree circuit graphs, and regular sunflower graphs in subexponential time. The proof of these classical lower bounds essentially follows the lower bound proof of the welded tree graph in [CCD+03].

We summarize the graph structures and quantum approaches that demonstrate exponential quantum-classical separations for pathfinding-related problems in the following Table 4.1:

Graphs	Input	Output	Techniques
Welded Tree Graph [CCD+03]	s	t	CQW
Random Hierarchical Graph [BLH23]	s	t	CQW
Welded Tree Path Graph $\mathcal{G}_P$ [Li23]	s	s-t path	CQW
Welded Tree Circuit Graph $\mathcal{G}_C$ [LZ23]	s	s-t path	MEN
Regular Sunflower Graph $\mathcal{G}_S$ [LT24]	s	s-t path	QEF

Table 4.1. CQW: Continuous-Time Quantum Walk, MEN: Multidimensional Electrical Network, QEF: Quantum Eigenstate Filtering.

Organization: In Section 4.2, we give the definition of the welded tree path graph and then provide an efficient quantum algorithm for finding an s-t path in the welded tree path graph. The quantum algorithm is based on the problem reduction technique, which reduces the problem of finding an s-t path in the welded tree path graph to the problem of finding a marked vertex in the welded tree graph. In Section 4.3, we first extend the electrical network framework to the multidimensional electrical network by redefining Kirchhoff's Law and Ohm's Law as alternative Kirchhoff's Law and alternative Ohm's Law. This new definition follows the alternative neighborhoods technique introduced by Jeffery and Zur in their multidimensional quantum walk framework [JZ23]. We then use this multidimensional electrical network framework to show an exponential quantum-classical separation for finding an s-t path in the welded tree circuit graph. In Section 4.4, we first introduce the definition of the regular sunflower graph and its graph expansion properties. Then we provide an efficient quantum algorithm for for finding an s-t path in the regular sunflower graph. The key idea of the quantum algorithm is to generate a quantum superposition state over vertices that encode a 0-eigenvector of the adjacency matrix of the regular sunflower graph. In Section 4.5, we establish the classical lower bounds of the pathfinding problems in the aforementioned three types of

graphs and show that no classical algorithm can solve these problems efficiently.

4.2 Problem Reduction

In this section, we first introduce the main result of the continuous quantum walk algorithm for the welded tree problem. Then, using the welded tree graph as a building block, we give the definitions of the welded tree path graph and the associated pathfinding problem. We then provide an efficient quantum algorithm for the pathfinding problem, which uses the quantum algorithm for the welded tree problem as a key subroutine.

It should be noted that several quantum algorithms have been developed to solve the welded tree problem, encompassing techniques such as continuous quantum walks [CCD$^+$03], multidimensional quantum walks [JZ23], discrete quantum walks [LLL24] and the coupled classical oscillator approach [BBK$^+$23]. Here, we specifically focus on the original continuous quantum walk approach for the welded tree problem.

Definition 4.2.1 (Welded Tree Graph W)**.** *A* welded tree graph W *consists of two balanced binary trees of height n with roots s and t and a random cycle that alternates between the leaves of the two binary trees.*

The number of vertices in W is $2^{n+2}-2$ and the names of the vertices are randomly assigned from the set $\{0,1\}^{2n}$. To access the neighbors of a particular vertex, we use an adjacency list oracle denoted as O for the graph G. Given a $2n$-bit string $u \in \{0,1\}^{2n}$, the adjacency list oracle O provides the neighboring vertices of u, or it returns $\perp$ if u is not a valid vertex name in the graph.

Problem 4.2.1 (The welded tree problem)**.** *Given an adjacency list oracle O of the welded tree W and the name of the starting vertex $s \in \{0,1\}^{2n}$, output the name of the other root t.*

Let A be the adjacency matrix of a graph G, τ be a real number, and $|\psi_0\rangle = |s\rangle$ as the initial input state, the continuous-time quantum walk is defined as

$$|\psi_\tau\rangle = e^{-iA\tau}\,|\psi_0\rangle\,.$$

Lemma 4.2.2 (Theorem 3 in [CCD$^+$03])**.** *Let A be the adjacency matrix of the welded tree graph W. With the adjacency list oracle O of the welded tree W, the name of the starting vertex $s \in \{0,1\}^{2n}$ and let $|\psi_0\rangle = |s\rangle$, running the continuous quantum walk $|\psi_\tau\rangle = e^{-iA\tau}\,|\psi_0\rangle$ for a time τ chosen uniformly in $[0, n^5]$ and then measuring on the*

computational basis yields a probability of finding the name of the other root vertex t that is greater than $\Omega(\frac{1}{n})$.

By querying the adjacency list oracle O of a given vertex, one can determine whether it has a degree of 2 or not. Since only two vertices, s and t, have a degree of 2 in the welded tree W, repeating the quantum algorithm can find the other root t in poly(n) time. Furthermore, no classical algorithm can solve the welded tree problem in subexponential time by the following lemma.

Lemma 4.2.3 (Theorem 9 in [CCD$^+$03])**.** *For the welded tree problem, any classical algorithm that makes at most* $2^{n/6}$ *queries to the oracle finds the ending vertex or a cycle with probability at most* $4 \cdot 2^{-n/6}$.

Note that the exact statement of Theorem 9 in [CCD$^+$03] does not mention finding a cycle, but the classical lower bound result also holds for this easier case, as stated in their proof.

4.2.1 The Welded Tree Path Graph

The key idea to construct the welded tree path graph $\mathcal{G}_P$ is to associate an s-t path $\{p_1, p_2, \cdots, p_n\}$ of length n with n distinct welded trees. The quantum algorithm finds the s-t path by reducing it to find a marked vertex in the welded tree graph. For each step, we use the quantum algorithm for the welded tree problem [CCD$^+$03] to detect and select the edge $\{s, u_i\}$ that is in the path $\{p_1, p_2, \cdots, p_n\}$. Then we remove the selected edge and update the starting vertex $s = u_i$. Repeat until $s = t$ and output all selected edges as an s-t path.

Definition 4.2.2 (Welded Tree Path Graph $\mathcal{G}_P$)**.** *Let n be an even integer. Given n disjoint welded trees W^i with roots s_i and t_i, the graph $\mathcal{G}_P$ is constructed as follows:*

1. *For each $i \in [n]$, adding $i+1$ new isolated edges $e_k = (m_k^i, n_k^i), k \in [i+1]$, then adding $i+1$ new edges between t_i and vertices m_k^i such that the root vertex t_i has degree $i+3$.*

2. *Given a path graph P_n with n vertices $p_1, \cdots, p_n$, for each $i \in [n]$, add an edge between p_i and s_i .*

3. *Adding a random cycle between all vertices n_k^i for each $i \in [n]$ and $k \in [i+1]$. Denote the resulting graph as the welded tree path graph $\mathcal{G}_P$.*

Let $s = p_1$ be the starting vertex, $t = p_n$ be the ending vertex, and given the names of s and t, the goal is to find an s-t path in the welded tree path graph $\mathcal{G}_P$ (Fig. 4.1).

Problem 4.2.4 (Pathfinding problem in $\mathcal{G}_P$)**.** *Given the adjacency list oracle O of the welded tree path graph $\mathcal{G}_P$ and the name of the starting vertex $s \in \{0,1\}^{2n}$, output an s-t path in $\mathcal{G}_P$.*

4.2.2 The algorithm

Algorithm 2 Quantum algorithm for finding an s-t path in the graph $\mathcal{G}_P$

Input: Graph $\mathcal{G}_P = (V, E)$, $s, t \in V$ and $i = 1$.
Output: an s-t path

1. Given the name of the vertex $s \in \{0,1\}^{2n}$, the adjacency list oracle O returns the names of the two neighbors of s, that is $u_1, u_2 \in \{0,1\}^{2n}$. Without loss of generality, pick one of the two neighbors u_1 as the initial state $|u_1\rangle$.

2. Let A' be the modified adjacency matrix of the adjacency matrix A of G by removing all the edges adjacent to a degree 2 vertex. Let O' be O except it returns no edge if one endpoint of the edge is a degree 2 vertex. Run the continuous quantum walk $e^{-iA'\tau}|u_1\rangle$ for a uniform random time $\tau \in [0, n^5]$. Measure the resulting state in the computational basis and get the name of an outcome vertex $v \in \{0,1\}^{2n}$. Compute the degree of the vertex v by querying the adjacency list oracle O of G.

3. Repeat Step 2 n^2 times or until the degree of the measured vertex equals $i + 3$.

4. If the degree of the measured vertex v equals $i + 3$, then collect the edge (s, u_2) as an s-t path edge and let $s = u_2$. Otherwise, collect the edge (s, u_1) as an s-t path edge and let $s = u_1$.

5. Let i=i+1 and update the graph G by deleting the selected edge. Repeat all the above steps until $s = t$, then output all the selected edges as an s-t path.

Theorem 4.2.5. *With high probability, Algorithm 2 outputs an s-t path in* $\mathrm{poly}(n)$ *time.*

Proof. For the i-th iteration, the key subroutine is to implement the continuous quantum walk $e^{-iA'\tau}$ for $\tau \in [0, n^5]$ in Step 2. The adjacency list oracle O' can be performed by removing all edges adjacent to a degree 2 vertex from the adjacency list oracle O. Then use the adjacency list oracle O' to construct a block encoding of the modified

adjacency matrix A'. With this block encoding, we can perform the Hamiltonian simulation $e^{-iA'\tau}|u_1\rangle$ in poly(n) time by implementing polynomials of A' as indicated by Theorem 2.3.2 in Section 2.3.

Recall that the only vertex that has degree $i+3$ is the root t_i of the welded tree W^i in the graph G. If u_1 is the same as the root vertex s_i of the welded tree W^i, then by Lemma 4.2.2, the success probability of measuring a vertex of degree $i+3$ is $\Omega(1/n)$. The probability of success is $1-\exp(-\Omega(n))$ by repeating Step 2 n^2 times. Otherwise, the probability of measuring a vertex of degree $i+3$ is 0. This is true because the initial starting vertex $|u_1\rangle$ is disconnected from the welded tree $W^1, \cdots, W^i$ and connected to the welded trees $W^{i+1}, \cdots W^n$ in the modified graph associated with the adjacency list oracle O'. Therefore, for each iteration, Algorithm 2 outputs an edge of the s-t path with probability $1-\exp(-\Omega(n))$ in poly(n) time.

The length of the s-t path outputted by Algorithm 2 is n, the same as the total number of iterations. Therefore, with high probability, Algorithm 2 outputs an s-t path in poly(n) time. □

4.3 Edge Superposition

Recently, Apers and Piddock [AP22] strengthened the natural connection between quantum walks and electrical networks by considering Kirchhoff's Law and Ohm's Law. In this section, we develop a new multidimensional electrical network by defining Alternative Kirchhoff's Law and Alternative Ohm's Law based on the multidimensional quantum walk framework by Jeffery and Zur [JZ23]. In analogy to the connection between the (edge-vertex) incidence matrix of a graph and Kirchhoff's Law and Ohm's Law in an electrical network, we rebuild the connection between the alternative incidence matrix and Alternative Kirchhoff's Law and Alternative Ohm's Law. This multidimensional electrical network framework enables generating an alternative electrical flow over the edges of graphs, which has the potential to be applied to a broader range of graph problems, benefiting both quantum and classical algorithm design.

We first use this framework to generate quantum alternative electrical flow states and use it to find a marked vertex in one-dimensional random hierarchical graphs as defined by Balasubramanian, Li, and Harrow [BLH23]. In this work, they generalized the well-known exponential quantum-classical separation of the welded tree graph by Childs, Cleve, Deotto, Farhi, Gutmann, and Spielman [CCD$^+$03] to random hierarchical graphs. We partially recover their results with an arguably simpler analysis within the

multidimensional electrical network framework.

Furthermore, by constructing a family of regular graphs from the welded tree, named the welded tree circuit graph $\mathcal{G}_C$, this framework allows us to generate alternative electrical flow states and sample from these to exhibit an exponential speedup for the pathfinding problem in an almost regular graph under the folklore assumption that finding an s-t path in the welded tree graph is hard. Compared with an exponential speedup for the pathfinding problem in the welded tree path graph, this new multidimensional electrical network framework enables a new type of quantum algorithm for finding an s-t path by generating a quantum superposition state over the edges and achieving exponential speedups.

The remainder of this section is organized as follows. In Section 4.3.1, we provide an overview of the multidimensional electrical network and its applications. In Section 4.3.2 we show how the concepts related to electrical flow can be generalized to the multidimensional electrical network under the multidimensional quantum walk framework. This results in our new Alternative Kirchhoff's Law and Alternative Ohm's Law. In addition, we rebuild the connection between the alternative incidence matrix and Alternative Kirchhoff's Law and Alternative Ohm's Law, showing that our new laws seem to be natural definitions and that they generalize known results regarding electrical networks. In Section 4.3.6, we apply the multidimensional electrical network to one-dimensional random hierarchical graphs and show how the framework allows us to sample exponentially faster from the electrical flow state than any classical algorithm. In Section 4.3.7, using the multidimensional electrical network, we construct the welded tree circuit graph $\mathcal{G}_C$ where we show that quantum walks can exhibit exponential speedups when it comes to pathfinding problems.

4.3.1 Overview of the multidimensional electrical network and its applications

The connection between random walks, electrical networks, and Laplacian linear systems is of great significance in multiple aspects of theoretical computer science. When the edges of a graph are seen as resistors, the effective resistance between two vertices in an electrical network plays an important role in analyzing the behavior of random walks (or more general Markov chains) on undirected graphs, such as their hitting time and cover time [DS84, Lov96]. By restating Kirchhoff's Law and Ohm's Law in electrical networks (graphs) in terms of the graph's Laplacian matrix and incidence matrix, one

can compute the effective resistance and electrical flow between two vertices by solving a Laplacian linear system [Vis13]. In addition, the effective resistance and electrical flow used in electrical networks have been used to design new graph algorithms, such as graph sparsification via effective resistances [SS08], computing max flow via electrical flow [CKM$^+$11], and many more [Mad13, GLP23]. Furthermore, the mixing time of a random walk (Markov chain) on a connected graph is related to the smallest nonzero eigenvalue of the Laplacian matrix of the underlying graph [Spi19].

The connection between (discrete) quantum walks, electrical networks, and Laplacian linear systems is less understood and has evolved separately. Specifically, the connection between quantum walks and electrical networks was established for the first time by [Bel13], where this connection was used to derive and analyze a phase estimation algorithm with the goal of detecting the existence of a marked vertex in a graph. Not until recently, this connection has been strengthened by Apers and Piddock [Pid19, AP22] by considering Kirchhoff's Law and Ohm's Law. They showed that for the quantum walk operator based on the electrical network framework, if the phase value returned by the phase estimation algorithm from [Bel13] is "0", indicating that there is a marked vertex, then the resulting state is actually a quantum state representing the electrical flow between the starting vertex and the marked vertex. Separately, [Wan17] provided a quantum algorithm based on the Quantum Linear System (QLS) [GSLW19] algorithm to solve a Laplacian-related linear system in the analysis of large electrical networks, such as computing their electrical flow and effective resistance between two vertices. However, there is currently little known about the connection between quantum walks and the eigenvalues of the Laplacian matrix of the underlying graph.

There are multiple examples of quantum walk based algorithms that can solve certain graph problems exponentially faster than any classical algorithm with oracle access to the graph, but currently none of them are based on the electrical network framework. For example, continuous quantum walks have been used to exhibit an exponential quantum-classical separation to solve the welded tree problem [CCD$^+$03]. A welded tree graph consists of two full binary trees of depth n and the leaves of both trees are connected via two disjoint perfect matching. There are a total number of $2^{n+2} - 2$ vertices. Given an adjacency list oracle O_G to the welded tree graphs G and the name of one of the roots s, the goal of the welded tree problem is to output the name of the other root t. This exponential separation result of the welded tree problem has recently been recovered (and improved) by applying the multidimensional quantum walk framework [JZ23] or using the conceptually simpler coined quantum walk [LLL23]. In addition, using a more refined

analysis of the continuous quantum walk algorithm, [BLH23] generalized this exponential separation result on the welded tree graph to certain random hierarchical graphs where the goal is once again to find some special vertex t given an initial vertex s. The idea of finding a marked vertex in certain graphs has also been used to show the exponential advantage of adiabatic quantum computation with no sign problem in [GHV21], whose graph is also related to the welded tree graph. Another graph constructed from welded tree graphs has also allowed for an exponential quantum-classical separation for a type of graph property testing problem [BDCG$^+$20].

Among the three types of quantum walk design paradigms mentioned that achieve this exponential speedup (continuous quantum walks, coined quantum walks, and multidimensional quantum walks), the multidimensional quantum walk has the closest relationship with the quantum walk operator in an electrical network. However, it is far from obvious how one can combine the design paradigm of multidimensional quantum walks, with Kirchhoff's Law and Ohm's Law in an electrical network.

Multidimensional electrical network We take the first steps in establishing this connection, which we name the multidimensional electrical network framework, achieved through generalizing Kirchhoff's Law as Alternative Kirchhoff's Law and Ohm's Law as Alternative Ohm's Law. Roughly speaking, for each vertex other than the source and sink, Kirchhoff's Law forces the electrical flow to be orthogonal to a single vector, whereas Alternative Kirchhoff's Law forces the electrical flow to be orthogonal to a potentially larger subspace. Moreover, instead of associating each vertex with a single (unique) potential value for each vertex as in Ohm's Law, in Alternative Ohm's Law we associate (possibly distinct, but unique) potential values $p_{(u,v)}$, $p_{(v,u)}$ to each edge (u, v). The ideas behind the definition of Alternative Kirchhoff's Law and Alternative Ohm's Law in the multidimensional electrical network are derived from the quantum walk based on an electrical network [Bel13, Pid19, AP22] and the alternative neighborhood technique introduced by the multidimensional quantum walk [JZ23].

Similarly to the quantum walk frameworks that are based on electrical networks, we model a graph $G = (V, E, \mathsf{w})$ as an electrical network with each edge assigned a positive weight $\mathsf{w}_{u,v}$, that is, its conductance. This weight assignment gives rise to a weighted superposition of neighbors in the neighborhood $\Gamma(u)$ of any vertex u, normalized by the quantity w_u, which is known as the *star state* of u:

$$|\psi_u\rangle := \frac{1}{\sqrt{\mathsf{w}_u}} \sum_{v \in \Gamma(u)} (-1)^{\Delta_{u,v}} \sqrt{\mathsf{w}_{u,v}}\, |u, v\rangle\,. \tag{4.1}$$

We study s-t flows on our graph G between any two of its vertices s and t. An s-t flow assigns an amount of value of flow along each edge of the graph G. Kirchhoff's Law states that any s-t flow is conserved at every vertex $u \in V \backslash \{s,t\}$, meaning that the amount of flow entering any vertex u is equal to the amount of flow exiting u. More specifically, we are interested in the *s-t electrical flow* θ, which is the "smallest" s-t flow, meaning it has the smallest energy out of all valid s-t flows satisfying Kirchhoff's Law:

$$\mathcal{E}(\theta) := \sum_{(u,v)\in\overrightarrow{E}} \frac{\theta_{u,v}^2}{\mathsf{w}_{u,v}}.$$

The energy of the s-t electrical flow θ is called the *effective resistance* $\mathcal{R}_{s,t}$ and this flow gives rise to the normalized electrical flow state $|\theta\rangle$:

$$|\theta\rangle := \frac{1}{\sqrt{2\mathcal{R}_{s,t}}} \sum_{(u,v)\in\overrightarrow{E}} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}} \left(|u,v\rangle + |v,u\rangle\right).$$

By viewing G as an electrical network, this electrical flow precisely captures the electrical dynamics of how one unit of current send from s to t would traverse the electrical network G. Kirchhoff's Law states that this s-t electrical flow is conserved at every vertex $u \in V \backslash \{s,t\}$, meaning that the amount of flow θ coming into u is equal to the amount of flow exiting u. This law can be equivalently read in terms of $|\psi_u\rangle$ and $|\theta\rangle$, in which case it states that for every vertex $u \in V \backslash \{s,t\}$ we require

$$\langle \psi_u | \theta \rangle = 0.$$

The quantum walk operator $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$ in [Bel13] consists of a reflection around two spaces: the antisymmetric subspace $\mathcal{A}$ and the span of (almost all) star states:

$$\mathcal{B} := \text{span}\{|\psi_u\rangle : u \in V \backslash \{s,t\}\}.$$

The s-t electrical flow θ is special in regard to this quantum walk operator $U_{\mathcal{AB}}$, as its flow state $|\theta\rangle$ lives in the $+1$-eigenspace of U. Moreover, it can also be written as a linear combination of projected star states $(I - \Pi_{\mathcal{A}})|\psi_u\rangle$ for $u \in V$. The coefficients in this linear combination are precisely the potentials p_u given by the *potential vector* p corresponding to the s-t electrical flow θ such that together they satisfy Ohm's Law: $p_u - p_v = \theta_{u,v}/\mathsf{w}_{u,v}$. By combining all these properties, [Pid19, AP22] showed that with the use of phase estimation on the quantum walk operator $U_{\mathcal{AB}}$, one can approximate

$|\theta\rangle$ and measure it, allowing one to (approximately) sample from the s-t electrical flow θ, where an edge is obtained with a probability equal to the relative amount of flow that traverses this edge.

The exact complexity of approximating the s-t electrical flow state in an electrical network is given in Theorem 2.5.2, but it depends (apart from the approximation error) on two factors: the effective resistance $\mathcal{R}_{s,t}$ and the norm of the potential vector $|p\rangle$, defined as

$$|p\rangle = \sqrt{\frac{2}{\mathcal{R}_{s,t}}} \sum_{u \in V \setminus \{s,t\}} p_u \sqrt{\mathsf{w}_u} \, |\psi_u\rangle \, .$$

The multidimensional quantum walk framework is a generalization of the previously known quantum walks based on electrical networks and works by running phase estimation on the modified quantum walk operator $U_{\mathcal{A}\mathcal{B}^{\mathsf{alt}}}$, which reflects around the larger space that contains $\mathcal{B}$:

$$\mathcal{B}^{\mathsf{alt}} := \mathrm{span}\{\mathrm{span}(\Psi_\star(u)) : u \in V \setminus \{s,t\}\}.$$

By associating each vertex u with a subspace $\mathrm{span}(\Psi_\star(u))$, the multidimensional quantum walk has shown advantages in solving certain problems, such as the welded tree problem and the k-distinctness problem in [JZ23] faster than the quantum walk. The algorithm works by applying phase estimation of the multidimensional quantum walk operator $U_{\mathcal{A}\mathcal{B}^{\mathsf{alt}}}$ and checking whether the returned phase value is "0". Especially for the welded tree problem, this phase value being "0" indicating that the root t is "marked", which allows them to infer the name of t, which recovers the exponential quantum-classical separation for finding a marked vertex in the welded tree graph using a continuous quantum walk [CCD$^+$03].

The reason why multidimensional quantum walks achieve this potentially exponential speedup, whereas other discrete quantum walks can achieve up to a quadratic speedup compared to the underlying classical random walk, has to do with the cost of calling $U_{\mathcal{A}\mathcal{B}}$. The cost of applying this quantum walk operator, and hence the cost of this phase estimation procedure, is based on the cost of generating the star state $|\psi_u\rangle$. In [JZ23], the authors deal with the case where it might be computationally costly to generate $|\psi_u\rangle$, but where the algorithm knows that $|\psi_u\rangle$ is one of a small set of easily preparable states $\Psi_\star(u) = \{|\psi_{u,1}\rangle, |\psi_{u,2}\rangle, \dots\}$, known as the *alternative neighborhoods* for u. To give some intuition as to why this might be useful, let's look at the definition of $|\psi_u\rangle$ in Eq. (4.1) again. If different neighbors $v_1, v_2 \in \Gamma(u)$ have different amplitudes in $|\psi_u\rangle$,

perhaps due to different weights/conductances $\mathsf{w}_{u,v_1}, \mathsf{w}_{u,v_2}$, or perhaps a different sign due to $\Delta_{u,v_1}, \Delta_{u,v_2}$, than to generate the star state $|\psi_u\rangle$, it would be necessary to distinguish the neighbors v_1, v_2 from each other. This might be computationally costly if we can access G via an adjacency list oracle, where by querying u we learn $\Gamma(u)$. However, such problems can also arise in a much more nuanced fashion outside of the oracle model, as exhibited in the algorithm tackling the k-distinctness problem in [JZ23]. Therefore, by generating a reflection around $\Psi_\star(u)$, instead of $|\psi_u\rangle$, the cost of each step of the phase estimation algorithm is potentially drastically reduced.

To extend the previously discussed relationship between electrical networks and quantum walks to this new quantum walk operator $U_{\mathcal{AB}^{\mathsf{alt}}}$, we introduce the s-t alternative electrical flow θ^{alt}. This is again the "smallest" s-t flow in terms of energy that satisfies Alternative Kirchhoff's Law, which requires $|\theta^{\mathsf{alt}}\rangle$ to be orthogonal to all of $\mathrm{span}(\Psi_\star(u))$ instead of only to $|\psi_u\rangle \in \Psi_\star(u)$, ensuring that $|\theta^{\mathsf{alt}}\rangle$ lives in the $+1$-eigenspace of $U_{\mathcal{AB}^{\mathsf{alt}}}$. This $|\theta^{\mathsf{alt}}\rangle$, whose energy is the *alternative effective resistance* $\mathcal{R}^{\mathsf{alt}}_{s,t}$, can also be written as a linear combination of projected alternative neighborhoods $(I - \Pi_{\mathcal{A}})|\psi_{u,i}\rangle$ for $u \in V$ and $|\psi_{u,i}\rangle \in \Psi_\star(u)$. As we have seen in the non-alternative neighborhood case, this gives rise to an alternative potential vector p^{alt}, this time not acting on vertices, but on edges instead. This p^{alt} is related to θ^{alt} through Alternative Ohm's Law, which states that $p^{\mathsf{alt}}_{u,v} - p^{\mathsf{alt}}_{v,u} = \theta^{\mathsf{alt}}_{u,v}/\mathsf{w}_{u,v}$. By a similar analysis as in regular electrical networks, we show that with phase estimation on the quantum walk operator $U_{\mathcal{AB}^{\mathsf{alt}}}$ we can approximate $|\theta^{\mathsf{alt}}\rangle$, allowing one to (approximately) sample from the s-t alternative electrical flow θ^{alt}.

Theorem 4.3.1. *Let $\Psi_\star$ be a collection of alternative neighborhoods on a network $G = (V, E, \mathsf{w})$ and let $U_{\mathcal{AB}^{\mathsf{alt}}}$ be the quantum walk operator with respect to $\Psi_\star$ as defined in Eq.* (4.2)*. Then by performing phase estimation on the initial state $|\psi^+_s\rangle$ with the operator $U_{\mathcal{AB}^{\mathsf{alt}}}$ and precision $O\left(\frac{\epsilon^2}{\sqrt{\mathcal{R}^{\mathsf{alt}}_{s,t}\mathsf{w}_s}\|p^{\mathsf{alt}}\|}\right)$, the phase estimation algorithm outputs "0" with probability $\Theta\left(\frac{1}{\mathcal{R}^{\mathsf{alt}}_{s,t}\mathsf{w}_s}\right)$, leaving a state $|\theta'\rangle$ satisfying*

$$\frac{1}{2}\left\||\theta'\rangle\langle\theta'| - |\theta^{\mathsf{alt}}\rangle\langle\theta^{\mathsf{alt}}|\right\|_1 \leq \epsilon.$$

As can be seen in the above theorem, the complexity of generating the s-t alternative electrical flow state in a multidimensional electrical network depends on the alternative versions of the effective resistance and the norm of the potential vector. These are $\mathcal{R}^{\mathsf{alt}}_{s,t}$

and the norm of the alternative potential vector

$$|p^{\mathsf{alt}}\rangle = \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{(u,v)\in\overrightarrow{E}:s,t\notin(u,v)} \sqrt{\mathsf{w}_{u,v}}(p^{\mathsf{alt}}_{u,v}\,|u,v\rangle - p^{\mathsf{alt}}_{v,u}\,|v,u\rangle).$$

With this new multidimensional electrical network framework, more specifically due to Theorem 4.3.1, we show that quantum algorithms can provide exponential quantum-classical separations for certain graph problems, such as finding a marked vertex and finding an s-t path in some special graphs related to the welded tree graph. On the other hand, all the previous quantum algorithms based on the electrical network can provide only a quadratic speedup.

Our algorithms work by approximate sampling from the alternative electrical flow θ^{alt}, meaning we approximate the state $|\theta^{\mathsf{alt}}\rangle$ using Theorem 4.3.1 and then measure it to obtain an edge with probability scaling with the amount of alternative flow that traverses this edge. In both of these applications, the way that $\Psi_\star$ is extended is fairly natural, as it will be constructed from Fourier basis states.

Alternative incidence matrix and alternative electrical flow These new alternative electrical network laws and definitions may seem constructed in an ad-hoc fashion to fit with the analysis in [Pid19, AP22], but we give proof that these are in fact natural definitions. It is well known in electrical network theory [Vis13] that both Kirchhoff's Law as well as Ohm's Law can be phrased as linear equations involving the edge-vertex incidence matrix B, whose entries contain the square root of the weights $\mathsf{w}_{u,v}$. These linear relations are useful to show important physical properties of the s-t electrical flow θ and its potential vector p, such as their existence and the fact that p_s is equal to the energy of θ, also known as the effective resistance $\mathcal{R}_{s,t}$. By extending this incidence matrix B in a natural fashion to also incorporate the alternative neighborhoods $\Psi_\star(u)$, we obtain the alternative incidence matrix B_{alt}. For the actual technical definition of B_{alt}, see Definition 4.3.7, but for now B_{alt} can be thought of as a generalized version of B. This matrix B_{alt} can then be substituted for B in the linear equations that correspond to Kirchhoff's Law and Ohm's Law, to recover Alternative Kirchhoff's Law and Alternative Ohm's Law respectively. As is done with regular electrical networks, we apply these linear equations to prove the existence of the s-t alternative electrical flow θ^{alt} and its alternative potential vector p^{alt}, as well as the fact that the potential $p_{s,u}$ along each adjacent to s is equal to the energy of θ^{alt}, which we call the alternative effective resistance $\mathcal{R}^{\mathsf{alt}}_{s,t}$. To summarize, we prove the following two results:

Theorem 4.3.2. *Let θ^{alt} be the s-t alternative electrical flow in an electrical network $G = (V, E, \mathsf{w})$ with respect to a collection of alternative neighborhoods $\Psi_\star$. Let B_{alt} be the alternative incidence matrix of G. Then $W\theta^{\mathsf{alt}}$ is given by*

$$W\theta^{\mathsf{alt}} = B_{\mathsf{alt}}^{T+}(\mathsf{e}_s - \mathsf{e}_t),$$

where $W \in \mathbb{C}^{\vec{E}\times\vec{E}}$ is the diagonal matrix with entries $W_{(u,v),(u,v)} = 1/\sqrt{\mathsf{w}_{u,v}}$.

Theorem 4.3.3. *Let θ^{alt} be the s-t alternative electrical flow in an electrical network $G = (V, E, \mathsf{w})$ with respect to a collection of alternative neighborhoods $\Psi_\star$. Then there exists an alternative potential vector p^{alt} satisfying Alternative Ohm's Law such that $p^{\mathsf{alt}}_{s,u} = \mathcal{R}^{\mathsf{alt}}_{s,t}$ and $p^{\mathsf{alt}}_{t,v} = 0$ for each $u \in \Gamma(s)$ and $v \in \Gamma(t)$.*

The perspective of the alternative incidence matrix of the multidimensional electrical network provides a new way to generate the alternative electrical flow by solving the alternative incidence linear system as indicated in Theorem 4.3.2.

Classically, solving such an alternative incidence linear system takes a polynomial time in the size of the graph to get the alternative electrical state flow vector. The electrical flow obtained by solving the related Laplacian linear system has recently played an important role in designing new graph algorithms for max-flow problems [CKM$^+$11, Mad13, GLP23]. It is conceivable that our alternative electrical flow could potentially be applied to design new graph algorithms for certain graph problems.

Quantumly, the complexity of solving the linear system to obtain the quantum alternative electrical flow state depends on the condition number of the alternative incidence matrix on a bounded degree graph [GSLW19]. Although in this work we focus on using multidimensional quantum walk to generate the *s-t* alternative electrical flow state $|\theta^{\mathsf{alt}}\rangle$ for our applications, this new way of generating $|\theta^{\mathsf{alt}}\rangle$ through the QLS algorithm could lead to new applications of the multidimensional electrical network for more types of problems.

Finding a marked vertex in random hierarchical graphs We first apply this multidimensional quantum electrical network framework to one-dimensional random hierarchical graphs with nodes $S_0, S_1, \cdots, S_n$, as defined in [BLH23]. Given the initial vertex s, which is the unique element in S_0, the goal is to transverse the exponentially large (in n) one-dimensional random hierarchical graph to find the vertex t, the unique element in S_n.

It has been shown that the continuous quantum walk approach can provide an exponential speedup (in n) in solving this problem compared to any classical algorithm

with certain additional assumptions on the structures of the one-dimensional random hierarchical graph. After applying minor differences to the assumptions regarding the structure of the graph, we show that multidimensional quantum walks also solve this problem in polynomial time by sampling from the quantum electrical flow state.

Instead of applying Theorem 4.3.1 directly, we describe and analyze our algorithm explicitly to provide more insight on how multidimensional quantum walks can be used to approximate the alternative electrical flow.

Interestingly, in this one-dimensional setting, the alternative electrical flow with respect to the multidimensional quantum walk operator on a one-dimensional unweighted random hierarchical graph matches the actual electrical flow of a weighted electrical network. We apply this result to the welded tree graph, which is an example of a one-dimensional random hierarchical graph, to provide an alternative quantum algorithm that solves the problem in polynomial time, recovering the exponential separation from [CCD$^+$03].

In the one-dimensional random hierarchical unweighted graph with an adjacency list oracle access of G, it is computationally costly to generate $|\psi_u\rangle$, but where the algorithm knows that $|\psi_u\rangle$ is one of a small set of easily preparable states $\Psi_\star(u) = \{|\psi_{u,1}\rangle, |\psi_{u,2}\rangle, \dots\}$, known as the *alternative Fourier neighborhoods* for u (see Definition 4.3.14). With this multidimensional quantum walk operator, we show that the cost of generating the quantum alternative electrical flow state is a polynomial, while the cost of generating the quantum electrical flow state is exponentially large.

Compared to the technical analysis from [BLH23] used for the continuous quantum walk approach, which contains more concepts from physics, our analysis is more suited for a computer science audience and arguably simpler.

Pathfinding problem in welded tree circuit graph $\mathcal{G}_C$ While there are known efficient quantum algorithms [CCD$^+$03, JZ23, LLL24] for finding the marked vertex in the welded tree graph, the problem of finding an s-t path in the welded tree graph is notably difficult and still an open problem in the field of quantum query complexity [Aar21]. In addition, the problem of finding an s-t path in isogeny graphs (a class of expander graphs) is assumed to be hard even for quantum algorithms, as the security of many isogeny-based cryptosystems is based on the hardness of solving this problem [CLG09, CFL$^+$18, Wes22].

Instead of finding an s-t path in the welded tree graph or isogeny graphs, Section 4.2 constructed a non-regular graph and exhibits an exponential quantum-classical separation in the context of pathfinding problems. The non-regular graph associating n

different welded tree graphs with an s-t path of length n. This quantum algorithm uses the polynomial-time continuous quantum walk algorithm from [CCD$^+$03] as a subroutine to output the s-t path and the quantum algorithm relies heavily on the constructed graph being non-regular. This degree information in some way propagates the algorithm into the direction of t, which makes this technique infeasible for less structured graphs, such as isogeny graphs where every vertex has the same degree [JDF11].

We apply our new multidimensional electrical network framework to show an exponential quantum-classical separation to find an s-t path in an unweighted regular graph. We construct a family of regular graphs that we name as welded tree circuit graph $\mathcal{G}_C$, where each graph contains multiple welded tree graphs of depth n as subgraphs (see Fig. 4.14), meaning that this graph has exponentially many vertices in n. Note that the weight assigned on each edge in Fig. 4.14 can be constructed by viewing it as an electrical network and can be tweaked to improve the complexity of the algorithm. The actual pathfinding problem is related to an unweighted graph. Our framework allows us to approximate the alternative electrical flow state $|\theta^{\mathsf{alt}}\rangle$ in polynomial time for these types of graph. We then show that the overlap between each edge of an explicit s-t path and the alternative electrical flow is at least an inverse polynomial. Therefore, by sampling from a polynomial number of copies of $|\theta^{\mathsf{alt}}\rangle$, we can obtain a polynomial-sized subgraph that contains an s-t path. A classical algorithm such as breadth first search or depth first search can then be used to traverse the subgraph and output this s-t path.

Unlike the alternative electrical flow generated in one-dimensional unweighted random hierarchical networks that matches the actual electrical flow in the electrical network, the alternative electrical flow in our pathfinding example is significantly different from any 'real' electrical flow. As indicated in Fig. 4.4, which is a subgraph of the actual constructed regular graph, the alternative electrical flow has to split evenly at blue vertices that employ alternative Fourier neighborhoods, which is a consequence of Alternative Kirchhoff's Law. In this case, the alternative electrical flow can be viewed as an s-t flow in the electrical network that satisfies Alternative Kirchhoff's Law, meaning its corresponding flow state is orthogonal to the alternative Fourier neighborhoods. There might be multiple such s-t flows, but this alternative electrical flow has the minimum energy among them all.

The idea of using alternative electrical flows to find an s-t path shares a similarity with the approach in [HL] and [JKP23], where they achieve a polynomial speedup to find an s-t path for different types of graphs. Under the adjacency matrix model, [JKP23] suggests a new way to generate s-t electrical flow states based on span programs. This

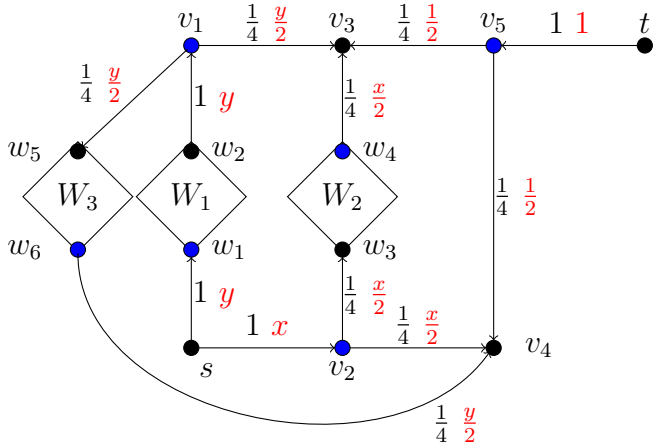

Figure 4.4. A graph and its corresponding edge directions where the blue vertices contain the Fourier alternative neighborhoods $\hat{\Psi}_\star(u)$ (see Definition 4.3.14). Each diamond, indexed by $i \in [3]$ represents a welded tree graph of depth n. For each $(u, v) \in \overrightarrow{E}$, the weights $\mathsf{w}_{u,v}$ are denoted in black, and the flow values $\theta^{\mathsf{alt}}_{u,v}$ in red for any valid unit s-t alternative flow parameterized by x and $y = 1 - x$.

approach can be used to sample an s-t path, which improves the query complexity of the previous quantum algorithm by [DHHM06] for certain types of graphs that have a unique s-t path, but the resulting complexities scale at least linearly with the number of vertices, which poses a problem when dealing with exponentially large graphs. In addition, for graphs with multiple s-t paths where all the s-t paths are short, [JKP23][Algorithm 4] also presents a different quantum algorithm, which does not use electrical flow, faster than the one in [DHHM06]. Under the adjacency list oracle model, [Wan17] uses the QLS algorithm to generate s-t electrical flow states and [HL] uses the electrical flow state showing a quadratic advantage to find an s-t path in graphs constructed from the welded tree graph over existing quantum and classical algorithms.

4.3.2 Multidimensional electrical networks

In this section, based on the multidimensional quantum walk framework [JZ23], we extend the electrical network to the multidimensional electrical network by generalizing Kirchhoff's Law and Ohm's Law as Alternative Kirchhoff's Law and Alternative Ohm's Law, respectively. One of the key techniques used in the multidimensional quantum walk framework is the introduction of alternative neighborhoods, where each vertex is associated with a subspace instead of a single vector (its star state), as was the case in Section 2.5.

As discussed and motivated previously, the multidimensional quantum walk framework modifies the quantum walk operator through the use of *alternative neighborhoods*.

Definition 4.3.1 (Alternative neighborhoods)**.** *For a network $G = (V, E, \mathsf{w})$ and for each vertex $u \in V$, a set of* alternative neighborhoods *is a collection of states $\Psi_\star(u)$ such that $|\psi_u\rangle \in \Psi_\star(u)$ and*

$$\Psi_\star = \{\Psi_\star(u) \subset \mathrm{span}_{\mathbb{R}}\{|u, v\rangle : v \in \Gamma(u)\} : u \in V\}$$

We view the states of $\Psi_\star(u)$ as different possibilities for $|\psi_u\rangle$, only one of which is "correct". We say we can generate $\Psi_\star$ in complexity $\mathsf{A}_\star$ *if there is a map $U_\star$ that can be implemented with complexity $\mathsf{A}_\star$ and for each $u \in V$, an orthonormal basis $\overline{\Psi}(u) = \{|\psi_{u,0}\rangle, \ldots, |\psi_{u,a_u-1}\rangle\}$ of size $a_u < |\Gamma(u)|$ for $\mathrm{span}\{\Psi_\star(u)\}$, such that for all $i \in \{0, \ldots, a_u - 1\}$, $U_\star |u, i\rangle = |\psi_{u,i}\rangle$.*

In Definition 4.3.1 we never exclude the possibility that the dimension a_u of the alternative neighborhood $\Psi_\star(u)$ is equal to one, in which case $\Psi_\star(u) = \{|\psi_{u,0}\rangle\} = \{|\psi_u\rangle\}$. If that is the case, we will say that u has no additional alternative neighborhoods. These alternative neighborhoods were introduced in [JZ23] to tackle the case where it might be computationally easier to generate $\Psi_\star(u)$ instead of $|\psi_u\rangle$. This can happen for example when dealing with an adjacency list oracle, where a single query to a vertex u does not allow us to distinguish its neighbors, which is needed when the star state $|\psi_u\rangle$ that we want to generate has different weights $\mathsf{w}_{u,v}$ for different neighbors v. In the well known welded tree problem [CCD+03] we are indeed dealing with such an adjacency list oracle and this hurdle is tackled in [JZ23] using alternative neighborhoods. This specific approach for the welded tree graph will be discussed and generalized in Section 4.3.6.

By modifying the quantum walk operator $U_{\mathcal{AB}}$ to reflect around the span of $\Psi_\star$ instead of the span of all star states $|\psi_u\rangle$, this reduces the cost of applying the walk operator $U_{\mathcal{AB}}$. As a result, one can reduce the precision needed in the phase estimation algorithm by reducing the weight of the graph, which directly reduces $\left\|p^{\mathsf{alt}}\right\|$, at the cost of increasing the effective resistance $\mathcal{R}_{s,t}$, without incurring an additional cost in calling $U_{\mathcal{AB}}$.

The addition of these alternative neighborhoods in $\Psi_\star$ modifies the quantum walk operator $U_{\mathcal{AB}^{\mathsf{alt}}}$, by increasing the star space $\mathcal{B}$:

$$\mathcal{B}^{\mathsf{alt}} = \mathrm{span}\{|\psi_{u,i}\rangle : u \in V \backslash \{s, t\}, i \in \{0, \ldots, a_u - 1\}\}.$$

Through this modification, the quantum walk operator $U_{\mathcal{AB}}$ with respect to $\Psi_\star$ is altered to

$$U_{\mathcal{AB}^{\mathsf{alt}}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}^{\mathsf{alt}}} - I), \tag{4.2}$$

where $\Pi_{\mathcal{A}}$ and $\Pi_{\mathcal{B}^{\mathsf{alt}}}$ are orthogonal projectors onto $\mathcal{A}$ and $\mathcal{B}^{\mathsf{alt}}$ respectively, meaning

$$2\Pi_{\mathcal{A}} - I = -\mathsf{SWAP}, \qquad 2\Pi_{\mathcal{B}^{\mathsf{alt}}} - I = 2\sum_{u\in V\setminus\{s,t\}}\sum_{i=0}^{a_u-1} |\psi_{u,i}\rangle\langle\psi_{u,i}| - I.$$

We would like to be able to apply Lemma 2.5.1 to this more general walk operator as well, meaning we want to find an alternative unit s-t flow θ^{alt}, an (unnormalized) state $|p^{\mathsf{alt}}\rangle$ and (normalized) state $|\psi\rangle$ such that the following conditions are satisfied:

1. $U|\theta^{\mathsf{alt}}\rangle = |\theta^{\mathsf{alt}}\rangle$.
2. $(I - \Pi_{\mathcal{A}})|p^{\mathsf{alt}}\rangle + \sqrt{\frac{2}{\mathcal{E}(\theta^{\mathsf{alt}})}}|\psi\rangle = |\theta^{\mathsf{alt}}\rangle$.
3. $\Pi_{\mathcal{B}^{\mathsf{alt}}}|p^{\mathsf{alt}}\rangle = |p^{\mathsf{alt}}\rangle$.

For simplicity, we will assume in the rest of this work that s and t do not contain any additional alternative neighborhoods, as it greatly simplifies notation and intuition. In our applications in Section 4.3.6 and Section 4.3.7 these simplifying assumptions will also hold.

Recall the definition of a flow state from Eq. (2.6) for any flow θ. By construction, $|\theta\rangle$ lives in the symmetric subspace $\mathcal{A}^{\perp}$, since

$$\Pi_{\mathcal{A}}(|u,v\rangle + |v,u\rangle) = \frac{I - \mathsf{SWAP}}{2}(|u,v\rangle + |v,u\rangle) = 0.$$

Hence any s-t flow θ that we select will satisfy $\Pi_{\mathcal{A}}|\theta\rangle = 0$. For the flow state $|\theta\rangle$ to live in the $+1$-eigenspace of U, it rests us to find some θ such that $\Pi_{\mathcal{B}}|\theta\rangle = 0$. In Eq. (2.8), we used Kirchhoff's Law for this goal, which showed that for any s-t -flow θ and vertex $u \in V\setminus\{s,t\}$, we have $\langle\psi_u|\theta\rangle = 0$. However, in the multidimensional electrical network, it must be orthogonal to all states in $\mathcal{B}^{\mathsf{alt}}$ instead of $\mathcal{B}$. That is, the state $|\theta^{\mathsf{alt}}\rangle$ must be orthogonal to all of $\mathrm{span}(\Psi_\star(u))$ for every $u \in V\setminus\{s,t\}$. We therefore modify Kirchhoff's Law to be *Alternative Kirchhoff's Law.*

Definition 4.3.2 (Alternative Kirchhoff's Law)**.** *For any s-t alternative flow θ^{alt} with respect to a collection of alternative neighborhoods $\Psi_\star$ on an electrical network G =*

(V, E, w) *with* $s, t \in V$*, the corresponding flow state* $|\theta^{\mathsf{alt}}\rangle$ *is orthogonal to* $\operatorname{span}(\Psi_\star(u))$ *for every* $u \in V \backslash \{s, t\}$*, that is,* $\langle \psi_{u,i} | \theta \rangle = 0$ *for each* $i \in \{0, 1, \cdots, a_u - 1\}$.

We refer to any unit s-t flow satisfying Alternative Kirchhoff's Law as an alternative unit s-t flow. Similarly as in Definition 2.5.2, we define the s-t alternative electrical flow with respect to $\Psi_\star$ as the alternative unit s-t flow achieving minimal energy:

Definition 4.3.3 (Alternative Electrical Flow)**.** *For a collection of alternative neighborhoods* $\Psi_\star$ *on an electrical network* $G = (V, E, \mathsf{w})$ *with* $s, t \in V$*, the* s-t alternative electrical flow *is the alternative unit* s-t *flow with minimal energy* $\mathcal{E}(\theta^{\mathsf{alt}})$*. We call this minimal energy the* alternative effective resistance $\mathcal{R}^{\mathsf{alt}}_{s,t}$.

Right now the definition might seem ill-defined, as at first glance there could very well be multiple alternative unit s-t flows that achieve the minimal energy $\mathcal{R}^{\mathsf{alt}}_{s,t}$, but we prove in Theorem 4.3.8 that the s-t alternative electrical flow is indeed unique (as long as any alternative unit s-t flow exists at all). It might be that the s-t electrical flow also satisfies Alternative Kirchhoff's Law, meaning that it coincides with the s-t alternative electrical flow. We show an example of this in Section 4.3.6 and this allows us to apply Lemma 2.5.1 directly using similar parameters as in Theorem 2.5.2. The other side of the spectrum is that there might not be any s-t flow at all that satisfies Alternative Kirchhoff's Law, in which case the s-t alternative electrical flow does not exist. We show an example of this shortly. The most likely scenario however is that we are right in the middle where the s-t electrical flow and s-t alternative electrical flow do not coincide, meaning we can not rely on Ohm's Law.

To apply Lemma 2.5.1, we still need to find an (unnormalized) state $|p^{\mathsf{alt}}\rangle$ and (normalized) state $|\psi\rangle$ such that

1. $(I - \Pi_{\mathcal{A}}) |p^{\mathsf{alt}}\rangle + \sqrt{\frac{2}{\mathcal{E}(\theta^{\mathsf{alt}})}} |\psi\rangle = |\theta^{\mathsf{alt}}\rangle$.
2. $\Pi_{\mathcal{B}^{\mathsf{alt}}} |p^{\mathsf{alt}}\rangle = |p^{\mathsf{alt}}\rangle$.

In the case that the s-t alternative electrical flow θ^{alt} does not overlap with the s-t electrical flow, we will not be able to find a potential vector p defined on the vertices V satisfying Ohm's Law. So instead we will be looking for a potential vector p^{alt} on the edges E, meaning it assigns a potential $p^{\mathsf{alt}}_{u,v}$ to each edge $(u, v) \in E$.

Definition 4.3.4 (Alternative Potential)**.** *An* alternative potential vector *(or alternative potential function) on a network* $G = (V, E, \mathsf{w})$ *is a real-valued function* $p^{\mathsf{alt}} : E \to \mathbb{R}$ *that assigns a potential* $p_{u,v}$ *to each ordered pair* $(u, v) \in E$.

Similarly to how the potential vector satisfied $p_s = \mathcal{R}_{s,t}$ and $p_t = 0$, we require the alternative potential vector p^{alt} to satisfy $p^{\mathsf{alt}}_{s,v} = \mathcal{R}^{\mathsf{alt}}_{s,t}$ and $p^{\mathsf{alt}}_{t,v} = 0$ for every $v \in \Gamma(s)$ (resp. $v \in \Gamma(t)$). We define its corresponding state in $\mathcal{H}$ as

$$|p^{\mathsf{alt}}\rangle = \sqrt{\frac{2}{\mathcal{R}(\theta^{\mathsf{alt}})}} \sum_{(u,v)\in\overrightarrow{E}: s\notin(u,v)} \sqrt{\mathsf{w}_{u,v}}(p^{\mathsf{alt}}_{u,v}\,|u,v\rangle - p^{\mathsf{alt}}_{v,u}\,|v,u\rangle). \tag{4.3}$$

Definition 4.3.5 (Alternative Ohm's Law)**.** *Let θ^{alt} be the s-t alternative electrical flow with respect to a collection of alternative neighborhoods $\Psi_\star$ on an electrical network $G = (V, E, \mathsf{w})$ with $s, t \in V$. Then there exists an alternative potential vector p^{alt} that assigns a potential $p^{\mathsf{alt}}_{u,v}$ on each edge $(u, v) \in E$ such that the associated state $|p^{\mathsf{alt}}\rangle$ (see Eq. (4.3)) satisfies $\Pi_{\mathcal{B}^{\mathsf{alt}}}|p^{\mathsf{alt}}\rangle = |p^{\mathsf{alt}}\rangle$ and the potential difference between (u, v) and (v, u) is equal to the amount of electrical flow $\theta^{\mathsf{alt}}_{u,v}$ along (u, v) multiplied with the resistance $1/\mathsf{w}_{u,v}$, that is, $p^{\mathsf{alt}}_{u,v} - p^{\mathsf{alt}}_{v,u} = \theta^{\mathsf{alt}}_{u,v}/\mathsf{w}_{u,v}$.*

We have not yet introduced the necessarily tools to show that there always exists a potential vector p^{alt} satisfying Alternative Ohm's Law, we will do this in Theorem 4.3.9. In the following examples and applications, we therefore show existence by explicitly constructing $|p^{\mathsf{alt}}\rangle$. If the potential vector p^{alt} satisfies Alternative Ohm's Law, then $|p^{\mathsf{alt}}\rangle$ is precisely the state we need to apply Lemma 2.5.1:

$$\begin{aligned}
|\theta^{\mathsf{alt}}\rangle &= \frac{1}{\sqrt{2\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{(u,v)\in\overrightarrow{E}} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}(|u,v\rangle + |v,u\rangle) \\
&= \frac{1}{\sqrt{2\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{(u,v)\in\overrightarrow{E}} \left(\sqrt{\mathsf{w}_{u,v}}(p^{\mathsf{alt}}_{u,v} - p^{\mathsf{alt}}_{v,u})\,|u,v\rangle + \sqrt{\mathsf{w}_{u,v}}(p^{\mathsf{alt}}_{u,v} - p^{\mathsf{alt}}_{v,u})\,|v,u\rangle\right) \\
&= \frac{1}{\sqrt{2\mathcal{R}^{\mathsf{alt}}_{s,t}}} \left(\sum_{(u,v)\in\overrightarrow{E}} \sqrt{\mathsf{w}_{u,v}}(p^{\mathsf{alt}}_{u,v}\,|u,v\rangle - p^{\mathsf{alt}}_{v,u}\,|v,u\rangle) + \mathsf{SWAP} \sum_{(u,v)\in\overrightarrow{E}} \sqrt{\mathsf{w}_{u,v}}(p^{\mathsf{alt}}_{u,v}\,|u,v\rangle - p^{\mathsf{alt}}_{v,u}\,|v,u\rangle)\right) \\
&= (I - \Pi_{\mathcal{A}})\sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{(u,v)\in\overrightarrow{E}} \sqrt{\mathsf{w}_{u,v}}(p^{\mathsf{alt}}_{u,v}\,|u,v\rangle - p^{\mathsf{alt}}_{v,u}\,|v,u\rangle) \\
&= (I - \Pi_{\mathcal{A}})\,|p^{\mathsf{alt}}\rangle + (I - \Pi_{\mathcal{A}})\sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{v\in\Gamma(s)} (-1)^{\Delta_{s,v}} p^{\mathsf{alt}}_{s,v}\sqrt{\mathsf{w}_{s,v}}\,|s,v\rangle \\
&= (I - \Pi_{\mathcal{A}})\,|p^{\mathsf{alt}}\rangle + \sqrt{\mathcal{R}^{\mathsf{alt}}_{s,t}\mathsf{w}_s}\,|\psi^+_s\rangle\,.
\end{aligned}$$

In the following examples and applications where we explicitly construct the state

$|p^{\mathsf{alt}}\rangle$, we need to verify that it satisfies $\Pi_{\mathcal{B}^{\mathsf{alt}}} |p^{\mathsf{alt}}\rangle = |p^{\mathsf{alt}}\rangle$. To assist in this verification, we introduce the states $|p^{\mathsf{alt}}_{|u}\rangle = (|u\rangle\langle u| \otimes I)|p^{\mathsf{alt}}\rangle$ for $u \in V$. To verify whether $\Pi_{\mathcal{B}^{\mathsf{alt}}} |p^{\mathsf{alt}}\rangle = |p^{\mathsf{alt}}\rangle$, it will be sufficient to verify whether each $|p^{\mathsf{alt}}_{|u}\rangle$ lies in $\mathrm{span}\{\Psi_\star(u)\}$, since we can decompose $|p^{\mathsf{alt}}\rangle$ as

$$\begin{aligned}
|p^{\mathsf{alt}}\rangle &= \sqrt{\frac{2}{\mathcal{R}(\theta^{\mathsf{alt}})}} \sum_{(u,v)\in\overrightarrow{E}:s\notin(u,v)} \sqrt{\mathsf{w}_{u,v}}(p^{\mathsf{alt}}_{u,v}|u,v\rangle - p^{\mathsf{alt}}_{v,u}|v,u\rangle) \\
&= \sqrt{\frac{2}{\mathcal{R}(\theta^{\mathsf{alt}})}} \sum_{u\in V}\sum_{v\in\Gamma(u)} (-1)^{\Delta_{u,v}} p^{\mathsf{alt}}_{u,v}\sqrt{\mathsf{w}_{u,v}}|u,v\rangle \qquad (4.4)\\
&= \sqrt{\frac{2}{\mathcal{R}(\theta^{\mathsf{alt}})}} \sum_{u\in V} |p^{\mathsf{alt}}_{|u}\rangle\,.
\end{aligned}$$

In the special case where u has no additional alternative neighborhoods, for $|p^{\mathsf{alt}}_{|u}\rangle$ to lay in $\mathrm{span}\{\Psi_\star(u)\} = \mathrm{span}\{|\psi_u\rangle\}$, the edge potentials $p_{u,v}$ must be the same for each $v \in \Gamma(u)$.

4.3.3 Examples

Having rebuilt the connection between the alternative potential vector and s-t alternative electrical flow in the multidimensional quantum electrical network framework, we now provide some intuition for these new definitions by providing a few examples.

Consider the network $G = (V, E, \mathsf{w})$ with the vertex set $V = \{s, x, y, t\}$ and directed edge set $\overrightarrow{E} = \{(s,x),(x,y),(x,t),(y,t)\}$, where each edge $(u,v) \in \overrightarrow{E}$ has weight $\mathsf{w}_{u,v} = 1/4$, except for the edge (s,x), which has weight $\mathsf{w}_{s,x} = 1$. This is visualized in Fig. 4.5. These directions and weight assignments give rise to the following star states for each of our 4 vertices:

$$\begin{aligned}
|\psi_s\rangle &= |s,x\rangle\,, & |\psi_x\rangle &= \sqrt{\frac{2}{3}}\left(-|x,s\rangle + \frac{1}{2}|x,y\rangle + \frac{1}{2}|x,t\rangle\right),\\
|\psi_y\rangle &= \sqrt{2}\left(-\frac{1}{2}|y,x\rangle + \frac{1}{2}|y,t\rangle\right), & |\psi_t\rangle &= \sqrt{2}\left(-\frac{1}{2}|t,x\rangle - \frac{1}{2}|t,y\rangle\right).
\end{aligned}$$

In Fig. 4.5 we show the s-t electrical flow θ on G and the corresponding potential vector p. It is straightforward to verify that θ and p satisfy Ohm's Law, meaning $p_u - p_v = \frac{\theta_{u,v}}{\mathsf{w}_{u,v}}$.

We now consider the case where only the vertex $x \in V$ contains an additional alter-

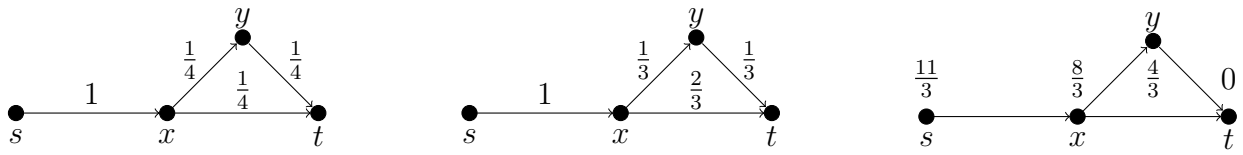

Figure 4.5. Graph G with its s-t electrical flow θ and corresponding potential p at each vertex.

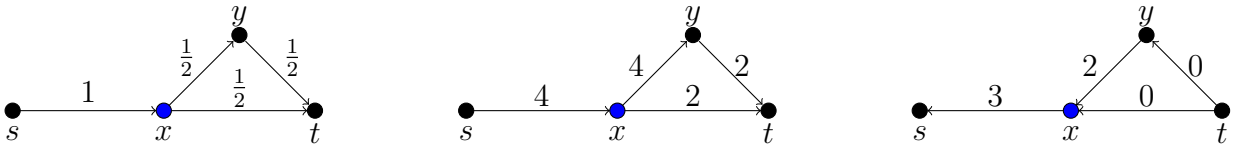

Figure 4.6. Graph G where the blue vertex x has an additional alternative neighborhood. The s-t alternative electrical flow θ^{alt} be with respect to this extra alternative neighborhood is displayed, as well as the corresponding potential vector p^{alt}.

native neighborhood: let $\Psi_\star(x) = \{|\psi_x\rangle, |\psi_x^{\mathsf{alt}}\rangle\}$ where

$$|\psi_x^{\mathsf{alt}}\rangle = \sqrt{\frac{2}{3}}(\frac{1}{2}|s,x\rangle - |x,y\rangle + \frac{1}{2}|x,t\rangle),$$

visualized in Fig. 4.6. Alternative Kirchhoff's Law states that the flow state $|\theta^{\mathsf{alt}}\rangle$ of any unit s-t flow θ^{alt} must additionally be orthogonal to $|\psi_x^{\mathrm{alt}}\rangle$. Together with being orthogonal to all the star states, meaning that the flow θ^{alt} is conserved at the vertices x and y, this leaves us with only a single option for θ^{alt}. This flow is visualized in Fig. 4.6 and the corresponding flow vector is given by

$$\begin{aligned}
|\theta^{\mathsf{alt}}\rangle &= \frac{1}{\sqrt{2\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{(u,v)\in\overrightarrow{E}} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}} (|u,v\rangle + |v,u\rangle) \\
&= \frac{1}{\sqrt{8}} \left(\frac{1}{1}(|s,x\rangle + |x,s\rangle) + \frac{1/2}{1/2}(|x,y\rangle + |y,x\rangle) + \frac{1/2}{1/2}(|x,t\rangle + |y,t\rangle) + \frac{1/2}{1/2}(|y,t\rangle + |t,y\rangle) \right) \\
&= \frac{1}{\sqrt{8}} (|s,x\rangle + |x,s\rangle + |x,y\rangle + |y,x\rangle + |x,t\rangle + |t,x\rangle + |y,t\rangle + |t,y\rangle)
\end{aligned}$$

Since this θ^{alt} is the only unit s-t flow satisfying Alternative Kirchhoff's Law, it is by default the s-t alternative electrical flow. For its alternative potential vector p^{alt}, we

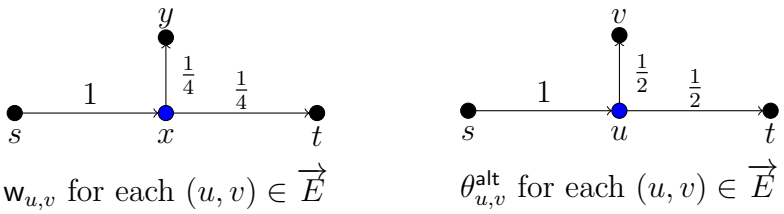

Figure 4.7. Graph G where the blue vertex x has an additional alternative neighborhood $|\psi_x^{\mathsf{alt}}\rangle$. There is no unit flow from s to t satisfying Alternative Kirchhoff's Law possible in this graph.

construct $|p^{\mathsf{alt}}\rangle$ from the bottom up by creating the states from Eq. (4.4):

$$|p^{\mathsf{alt}}_{|s}\rangle = 4\,|s,u\rangle\,, \qquad |p^{\mathsf{alt}}_{|x}\rangle = -3\,|x,s\rangle + 4\sqrt{\frac{1}{4}}\,|x,y\rangle + 2\sqrt{\frac{1}{4}}\,|x,t\rangle),$$
$$|p^{\mathsf{alt}}_{|y}\rangle = -2\sqrt{\frac{1}{4}}\,|y,x\rangle + 2\sqrt{\frac{1}{4}}\,|y,t\rangle\,, \qquad |p^{\mathsf{alt}}_{|t}\rangle = -0\sqrt{\frac{1}{4}}\,|t,x\rangle - 0\sqrt{\frac{1}{4}}\,|t,y\rangle\,.$$

Each such $|p^{\mathsf{alt}}_{|u}\rangle$ lies in $\mathrm{span}\{\Psi_\star(u)\}$ respectively. The alternative potential p^{alt} (see Fig. 4.6 for all the edge potentials) satisfies $p^{\mathsf{alt}}_{s,x} = \mathcal{R}^{\mathsf{alt}}_{s,t} = 4$ and $p^{\mathsf{alt}}_{t,x} = p^{\mathsf{alt}}_{t,y} = 0$, as well as Alternative Ohm's Law, meaning that each $(u,v) \in E$) satisfies $p^{\mathsf{alt}}_{u,v} - p^{\mathsf{alt}}_{v,u} = \theta^{\mathsf{alt}}_{u,v}/\mathsf{w}_{u,v}$. We have therefore found the alternative potential vector p^{alt} whose associated state $|p^{\mathsf{alt}}\rangle$ satisfies $\Pi_{\mathcal{B}}\,|p^{\mathsf{alt}}\rangle = |p^{\mathsf{alt}}\rangle$:

$$\begin{aligned}
|p^{\mathsf{alt}}\rangle &= \sqrt{\frac{2}{\mathcal{R}(\theta^{\mathsf{alt}})}} \sum_{(u,v)\in\overrightarrow{E}:s\notin(u,v)} \sqrt{\mathsf{w}_{u,v}}(p^{\mathsf{alt}}_{u,v}\,|u,v\rangle - p^{\mathsf{alt}}_{v,u}\,|v,u\rangle) \\
&= 4\,|s,x\rangle - 3\,|x,s\rangle + 2\,|x,y\rangle + |x,t\rangle - |y,x\rangle + |y,t\rangle \\
&= 4\,|\psi_s\rangle + \sqrt{\frac{3}{2}}\left(\frac{8}{3}\,|\psi_x\rangle + \frac{2}{3}\,|\psi_x^{\mathsf{alt}}\rangle\right) + \sqrt{2}\,|\psi_y\rangle + 0\,|\psi_t\rangle\,.
\end{aligned}$$

Depending on the alternative neighborhoods in $\Psi_\star$, the s-t alternative electrical flow might not exist, which is in contrast with regular electrical networks. As such a counterexample Fig. 4.7, we modify G once more, this time removing the edge (y,t) from $\overrightarrow{E}$. It is clear that any unit s-t flow θ^{alt} must satisfy $\theta^{\mathsf{alt}}_{s,x} = \theta^{\mathsf{alt}}_{x,t} = 1$ and $\theta^{\mathsf{alt}}_{x,y} = 0$, but in doing so, it will not satisfy Alternative Kirchhoff's Law, as the associated state $|\theta^{\mathsf{alt}}\rangle$ is not orthogonal to $|\psi_x^{\mathsf{alt}}\rangle$:

$$\langle\psi_x^{\mathsf{alt}}|\theta^{\mathsf{alt}}\rangle = \sqrt{\frac{2}{3}}.$$

4.3.4 The alternative incidence matrix, alternative Kirchhoff's Law and alternative Ohm's Law

In this section, inspired by the connection between the electrical network $G = (V, E, \mathsf{w})$ and the incidence matrix B of G, we rebuild the connection between the multidimensional electrical network and its alternative incidence matrix B_{alt}. We then use this connection to prove the uniqueness of the s-t alternative flow θ^{alt} and the existence of the alternative potential p^{alt} that satisfies Alternative Ohm's Law.

We start by restating the connection between on one hand the incidence matrix of a network G and on the other hand Kirchhoff's Law and Ohm's Law. We follow [Vis13, section 4] in doing so.

Definition 4.3.6 (The edge-vertex incidence matrix)**.** *Let $G = (V, E, \mathsf{w})$ be a network (See Definition 2.5.1). The incidence matrix $B \in \mathbb{C}^{\overrightarrow{E} \times V}$ of G, is the matrix whose rows are indexed by $(u, v) \in \overrightarrow{E}$, whose columns are indexed $u \in V$ and whose only non-zero entries are given by*

$$B_{(u,v),u} = \sqrt{\mathsf{w}_{u,v}}, \qquad\qquad B_{(u,v),v} = -\sqrt{\mathsf{w}_{u,v}}.$$

Let $W \in \mathbb{C}^{\overrightarrow{E} \times \overrightarrow{E}}$ be the diagonal matrix with entries $W_{(u,v),(u,v)} = 1/\sqrt{\mathsf{w}_{u,v}}$. By considering a flow θ on $G = (V, E, \mathsf{w})$ not only as a function on $\overrightarrow{E}$, but also as a vector in $\mathbb{C}^{\overrightarrow{E}}$, we can multiply it with the matrix W to obtain the weighted flow vector $W\theta \in \mathbb{C}^{\overrightarrow{E}}$ with entries $(W\theta)_{u,v} = \theta_{u,v}/\sqrt{\mathsf{w}_{u,v}}$ for the row indexed by $(u, v) \in \overrightarrow{E}$. The norm of $W\theta$ is therefore precisely given by $\sqrt{\mathcal{E}(f)}$. By the introduction of $W\theta$, we can rephrase Kirchhoff's Law from Definition 2.5.4 as a linear equation involving the incidence matrix B. Fix some ordering of the columns of B of the form $s, u_1, \ldots, u_2, t$ for some $u_1, u_2 \in V \backslash \{s, t\}$ and define the basis vectors $\mathsf{e}_i \in \mathbb{C}^n$ which have a 1 at the i-th location and zero elsewhere.

Proposition 4.3.4 (Kirchhoff's Law (incidence matrix))**.** *Let θ be any unit s-t flow on an electrical network $G = (V, E, \mathsf{w})$. Let B be the incidence matrix of G. Then θ*

satisfies

$$B^T W\theta = \begin{bmatrix} \sum_{v\in\Gamma(s)} \theta_{s,v} \\ \sum_{v\in\Gamma(u_1)} \theta_{u_1,v} \\ \vdots \\ \sum_{v\in\Gamma(u_2)} \theta_{u_2,v} \\ \sum_{v\in\Gamma(t)} \theta_{t,v} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix} = \mathsf{e}_s - \mathsf{e}_t. \tag{4.5}$$

Recall from Definition 2.5.2 that the s-t electrical flow is the flow that minimizes $\mathcal{E}(\theta)$ for all unit s-t flows θ. Since $\mathcal{E}(\theta) = \|W\theta\|^2$, this means that the s-t electrical flow corresponds to the 'smallest' (in norm) solution to Eq. (4.5), that is, the unique s-t flow θ such that its flow vector satisfies $W\theta \in \ker(B^T)^\perp$. We can therefore recover $W\theta$ by making use of the *Moore-Penrose inverse* (also known as the pseudoinverse) of B^T, denoted by B^{T+}. For any matrix A, the Moore-Penrose inverse A^+ (not to be confused with the conjugate transpose $A^\dagger$), is the unique matrix satisfying

$$AA^+A = A, \quad A^+AA^+ = A^+, \qquad (AA^+)^\dagger = AA^+, \qquad (A^+A)^\dagger = A^+A, \tag{4.6}$$

and it is well known that A^+ maps $\mathrm{ran}(A)$ to $\ker(A)^\perp$. Hence by left-multiplying both sides of Eq. (4.5) with B^{T+}, we recover the following important property of electrical networks:

Theorem 4.3.5 (Theorem 4.7 in [Vis13])**.** *Let θ be the s-t electrical flow on a network $G = (V, E, \mathsf{w})$. Let B be the incidence matrix of G. Then its flow vector $W\theta$ is given by*

$$W\theta = B^{T+}(\mathsf{e}_s - \mathsf{e}_t). \tag{4.7}$$

Just like we did with θ, we can also consider a potential vector p as a vector (hence the name) in $\mathbb{C}^V$ with entries p_u for the row indexed by $u \in V$. In doing so, we can rephrase Ohm's Law from Definition 2.5.5 as a linear equation involving the incidence matrix B. Fix some ordering of the rows of B of the form $(u_1, v_1), \ldots, (u_2, v_2) \in \overrightarrow{E}$.

Proposition 4.3.6 (Ohm's Law (incidence matrix))**.** *Let θ be the s-t electrical flow on an electrical network $G = (V, E, \mathsf{w})$. Let B be the incidence matrix of G. Then there*

exists a potential vector p such that

$$Bp = \begin{bmatrix} \sqrt{\mathsf{w}_{u_1,v_1}}\,(p_{u_1} - p_{v_1}) \\ \vdots \\ \sqrt{\mathsf{w}_{u_2,v_2}}\,(p_{u_2} - p_{v_2}) \end{bmatrix} = \begin{bmatrix} \frac{\theta_{u_1,v_1}}{\sqrt{\mathsf{w}_{u_1,v_1}}} \\ \vdots \\ \frac{\theta_{u_2,v_2}}{\sqrt{\mathsf{w}_{u_2,v_2}}} \end{bmatrix} = W\theta. \tag{4.8}$$

We may assume that the potential vector p satisfying Ohm's Law to satisfy $p_s = \mathcal{R}_{s,t}$ and $p_t = 0$, which is easier to see from the incidence matrix perspective.

Lemma 4.3.7. *Let θ be the s-t electrical flow on an electrical network $G = (V, E, \mathsf{w})$ with effective resistance $\mathcal{R}_{s,t}$. Then there exists a potential vector p satisfying Ohm's Law such that $p_s = \mathcal{R}_{s,t}$ and $p_t = 0$.*

Proof. From the incidence matrix B, we can obtain B^TB, which is known as *the weighted Laplacian* of G. It is well known in spectral graph theory (see e.g. Theorem 2.3 in [Vis13]), that B^TB has 0 as an eigenvalue with multiplicity 1. Since $\ker(B) = \ker(B^TB)$, not only does this mean that by setting $p_t = 0$, we still have a valid solution to Eq. (4.8), but this actually makes the remaining solution unique. By left-multiplying both sides of Eq. (4.7) with $(W\theta)^T$ we obtain together with Eq. (4.8) that

$$\mathcal{R}_{s,t} = \|W\theta\|^2 = (W\theta)^T B^{T+}(\mathsf{e}_s - \mathsf{e}_t) = p^T(\mathsf{e}_s - \mathsf{e}_t) = p_s - p_t = p_s. \tag{4.9}$$

□

With the Moore-Penrose inverse, we can in fact recover the potential from Lemma 4.3.7. To achieve this, we remove the last column of B and last row of p to obtain $\overline{B}$ and $\overline{p}$, effectively forcing $p_t = 0$:

$$p = \begin{bmatrix} \overline{p} \\ 0 \end{bmatrix} = \begin{bmatrix} \overline{B}^+ W\theta \\ 0 \end{bmatrix}. \tag{4.10}$$

What is it that makes the s-t electrical flow θ special, making it satisfy Ohm's Law. Why is Ohm's Law not necessarily true for our s-t alternative flow? Even though all flow states live in the symmetric subspace $\mathcal{A}^\perp$ by construction, we saw in Eq. (2.9) that the flow state $|\theta\rangle$ of the s-t electrical flow θ can be written as

$$|\theta\rangle = (I - \Pi_{\mathcal{A}})\sqrt{\frac{2}{\mathcal{R}_{s,t}}}\sum_{u\in V} p_u\sqrt{\mathsf{w}_u}\,|\psi_u\rangle\,,$$

meaning that $|\theta\rangle$ in fact lives in the *the symmetric star space* of $\mathcal{H}$, which is contained in $\mathcal{A}^\perp$:

$$H^{+\star} := \text{span}\{(I - \Pi_{\mathcal{A}})\,|\psi_u\rangle : u \in V\} = \text{span}\{|\psi_u^+\rangle : u \in V\}. \tag{4.11}$$

Out of all s-t flows, the s-t electrical flow is the unique unit flow such that $|\theta\rangle$ is the only corresponding flow state that is an element of $H^{+\star}$ (see e.g. [LP16]). We will not give a formal proof of this statement, but the intuition is that any other s-t flow has a higher energy, i.e. higher norm, which is due to containing a component that is orthogonal to all of $H^{+\star}$, namely a circulation. The column space of the incidence matrix B is in fact isomorphic to $H^{+\star}$, where the column of B indexed by $u \in V$ represents $\sqrt{\mathsf{w}_u}\,|\psi_u^+\rangle$ through the isometry

$$\mathcal{V} : \mathbb{C}^{|\overrightarrow{E}|} \mapsto \mathcal{A}^\perp, \text{ where } \mathcal{V}\,(u,v) = \sqrt{2}(I - \Pi_{\mathcal{A}})\,|u,v\rangle = \frac{1}{\sqrt{2}}\left(|u,v\rangle + |v,u\rangle\right). \tag{4.12}$$

Through the addition of alternative neighborhoods (see Definition 4.3.1), the space $H_G^{+\star}$ is effectively enlarged. Define

$$V^{\mathsf{alt}} := \{(u,i) \in V \times \mathbb{N} : i \in \{0, 1, \cdots, a_u - 1\}\}. \tag{4.13}$$

Instead of only considering the span of all $|\psi_u^+\rangle$ for $u \in V$, we now consider the span of all alternative neighborhoods projected onto the symmetric subspace, meaning $|\psi_{u,i}^+\rangle := \sqrt{2}(I - \Pi_{\mathcal{A}})\,|\psi_{u,i}\rangle$ for $(u,i) \in V^{\mathsf{alt}}$:

$$H^{+\mathsf{alt}} := \text{span}\{|\psi_{u,i}^+\rangle : u \in V, i \in \{0, 1, \cdots, a_u - 1\}\}. \tag{4.14}$$

By modifying the incidence matrix B to ensure that its column space still represents the newly modified $H_G^{+\mathsf{alt}}$, we obtain the alternative incidence matrix B_{alt}.

Definition 4.3.7 (Alternative incidence matrix)**.** *Let G be a network and let $\Psi_\star$ be a collection of alternative neighborhoods. Let $\{|\psi_{u,0}\rangle, \ldots, |\psi_{u,a_u-1}\rangle\}$ be an orthonormal basis for each $\Psi_\star(u) \in \Psi_\star$. The alternative incidence matrix $B_{\mathsf{alt}} \in \mathbb{C}^{\overrightarrow{E} \times V^{\mathsf{alt}}}$ of G is the matrix whose rows range over $(u,v) \in \overrightarrow{E}$, whose columns range over $(u,i) \in V^{\mathsf{alt}}$ and whose only non-zero entries are of the form*

$$B_{\mathsf{alt}(u,v),(u,i)} = \sqrt{\mathsf{w}_u}\langle u,v|\psi_{u,i}\rangle, \qquad B_{\mathsf{alt}(u,v),(v,j)} = \sqrt{\mathsf{w}_u}\langle u,v|\psi_{v,j}\rangle.$$

By Definition 4.3.1 we may assume that each $|\psi_{u,i}\rangle$ only has real coefficients and that $|\psi_{u,0}\rangle = |\psi_u\rangle$. By substituting B with B_{alt} in both Eq. (4.5) and Eq. (4.8), we can recover both Alternative Kirchhoff's Law and Alternative Ohm's Law, showing that these are indeed their natural definitions with respect to the perspective of the incidence matrix. Fix some ordering of the columns of B of the form $s, (u_1, i_1), \ldots, (u_2, i_2), t$ for some $u_1, u_2 \in V \backslash \{s, t\}$ such that $(u_1, i_1), (u_2, i_2) \in V^{\mathsf{alt}}$.

Definition 4.3.8 (Alternative Kirchhoff's Law (incidence matrix))**.** *Let θ^{alt} be any alternative unit s-t flow on an electrical network $G = (V, E, \mathsf{w})$ with respect to a collection of alternative neighborhoods $\Psi_\star$. Let B_{alt} be the alternative incidence matrix of G. Then θ^{alt} satisfies*

$$B_{\mathsf{alt}}^T W \theta^{\mathsf{alt}} = \begin{bmatrix} \sum_{v \in \Gamma(s)} \theta^{\mathsf{alt}}_{s,v} \\ \sum_{v \in \Gamma(u_1)} \frac{\theta^{\mathsf{alt}}_{u_1,v}}{\sqrt{\mathsf{w}_{u_1,v}}} \sqrt{\mathsf{w}_{u_1}} \langle u_1, v | \psi_{u_1,i_1} \rangle \\ \vdots \\ \sum_{v \in \Gamma(u_2)} \frac{\theta^{\mathsf{alt}}_{u_2,v}}{\sqrt{\mathsf{w}_{u_2,v}}} \sqrt{\mathsf{w}_{u_2}} \langle u_2, v | \psi_{u_2,i_2} \rangle \\ \sum_{v \in \Gamma(t)} \theta^{\mathsf{alt}}_{t,v} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix} = \mathsf{e}_s - \mathsf{e}_t. \tag{4.15}$$

Recall from Definition 4.3.3 that the s-t alternative electrical flow is the flow that minimizes $\mathcal{E}(\theta^{\mathsf{alt}})$ for all alternative unit s-t flows θ^{alt} (if any such flow exists). By applying the Moore-Penrose inverse of B_{alt}^T to Eq. (4.15), we prove that the s-t electrical flow is unique and thus well defined:

Theorem 4.3.8. *Let θ^{alt} be the s-t alternative electrical flow on a network $G = (V, E, \mathsf{w})$. Let B_{alt} be the alternative incidence matrix of G. Then $W\theta^{\mathsf{alt}}$ is given by*

$$W\theta^{\mathsf{alt}} = B_{\mathsf{alt}}^{T+}(\mathsf{e}_s - \mathsf{e}_t). \tag{4.16}$$

Recall the isometry $\mathcal{V}$ defined in Eq. (4.12). The column of B_{alt} indexed by $(u, i) \in V^{\mathsf{alt}}$ is equal to $\mathcal{V}^T\left(\sqrt{\mathsf{w}_u}\,|\psi^+_{u,i}\rangle\right)$, meaning that the column space of B_{alt} is equal to $\mathcal{V}^T\left(H^{+\mathsf{alt}}\right)$. Moreover, the column space of B_{alt} is equal to the column space of B_{alt}^{T+}, due to the properties of the Moore-Penrose inverse in Eq. (4.6). Combined with the fact that the state $|\theta^{\mathsf{alt}}\rangle$ is related to the vector $W\theta^{\mathsf{alt}}$ via the equality $\sqrt{\mathcal{R}^{\mathsf{alt}}_{s,t}}\,|\theta^{\mathsf{alt}}\rangle = \mathcal{V}\,(W\theta)$, we find that $|\theta^{\mathsf{alt}}\rangle$ is an element of $H^{+\mathsf{alt}}$. This means that there exist coefficients $p^{\mathsf{alt}}_{(u,i)}$

such that

$$|\theta^{\mathsf{alt}}\rangle = \frac{1}{\sqrt{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{u \in V} \sum_{i=0}^{a_u - 1} p^{\mathsf{alt}}_{(u,i)} \sqrt{\mathsf{w}_u}\, |\psi^+_{u,i}\rangle\,. \tag{4.17}$$

The notation $p^{\mathsf{alt}}_{(u,i)}$ seems to hint that these coefficients are related to the alternative potential vector p^{alt}. This is indeed the case: by defining the potential vector p^{alt} as

$$p^{\mathsf{alt}}_{u,v} := \frac{(-1)^{\Delta_{u,v}}}{\sqrt{\mathsf{w}_{u,v}}} \sum_{i=0}^{a_u - 1} p^{\mathsf{alt}}_{(u,i)} \sqrt{\mathsf{w}_u} \langle u, v | \psi_{u,i} \rangle, \tag{4.18}$$

we guarantee that the state $|p^{\mathsf{alt}}\rangle$ satisfies $\Pi_{\mathcal{B}} |p^{\mathsf{alt}}\rangle$:

$$\begin{aligned}
|p^{\mathsf{alt}}\rangle &= \sqrt{\frac{2}{\mathcal{R}(\theta^{\mathsf{alt}})}} \sum_{(u,v) \in \overrightarrow{E} : s \notin (u,v)} \sqrt{\mathsf{w}_{u,v}} (p^{\mathsf{alt}}_{u,v} |u, v\rangle - p^{\mathsf{alt}}_{v,u} |v, u\rangle) \\
&= \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{u \in V \setminus \{s\}} \sum_{v \in \Gamma(u)} \sum_{i=0}^{a_u - 1} p^{\mathsf{alt}}_{(u,i)} \sqrt{\mathsf{w}_u} \langle u, v | \psi_{u,i} \rangle\, |u, v\rangle \\
&= \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{u \in V \setminus \{s\}} \sum_{i=0}^{a_u - 1} p^{\mathsf{alt}}_{(u,i)} \sqrt{\mathsf{w}_u}\, |\psi_{u,i}\rangle\,.
\end{aligned}$$

Due to the coefficients $p^{\mathsf{alt}}_{(u,i)}$, we can therefore consider the alternative potential vector p^{alt} as a vector in $\mathbb{C}^{V^{\mathsf{alt}}}$ with entries $p^{\mathsf{alt}}_{(u,i)}$ for the row indexed by $(u, i) \in V^{\mathsf{alt}}$. By substituting B with B_{alt} in Eq. (4.8)Eq. (4.8) and combining this with Eq. (4.18), we recover Alternative Ohm's Law:

Definition 4.3.9 (Alternative Ohm's Law (incidence matrix))**.** *Let θ^{alt} be any alternative unit s-t flow on an electrical network $G = (V, E, \mathsf{w})$ with respect to a collection of alternative neighborhoods $\Psi_\star$. Let B_{alt} be the alternative incidence matrix of G. Then there exists an alternative potential vector p^{alt} such that $\Pi_{\mathcal{B}} |p^{\mathsf{alt}}\rangle = |p^{\mathsf{alt}}\rangle$ and*

$$B p^{\mathsf{alt}} = \begin{bmatrix} \sqrt{\mathsf{w}_{u_1,v_1}} \left(p^{\mathsf{alt}}_{u_1,v_1} - p^{\mathsf{alt}}_{v_1,u_1}\right) \\ \vdots \\ \sqrt{\mathsf{w}_{u_2,v_2}} \left(p^{\mathsf{alt}}_{u_2,v_2} - p^{\mathsf{alt}}_{v_2,u_2}\right) \end{bmatrix} = \begin{bmatrix} \frac{\theta_{u_1,v_1}}{\sqrt{\mathsf{w}_{u_1,v_1}}} \\ \vdots \\ \frac{\theta_{u_2,v_2}}{\sqrt{\mathsf{w}_{u_2,v_2}}} \end{bmatrix} = W \theta^{\mathsf{alt}}. \tag{4.19}$$

Just like with the potential vector p, we may assume that the alternative potential

vector p^{alt} satisfying Alternative Ohm's Law also satisfies $p_s^{\mathsf{alt}} = \mathcal{R}_{s,t}^{\mathsf{alt}}$ and $p_t^{\mathsf{alt}} = 0$

Theorem 4.3.9. *Let θ^{alt} be the s-t alternative electrical flow on an electrical network $G = (V, E, \mathsf{w})$ with respect to a collection of alternative neighborhoods $\Psi_\star$. Let B_{alt} be the alternative incidence matrix of G. Then there exists an alternative potential vector p^{alt} satisfying Alternative Ohm's Law such that $p_s^{\mathsf{alt}} = \mathcal{R}_{s,t}^{\mathsf{alt}}$ and $p_t^{\mathsf{alt}} = 0$.*

Proof. We apply the same trick as in Eq. (4.10), so we remove the last column of B_{alt} and last row of p^{alt} to obtain $\overline{B_{\mathsf{alt}}}$ and $\overline{p^{\mathsf{alt}}}$, forcing $p_t^{\mathsf{alt}} = 0$ for the solution satisfying Eq. (4.19):

$$p^{\mathsf{alt}} = \begin{bmatrix} \overline{p^{\mathsf{alt}}} \\ 0 \end{bmatrix} = \begin{bmatrix} \overline{B_{\mathsf{alt}}}^{+} W \theta^{\mathsf{alt}} \\ 0 \end{bmatrix}. \tag{4.20}$$

By left-multiplying both sides of Eq. (4.16) with $(W\theta^{\mathsf{alt}})^T$ we obtain together with Eq. (4.19) that

$$\mathcal{R}_{s,t}^{\mathsf{alt}} = \left\| W\theta^{\mathsf{alt}} \right\|^2 = (W\theta^{\mathsf{alt}})^T B^{T+}(\mathsf{e}_s - \mathsf{e}_t) = {p^{\mathsf{alt}}}^T(\mathsf{e}_s - \mathsf{e}_t) = p_s^{\mathsf{alt}} - p_t^{\mathsf{alt}} = p_s^{\mathsf{alt}}. \tag{4.21}$$

□

Due to Theorem 4.3.9, we may now apply Lemma 2.5.1 with the parameters $|\psi\rangle = |\psi_s^+\rangle$, $|\varphi\rangle = |\theta^{\mathsf{alt}}\rangle$, $|\phi\rangle = -\frac{1}{\sqrt{\mathcal{R}_{s,t}^{\mathsf{alt}} \mathsf{w}_s}} |p^{\mathsf{alt}}\rangle$ and $p = \frac{1}{\mathcal{R}_{s,t}^{\mathsf{alt}} \mathsf{w}_s}$, proving the following generalization of Theorem 2.5.2:

Theorem 4.3.10. *Let $\Psi_\star$ be a collection of alternative neighborhoods on a network $G = (V, E, \mathsf{w})$ and let $U_{\mathcal{AB}^{\mathsf{alt}}}$ be the quantum walk operator with respect to $\Psi_\star$ as defined in Eq. (4.2). Then by performing phase estimation on the initial state $|\psi_s^+\rangle$ with the operator $U_{\mathcal{AB}^{\mathsf{alt}}}$ and precision $O\left(\frac{\epsilon^2}{\sqrt{\mathcal{R}_{s,t}^{\mathsf{alt}} \mathsf{w}_s} \left\| p^{\mathsf{alt}} \right\|}\right)$, the phase estimation algorithm outputs "0" with probability $\Theta\left(\frac{1}{\mathcal{R}_{s,t}^{\mathsf{alt}} \mathsf{w}_s}\right)$, leaving a state $|\theta'\rangle$ satisfying*

$$\frac{1}{2} \left\| |\theta'\rangle\langle\theta'| - |\theta^{\mathsf{alt}}\rangle\langle\theta^{\mathsf{alt}}| \right\|_1 \leq \epsilon.$$

4.3.5 Examples

We will now show how these results apply to the examples Fig. 4.5 and Fig. 4.6 from section Section 4.3.3, which we have restated here in Fig. 4.8 and Fig. 4.9. Consider the graph G, consisting of the vertex set $V = \{s, x, y, t\}$ and directed edge set $\overrightarrow{E} =$

$\{(s,x),(x,y),(x,t),(y,t)\}$, where each edge $(u,v) \in \overrightarrow{E}$ has weight $\mathsf{w}_{u,v} = 1/4$, except for the edge (s,x), which has weight $\mathsf{w}_{s,x} = 1$. This graph is visualized in Fig. 4.8. These directions and weight assignments give rise to the following star states for each of our 4 vertices:

$$|\psi_s\rangle = |s,x\rangle\,, \qquad |\psi_x\rangle = \sqrt{\frac{2}{3}}\left(-|x,s\rangle + \frac{1}{2}|x,y\rangle + \frac{1}{2}|x,t\rangle\right),$$
$$|\psi_y\rangle = \sqrt{2}\left(-\frac{1}{2}|y,x\rangle + \frac{1}{2}|y,t\rangle\right), \qquad |\psi_t\rangle = \sqrt{2}\left(-\frac{1}{2}|t,x\rangle - \frac{1}{2}|t,y\rangle\right).$$

By ordering the directed edges as $(s,x),(x,y),(x,t),(y,t)$ and the vertices as s,x,y,t, we have that the incidence matrix B of G and the Moore-Penrose inverse B^{T+} of its transpose are equal to

$$B = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}, \qquad B^{T+} = \begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & -\frac{5}{6} & -\frac{1}{6} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{6} & -\frac{5}{6} \\ 0 & 0 & \frac{2}{3} & -\frac{2}{3} \end{bmatrix}. \tag{4.22}$$

The weighted diagonal matrix W is given by

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}. \tag{4.23}$$

We can recover the electrical flow θ in Fig. 4.8 using Theorem 4.3.5 to derive

$$W\theta = \begin{bmatrix} \frac{\theta_{s,x}}{\sqrt{\mathsf{w}_{s,x}}} \\ \frac{\theta_{x,y}}{\sqrt{\mathsf{w}_{x,y}}} \\ \frac{\theta_{x,t}}{\sqrt{\mathsf{w}_{x,t}}} \\ \frac{\theta_{y,t}}{\sqrt{\mathsf{w}_{y,t}}} \end{bmatrix} = B^{T+}(\mathsf{e}_s - \mathsf{e}_t) = \begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & -\frac{5}{6} & -\frac{1}{6} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{6} & -\frac{5}{6} \\ 0 & 0 & \frac{2}{3} & -\frac{2}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{2}{3} \\ \frac{4}{3} \\ \frac{2}{3} \end{bmatrix}.$$

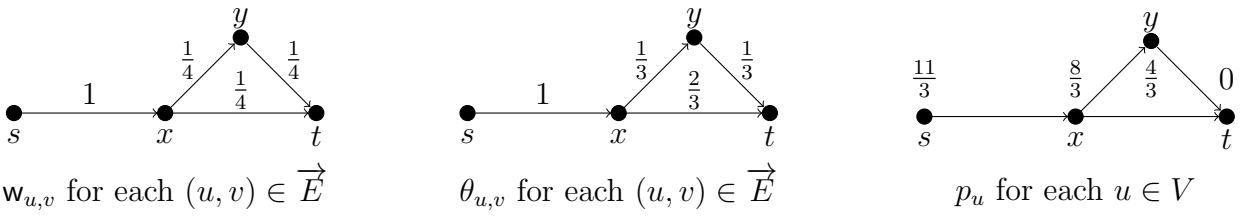

Figure 4.8. Graph G with its s-t electrical flow θ and corresponding potential p at each vertex.

This means that $\mathcal{R}_{s,t} = 1 + \frac{4}{9} + \frac{16}{9} + \frac{4}{9} = \frac{11}{3}$. By invoking Eq. (4.10), where the matrix $\overline{B}$ and its Moore-Penrose inverse $\overline{B}^+$ are equal to

$$\overline{B} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}, \qquad \overline{B}^+ = \begin{bmatrix} 1 & \frac{2}{3} & \frac{4}{3} & \frac{2}{3} \\ 0 & \frac{2}{3} & \frac{4}{3} & \frac{2}{3} \\ 0 & -\frac{2}{3} & \frac{2}{3} & \frac{4}{3} \end{bmatrix}, \tag{4.24}$$

we obtain that the potential at each vertex is given by

$$p = \begin{bmatrix} \overline{p} \\ 0 \end{bmatrix} = \begin{bmatrix} \overline{B}^+ W \theta \\ 0 \end{bmatrix} = \left(\begin{bmatrix} 1 & \frac{2}{3} & \frac{4}{3} & \frac{2}{3} \\ 0 & \frac{2}{3} & \frac{4}{3} & \frac{2}{3} \\ 0 & -\frac{2}{3} & \frac{2}{3} & \frac{4}{3} \end{bmatrix} \begin{bmatrix} 1 \\ \frac{2}{3} \\ \frac{4}{3} \\ \frac{2}{3} \end{bmatrix} \right) = \begin{bmatrix} \frac{11}{3} \\ \frac{8}{3} \\ \frac{4}{3} \\ 0 \end{bmatrix}, \tag{4.25}$$

meaning that the potential state $|p\rangle$ is equal to

$$|p\rangle = \sqrt{\frac{2}{\mathcal{R}_{s,t}}} \sum_{u \in V} p_u \sqrt{\mathsf{w}_u} \, |\psi_u\rangle = \frac{11}{3} \, |s,x\rangle - \frac{8}{3} \, |x,s\rangle + \frac{4}{3} \, |x,y\rangle + \frac{4}{3} \, |x,t\rangle - \frac{2}{3} \, |y,x\rangle + \frac{2}{3} \, |y,t\rangle \, . \tag{4.26}$$

We now consider the case where the vertex $x \in V$ contains an additional alternative

neighborhood: let $\Psi_\star(x) = \{|\psi_x\rangle, |\psi_x^{\mathsf{alt}}\rangle\}$ where

$$|\psi_x^{\mathsf{alt}}\rangle = \sqrt{\frac{2}{3}}(\frac{1}{2}|s,x\rangle - |x,y\rangle + \frac{1}{2}|x,t\rangle),$$

visualized in Fig. 4.9. By taking

$$\sqrt{\mathsf{w}_x}\,|\psi_{x,1}\rangle = \sqrt{\frac{3}{2}}\sqrt{\frac{1}{2}}(-|x,y\rangle + |x,t\rangle) = \frac{1}{2}\sqrt{3}(-|x,y\rangle + |x,t\rangle),$$

we find that $\{|\psi_{x,0}\rangle = |\psi_x\rangle, |\psi_{x,1}\rangle\}$ forms an orthonormal basis for $\Psi_\star(x)$. For this basis we find that the alternative incidence matrix B_{alt} of G and $\Psi_\star$ and the Moore-Penrose inverse B_{alt}^{T+} of its transpose are equal to

$$B_{\mathsf{alt}} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2}\sqrt{3} & -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2}\sqrt{3} & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}, \quad B_{\mathsf{alt}}^{T+} = \begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & 0 & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{3}\sqrt{3} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{3}\sqrt{3} & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{3}\sqrt{3} & 1 & -1 \end{bmatrix}. \tag{4.27}$$

We can recover the electrical flow θ^{alt} with respect to $\Psi_\star$ in Fig. 4.9 using Theorem 4.3.8 to derive

$$W\theta^{\mathsf{alt}} = \begin{bmatrix} \frac{\theta^{\mathsf{alt}}_{s,x}}{\sqrt{\mathsf{w}_{s,x}}} \\ \frac{\theta^{\mathsf{alt}}_{x,y}}{\sqrt{\mathsf{w}_{x,y}}} \\ \frac{\theta^{\mathsf{alt}}_{x,t}}{\sqrt{\mathsf{w}_{x,t}}} \\ \frac{\theta^{\mathsf{alt}}_{y,t}}{\sqrt{\mathsf{w}_{y,t}}} \end{bmatrix} = B_{\mathsf{alt}}^{T+}(\mathsf{e}_s - \mathsf{e}_t) = \begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & 0 & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{3}\sqrt{3} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{3}\sqrt{3} & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{3}\sqrt{3} & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

This means that $\mathcal{R}_{s,t}^{\mathsf{alt}} = 1+1+1+1 = 4$. By invoking Eq. (4.20), where the matrix

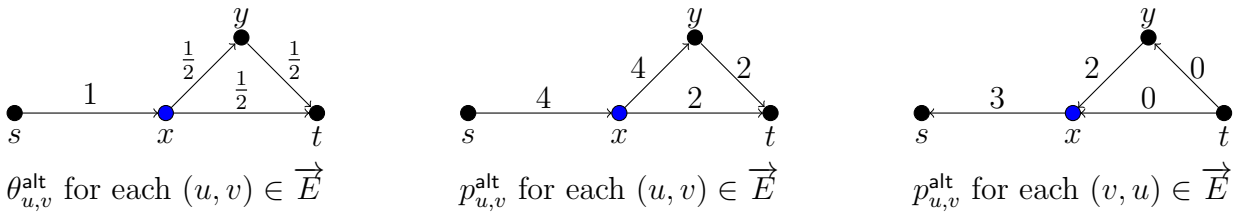

Figure 4.9. Graph G where the blue vertex x has an additional alternative neighborhood. The s-t alternative electrical flow θ^{alt} be with respect to this extra alternative neighborhood is displayed, as well as the corresponding potential vector p^{alt}.

$\overline{B_{\mathsf{alt}}}$ and its Moore-Penrose inverse $\overline{B}^{+}_{\mathsf{alt}}$ are equal to

$$\overline{B_{\mathsf{alt}}} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2}\sqrt{3} & -\frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2}\sqrt{3} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}, \qquad \overline{B_{\mathsf{alt}}}^{+} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & -\frac{1}{2}\sqrt{3} & \frac{1}{2}\sqrt{3} & -\frac{1}{2}\sqrt{3} \\ 0 & 0 & 0 & 2 \end{bmatrix}, \tag{4.28}$$

we obtain that the alternative potential at each alternative neighborhood is given by

$$p^{\mathsf{alt}} = \begin{bmatrix} \overline{p^{\mathsf{alt}}} \\ 0 \end{bmatrix} = \begin{bmatrix} \overline{B_{\mathsf{alt}}}^{+} W \theta^{\mathsf{alt}}_{s,t} \\ 0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & -\frac{1}{3}\sqrt{3} & \frac{1}{3}\sqrt{3} & -\frac{1}{3}\sqrt{3} \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ -\frac{1}{3}\sqrt{3} \\ 2 \\ 0 \end{bmatrix}, \tag{4.29}$$

meaning that the alternative potential state $|p^{\mathsf{alt}}\rangle$ is equal to

$$\begin{aligned} |p^{\mathsf{alt}}\rangle &= \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{u \in V, i \in \{0,\ldots,a_u - 1\}} p^{\mathsf{alt}}_{(u,i)} \sqrt{\mathsf{w}_u} \, |\psi_{(u,i)}\rangle \\ &= 4\,|s,x\rangle - 3\,|x,s\rangle + \frac{3}{2}\,|x,y\rangle + \frac{3}{2}\,|x,t\rangle + \frac{1}{2}\,|x,y\rangle - \frac{1}{2}\,|x,t\rangle - |y,x\rangle + |y,t\rangle \\ &= 4\,|s,x\rangle - 3\,|x,s\rangle + 2\,|x,y\rangle + |x,t\rangle - |y,x\rangle + |y,t\rangle\,. \end{aligned} \tag{4.30}$$

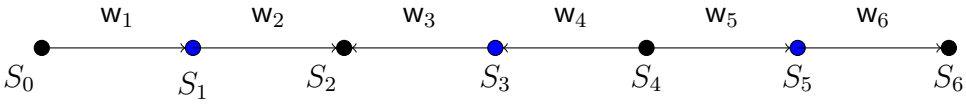

Figure 4.10. A line supergraph $\mathcal{G}$ with nodes $S_0, S_1, \ldots, S_6$. The black nodes are subsets of V_{even}, where the edge directions are reversed and where all adjacent edges have the same weight and direction.

4.3.6 Electrical flow sampling on one-dimensional random hierarchical graphs

Recently, [BLH23] have shown that there is an exponential separation between quantum and classical algorithms in finding a marked vertex in one-dimensional random hierarchical graphs, which is a generalization of the result of the welded tree problem [CCD+03]. In this section, we show that for one-dimensional random hierarchical graphs, we can efficiently generate a set of alternative neighborhoods $\Psi_\star$ such that the resulting s-t alternative electrical flow matches the s-t electrical flow, meaning it satisfies Ohm's Law. We show that this allows us to invoke Lemma 2.5.1 with similar parameters as in Theorem 2.5.2, allowing us to efficiently approximate the s-t electrical flow and sample from it to find a marked vertex, recovering some of the results from [BLH23]. Note that we could have invoked Theorem 4.3.10 directly, but instead aim for this approach to provide more intuition on how our results generalize Theorem 2.5.2.

Following [BLH23], we now define the one-dimensional random hierarchical graph model with nodes $S_0, S_1, \ldots, S_n$.

Definition 4.3.10 (Hierarchical graph on a line supergraph $\mathcal{G}$)**.** *A* hierarchical graph on a line supergraph $\mathcal{G} = (\mathcal{V} = \{0, \ldots, n\}, \mathcal{E})$ *of length n is defined by a set of nodes S_v for each $v \in \mathcal{V}$ and a set of edges $E_{u,v}$ for each $(u, v) \in \mathcal{E}$ such that $s_v = |S_v|$ and $e_{(u,v)} = |E_{u,v}|$. There are two special start and exit nodes $S_0 = \{s\}$ and $S_n = \{t\}$, meaning $s_0 = s_n = 1$. Define $V = \bigcup_{v \in \mathcal{V}} S_v$, $E = \bigcup_{(u,v) \in \mathcal{E}} E_{u,v}$ and $G = (V, E)$. For each $(u, v) \in E(G)$, the edge set $E_{u,v}$ denotes the set of edges between the nodes between S_u and S_v.*

Definition 4.3.11 (Balanced hierarchical graph)**.** *A hierarchical graph on a supergraph G is said to be* balanced *if for every $(u, v) \in E(G)$, the number of edges connecting a fixed node $\alpha \in S_u$ to nodes in S_v is the same for each α.*

Definition 4.3.12 (Edge-edge ratio)**.** *Consider a hierarchical graph on the line supergraph G which has nodes $S_0, S_1, \ldots, S_n$ where each node S_i contains $s_0, s_1, \ldots, s_n$ many*

vertices. Let e_k and $\mathcal{E}_k$ denote the number of edges and the set of edges between the nodes S_{k-1} and S_k respectively. Then the edge ratios *r_k for $k \in \{0, \ldots, n-1\}$ are defined as*

$$r_k = \frac{e_{k+1}}{e_k}.$$

Definition 4.3.13 (Edge-vertex ratio)**.** *A hierarchical graph on the line supergraph $G = (V, E)$ which has nodes $S_0, S_1, \ldots, S_n$ possesses* edge-vertex ratios *$\kappa_0, \kappa_1, \ldots, \kappa_n$ given by*

$$\kappa_j = \frac{e_j}{s_j}. \tag{4.31}$$

For a D-regular random balanced hierarchical graph on a line supergraph $G = (V, E)$, we have $e_i + e_{i+1} = e_i + r_i e_i = \kappa_i s_i + r_i \kappa_i s_i = D s_i$ and $\kappa_i(1 + r_i) = D$. Let $\ell = \Theta(n)$ be an integer such that $2^\ell \gg |V|$, where $|V|$ is the number of vertices in the one-dimensional random hierarchical graph G. This does impose the restriction that $|V|$ can be at most exponential in n. To each vertex in V, we assign a random name from the set $\{0,1\}^\ell$. To access the neighbors of a particular vertex, we are given quantum access to an adjacency list oracle O_G for the graph G. Given an ℓ-bit string $\sigma \in \{0,1\}^\ell$ corresponding to a vertex $u \in V$, the adjacency list oracle O_G provides the bit strings of the neighboring vertices in $\Gamma(u)$. If σ does not correspond to any vertex, which will most often be the case since $2^\ell \gg |V|$, the oracle instead returns $\perp$. This oracle structure effectively forces any algorithm to start in s and traverse the graph G from there, as it is infeasible to try and guess the name of any other vertex in V.

Problem 4.3.11 (One-dimensional random hierarchical graph problem)**.** *We are given an adjacency list oracle O_G to the one-dimensional random hierarchical graph G (D-regular) on the line supergraph of length n and the possibility to check whether any vertex u is equal to t. Given the ℓ-bit string associated with the starting vertex $s \in \{0,1\}^\ell$, output the ℓ-bit string corresponding to the other root t.*

Before we can use Lemma 2.5.1 to tackle this problem, we must turn G into an electrical network (see Definition 2.5.3), meaning we have to assign a weight and direction to each of its edges. We assign all edges in E_k the same weight for $k \in \{1, 2, \ldots, n\}$ and the weight w_k changes every two layers. Without loss of generality, we assume that n is an even number and set $\mathsf{w}_1 = 1$ and

$$\mathsf{w}_k = \prod_{i=1}^{\lfloor k/2 \rfloor} \left(\frac{1}{r_{2i-1}} \right)^2. \tag{4.32}$$

For each vertex $u \in S_i$ where $i \in \{0, 1, \ldots, 2n\}$, we find that $\mathsf{w}_u = \sum_{v \in \Gamma(u)} \mathsf{w}_{u,v} = \kappa_i \mathsf{w}_k + (D - \kappa_i)\mathsf{w}_{k+1}$. We define the set of directed edges as follows:

$$\overrightarrow{E} = \bigcup_{k \bmod 4 \in \{0,1\}} \{(u, v) : u \in S_{k-1}, v \in S_k\} \cup \bigcup_{k \bmod 4 \in \{2,3\}} \{(u, v) : v \in S_{k-1}, u \in S_k\}. \tag{4.33}$$

See Fig. 4.10 for an example of a line supergraph where this edge orientation and weight assignments are visualized. By viewing G as an electrical network, it is straightforward to directly compute the effective resistance $\mathcal{R}_{s,t}$ via the resistance laws for electrical circuits in series and parallel [Sie86]. As a result we find for the weight assignment from Eq. (4.32) that

$$\mathcal{R}_{s,t} = \frac{1}{D} + \sum_{k=2}^{n} \frac{1}{e_k \mathsf{w}_k}. \tag{4.34}$$

Since one-dimensional random hierarchical graphs generalize the welded tree graph, it should come as no surprise that we will use a collection of alternative neighborhoods that generalize the one used in [JZ23] to traverse the welded tree graph:

Definition 4.3.14 (Alternative Fourier neighborhood)**.** *Let G be a network. For any vertex $u \in V(G)$ with neighbors $\Gamma(u) = \{v_0, v_1, \ldots v_{D-1}\}$. Let $\omega_D = \exp(2\pi i/D)$ be the D-th root of unity. Then for each $j \in \{0, 1, 2, \ldots, D-1\}$, the j'th Fourier basis state is given by:*

$$|\hat{\psi}_u^j\rangle := \frac{1}{\sqrt{D}} \sum_{i=0}^{D-1} \omega_D^{i \cdot j} \, |u, v_i\rangle$$

We define the alternative Fourier neighborhood of dimension D *of the vertex u as*

$$\hat{\Psi}_\star(u) = \{|\hat{\psi}_u^1\rangle, |\hat{\psi}_u^2\rangle, \ldots |\hat{\psi}_u^{D-1}\rangle\}.$$

Recall that we defined the weights in Eq. (4.32) and the edge directions in Eq. (4.33) in an alternating fashion. This induces a partition of V into $V_{\text{even}} = \bigcup_{v \in \mathcal{V}: v \text{ is even}} S_v$ and $V_{\text{odd}} = \bigcup_{v \in \mathcal{V}: v \text{ is odd}} S_v$. We can assume without loss of generality that we know for any $u \in V$ whether it belongs to V_{even} or V_{odd} by keeping track of the parity of the distance from s that is initially 0 and flips every time the algorithm takes a step. For a more detailed argument of why this assumption is without loss of generality, we refer the reader to the end of Section 4 in [JZ23]. Note that at each vertex in V_{even} the edge directions are reversed and all adjacent edges have the same weight (see Fig. 4.10). It is therefore straightforward to generate the star state $|\psi_u\rangle$ for each $u \in V_{\text{even}}$, since $|\psi_u\rangle \propto |\hat{\psi}^0(u)\rangle$. For these vertices, we therefore do not consider any additional alternative

neighborhoods, meaning $\Psi_\star(u) = \{|\psi_u\rangle\}$. For $u \in S_i \subseteq V_{\text{odd}}\backslash\{t\}$, we let the set of alternative neighborhoods for any $u \in V_{\text{odd}}$ be the alternative Fourier neighborhood (see Definition 4.3.14): $\Psi_\star(u) = \hat{\Psi}_\star(u)$.

Lemma 4.3.12. *The quantum walk operator $U_{\mathcal{A}\mathcal{B}^{\mathsf{alt}}}$ as defined in Eq. (4.2) can be implemented in $O(1)$ queries to O_G and $O(nD)$ elementary operations.*

Proof. The unitary $U_{\mathcal{A}\mathcal{B}^{\mathsf{alt}}}$ consists of the two reflections $2\Pi_{\mathcal{A}} - 1$ and $2\Pi_{\mathcal{B}^{\mathsf{alt}}} - 1$. Since the former is (up to a sign difference) equal to the SWAP operator on two registers, each containing bit strings of length $\ell = \Theta(n)$, it can be implemented in 0 queries and $O(n)$ elementary operations. The cost of implementing $2\Pi_{\mathcal{B}^{\mathsf{alt}}} - 1$ follows almost directly from the proof of Lemma 4.4 in [JZ23], which proves the $D = 3$ case. By considering general D in their proof, it still holds that we only need $O(1)$ queries to O_G to apply $2\Pi_{\mathcal{B}^{\mathsf{alt}}} - 1$. The number of elementary operations needed in their proof is in the general case dominated by the cost of the following operation (needed to generate the state $|\hat{\psi}_u^j\rangle$), which for each $j \in \{0, \ldots, D-1\}$ applies the map

$$|j\rangle \left(\sum_{i=0}^{D-1} \omega_D^{i \cdot j} |i, 0\rangle \right) |v_0, v_1, \ldots v_{D-1}\rangle \mapsto |j\rangle \left(\sum_{i=0}^{D-1} \omega_D^{i \cdot j} |i, v_i\rangle \right) |v_0, v_1, \ldots v_{D-1}\rangle .$$

By conditioning on the value i, we can copy over the i'th value in the $|v_0, v_1, \ldots v_{D-1}\rangle$ register, but this will require $O(nD)$ elementary operations. This far exceeds the complexity of implementing the Quantum Fourier Transform F_D, which requires $O(\log(D) \log\log(D))$ elementary operations [HH00]. □

We now show how to apply Lemma 2.5.1 with the quantum walk operator $U_{\mathcal{A}\mathcal{B}^{\mathsf{alt}}}$, where $\mathcal{B}^{\mathsf{alt}} = \text{span}\{\text{span}(\Psi_\star(u)) : u \in V\backslash\{s,t\}\}$. We choose the same parameters as in Theorem 2.5.2. This means that for $|\theta\rangle$ we choose the state corresponding to the s-t electrical flow θ on G, which sends one unit of flow from s to t by evenly distributing the one unit of flow available at each layer S_i to the next layer S_{i+1} for each layer $i \in \{0, \ldots, n-1\}$. By Eq. (2.6) we obtain that

$$|\theta\rangle = \frac{1}{\sqrt{2\mathcal{R}_{s,t}}} \sum_{(u,v)\in\overrightarrow{E}} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}} (|u,v\rangle + |v,u\rangle) = \frac{1}{\sqrt{2\mathcal{R}_{s,t}}} \sum_{k=0}^{n-1} \sum_{(u,v)\in\overrightarrow{E}_k} (-1)^{\Delta_{u,v}} \frac{1}{e_k \sqrt{\mathsf{w}_k}} (|u,v\rangle + |v,u\rangle), \tag{4.35}$$

and it is straightforward to verify that $|\theta\rangle$ is normalized using Eq. (4.34), confirming that θ is in indeed the s-t electrical flow. By Eq. (2.10) we know that for its corresponding

potential vector p and with potential state $|p\rangle \in \mathcal{B} \subseteq \mathcal{B}^{\mathsf{alt}}$ (see Eq. (2.7)) we have

$$|\psi_s^+\rangle = \frac{1}{\sqrt{\mathcal{R}_{s,t} D}} |\theta\rangle - (I - \Pi_{\mathcal{A}}) \frac{1}{\sqrt{\mathcal{R}_{s,t} D}} |p\rangle .$$

Hence we can apply Lemma 2.5.1 by choosing $|\psi\rangle = |\psi_s^+\rangle$, $|\phi\rangle = -\frac{1}{\sqrt{\mathcal{R}_{s,t} D s}} |p\rangle$ and $p = \frac{1}{\mathcal{R}_{s,t} D}$ for remaining parameters, if we manage to show that $\Pi_{\mathcal{B}^{\mathsf{alt}}} |\theta\rangle = 0$. We achieve this with the following claim.

Claim 4.3.13. *For any $u \in V$, define $|\theta_u\rangle = (|u\rangle\langle u| \otimes I) |\theta\rangle$. If $u \in V_{\text{even}}$, then $|\theta_u\rangle \propto |\hat{\psi}^0(u)\rangle$. If $u \in V_{\text{odd}}$, then $|\theta_u\rangle \propto \sum_{v \in \Gamma(u)} \theta_{u,v} |u, v\rangle$. As a consequence, for every $u \in V$ and $|\psi_\star\rangle \in \Psi_\star(u)$ the state $|\theta_u\rangle$ satisfies $\langle \psi_\star | \theta_u \rangle = 0$.*

Proof. By construction of $|\theta\rangle$ (see Eq. (4.35)), we see for any $u \in V$ that the state $|\theta_u\rangle$ is equal to

$$|\theta_u\rangle = \frac{1}{\sqrt{2\mathcal{R}_{s,t}}} \sum_{v \in \Gamma(u)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}} |u, v\rangle .$$

Let k such that $u \in S_k$ and let $v_1, v_2, \ldots, v_l \in \Gamma(u) \cap S_{k-1}$ be the neighbors of u that lay in the node S_{k-1} and similarly let $v_{l+1}, \ldots, v_D \in \Gamma(u) \cap S_{k+1}$ be the neighbors of u that lay in the node S_{k+1}, where $l = D/(1 + r_k)$, which is in fact an integer. This means that $\theta_{u,v_i} = (-1)^{\Delta_{u,v_i}}/e_k$ for $i \in [l]$ and $\theta_{u,v_i} = (-1)^{\Delta_{u,v_i}}/e_{k+1}$ for $i \in [D]\backslash[l]$. If $u \in V_{\text{even}}$, then the weights (see Eq. (4.32)) satisfy

$$\sqrt{\frac{\mathsf{w}_{k+1}}{\mathsf{w}_k}} = \frac{e_k}{e_{k+1}} = \frac{1}{r_k},$$

meaning

$$\frac{1}{e_k \sqrt{\mathsf{w}_k}} = \frac{1}{e_{k+1} \sqrt{\mathsf{w}_{k+1}}}.$$

Additionally, since $u \in V_{\text{even}}$, it holds that $(-1)^{\Delta_{u,v_i}} (-1)^{\Delta_{u,v_j}} = -1$ for any $i \in [\ell]$ and $j \in [D]\backslash[\ell]$, meaning $|\theta_u\rangle \propto |\hat{\psi}^0(u)\rangle$. Since for $u \in V_{\text{even}}$ we defined $\Psi_\star(u)$ to be the alternative Fourier neighborhood (see Definition 4.3.14) and the Fourier basis states form an orthonormal basis, it follows that $\langle \psi_\star | \theta_u \rangle = 0$.

Now if instead $u \in V_{\text{odd}}$, then we know that $\mathsf{w}_k = \mathsf{w}_{k+1}$ and $(-1)^{\Delta_{u,v_i}} (-1)^{\Delta_{u,v_j}} = 1$ for any $i \in [\ell]$ and $j \in [D]\backslash[\ell]$. So $|\theta_u\rangle \propto \sum_{v \in \Gamma(u)} \theta_{u,v} |u, v\rangle$. Since for $u \in V_{\text{odd}}$ we defined $\Psi_\star(u) = \{|\psi_u\rangle = |\hat{\psi}^0(u)\rangle\}$, it follows by the conservation of the flow θ that $\langle \psi_u | \theta_u \rangle = \sum_{v \in \Gamma(u)} \theta_{u,v} = 0$. □

Knowing that we can apply Lemma 2.5.1 for our multidimensional electrical network,

we now show how to use this information to solve Problem 4.3.11.

We next provide a quantum algorithm that approximates the s-t electrical flow state and samples from it to find the ending vertex $t \in V$ in a one-dimensional random hierarchical graph. As an example of such a one-dimensional random hierarchical graph, we then apply our algorithm to the welded tree graph.

Algorithm 3 Solving the one-dimensional random hierarchical graph problem

---

Input: One-dimensional random hierarchical graph $G = (V, E)$ with adjacency list oracle O_G, the ℓ-bit string corresponding to the starting vertex $s \in V$, a success probability parameter δ.

Output: The ℓ-bit string corresponding to the ending vertex $t \in V$.

1. Set $i = 1$, $T_1 = \Theta(\mathcal{R}_{s,t} D)$ and $T_2 = \Theta(\mathcal{R}_{s,t} D \mathsf{w}_n \log(1/\delta))$.

2. For $j = 0$ to T_1, run phase estimation on the multidimensional quantum walk operator $U_{\mathcal{AB}^{\mathsf{alt}}}$ and state $|\psi_s^+\rangle$ to precision $O(\frac{\epsilon^2}{\sqrt{\mathcal{R}_{s,t}\mathsf{w}_s}\||p\rangle\|})$, where $\epsilon = \frac{1}{2\mathcal{R}_{s,t} D \mathsf{w}_n}$, and measure the phase register. If the output is "0", return the resulting state $|\theta'\rangle$ and immediately continue to Step 3.

3. Measure $|\theta'\rangle$ to obtain an outcome $|u, v\rangle$, representing the edge $(u, v) \in E$. Check if u or v is equal to t and if this is the case, return the ℓ-bit string corresponding to t. Otherwise, if $i < T_2$, increment i by 1 and return to Step 2.

Theorem 4.3.14. *Let G be a D-regular one-dimensional random hierarchical graph on the line supergraph of length n with edge ratios $r_0, \ldots, r_{n-1}$. Let $\mathsf{w}_n = \prod_{k=1}^{\lfloor n/2 \rfloor} (\frac{1}{r_{2k-1}})^2$ and let each vertex in G be identified by an ℓ-bit string where $\ell = \Theta(n)$. Given access to an adjacency list oracle O_G to the graph G, there exists a quantum algorithm that solves Problem 4.3.11 with success probability $1 - O(\delta)$ with*

$$O\left(\| \, |\mathsf{p}\rangle \, \| \mathcal{R}_{s,t}^{4.5} D^{4.5} \mathsf{w}_n^3 \log(1/\delta)\right) \text{ queries} \qquad O\left(n \| \, |\mathsf{p}\rangle \, \| \mathcal{R}_{s,t}^{4.5} D^{5.5} \mathsf{w}_n^3 \log(1/\delta)\right) \text{ time}$$

Proof. The proof consists of a cost and success probability analysis of Algorithm 3. By Lemma 2.5.1, each run of the phase estimation algorithm in Step 2 succeeds with probability at least $\Theta\left(\frac{1}{\mathcal{R}_{s,t} D}\right)$. Hence, the probability that at least a single out of the $T_1 = \Theta(\mathcal{R}_{s,t} D)$ runs succeed is constant.

Suppose that we had a perfect copy of $|\theta\rangle$, then after measuring it we would obtain

an edge $(u, v) \in E$ containing the vertex t with probability

$$\frac{1}{\mathcal{R}_{s,t}} \sum_{u \in \Gamma(t)} \frac{\theta_{u,t}^2}{\mathsf{w}_{u,t}} = \frac{1}{\mathcal{R}_{s,t} D \mathsf{w}_n}.$$

Instead, we have access to a state $|\theta'\rangle$, which by Lemma 2.5.1 satisfies

$$\frac{1}{2} \left\| |\theta'\rangle \langle\theta'| - |\theta\rangle \langle\theta| \right\|_1 \leq \epsilon = \frac{1}{2\mathcal{R}_{s,t} D \mathsf{w}_n}.$$

Hence by measuring $|\theta'\rangle$, we obtain an edge $(u, v) \in E$ that contains the vertex t with probability at least $\Theta\left(\frac{1}{\mathcal{R}_{s,t} D \mathsf{w}_n}\right)$. The probability that a single out of the at most $T_2 = \Theta(\mathcal{R}_{s,t} D \mathsf{w}_n \log(1/\delta))$ repetitions succeeds in returning the vertex t is therefore at least

$$1 - \left(1 - O\left(\frac{1}{\mathcal{R}_{s,t} D \mathsf{w}_n}\right)\right)^{T_2} \geq 1 - O(\delta).$$

For the cost of Step 2, each iteration of the phase estimation requires

$$O\left(\frac{\| |p\rangle \| \, \mathcal{R}_{s,t} D}{\epsilon^2}\right) = O(\| |p\rangle \| \, \mathcal{R}_{s,t}^3 D^3 \mathsf{w}_n^2)$$

calls to $U_{\mathcal{AB}^{\text{alt}}}$. By Lemma 4.3.12, each such call has a cost of $O(1)$ queries and $O(nD)$ elementary operations. Since we can set up the initial state $|\psi_s\rangle$ in the same cost and we run at most $T_1 \cdot T_2$ iterations of phase estimation, we find that the total contribution of Step 2 to the cost is

$$O\left(\| |\mathsf{p}\rangle \| \mathcal{R}_{s,t}^{4.5} D^{4.5} \mathsf{w}_n^3 \log(1/\delta)\right) \text{ queries} \qquad O\left(n \| |\mathsf{p}\rangle \| \mathcal{R}_{s,t}^{4.5} D^{5.5} \mathsf{w}_n^3 \log(1/\delta)\right) \text{ time}$$

For the cost of Step 3, we must only verify whether u or v is equal to t, which can be done in zero queries and $O(\ell) = O(n)$ elementary operations. So the cost of Step 2 dominates the total cost of the algorithm. □

As an example to show the power of this electrical flow sampling approach, we show that Algorithm 3 can be used to solve the welded tree problem in polynomial time, thus achieving an exponential speedup compared to any classical algorithms, which was originally shown in [CCD+03].

A welded tree graph consists of two full binary trees of depth h and contains $2^{h+2} - 2$ vertices. See Fig. 4.11 for an example of such a graph. The leaves of both trees are connected via two disjoint perfect matchings. This makes it a one-dimensional random

hierarchical graph on the line supergraph of length $n = 2h+1$. For each $k \in \{0, \ldots, 2h+1\}$, every node S_k contains

$$s_k = \begin{cases} 2^k & \text{if } k \in \{0, \ldots, h\} \\ 2^{2h+1-k} & \text{if } k \in \{h+1, \ldots, 2h+1\}, \end{cases}$$

vertices, meaning that its edge ratios are equal to

$$r_k = \begin{cases} 2 & \text{if } k \in \{1, \ldots, h\} \\ \frac{1}{2} & \text{if } k \in \{h+1, \ldots, 2h+1\}. \end{cases}$$

Since $V = 2^{h+2} - 2$, we find that $\ell = 2h$ satisfies $2^\ell \gg |V|$, meaning each vertex is assigned a $2h$-bit string as an identifier.

Problem 4.3.15 (The welded tree problem)**.** *Given an adjacency list oracle O_G for the welded tree graph G of depth h and the $2h$-bit string associated to the starting vertex $s \in \{0,1\}^{2h}$, output the $2h$-bit string associated to the other root t.*

Before we apply Theorem 4.3.14 to the welded tree graph, we first obtain a little more insight into its weights w_k. Our weight assignment from Eq. (4.32) will in this example match the weight assignment from [JZ23] (see Equation 31 in their work):

$$\mathsf{w}_k = \begin{cases} 2^{-2\lceil k/2 \rceil} & \text{if } k \in \{1, \ldots, h+1\} \\ 2^{-2(h+1-\lceil k/2 \rceil)} & \text{if } k \in \{h+2, \ldots, 2h+1\}. \end{cases} \tag{4.36}$$

Theorem 4.3.16. *Given an adjacency list oracle O_G to the welded tree graph G, there exists a quantum algorithm that solves Problem 4.3.15 with success probability $1 - O(\delta)$ and cost*

$$O\left(n^{5.5} \log(1/\delta)\right) \text{ queries}, \qquad O\left(n^{6.5} \log(1/\delta)\right) \text{ time}$$

Proof. The theorem can be derived by bounding the quantities $\mathcal{R}_{s,t}, D, \mathsf{w}_n$ and $\||p\rangle\|$ in Theorem 4.3.14. From Eq. (4.36) we see that $\mathsf{w}_n = 1/2$. Additionally, the effective resistance from Eq. (4.34) can be computed to find that $\mathcal{R}_{s,t} = \Theta(n)$. Since $D = 3$ and

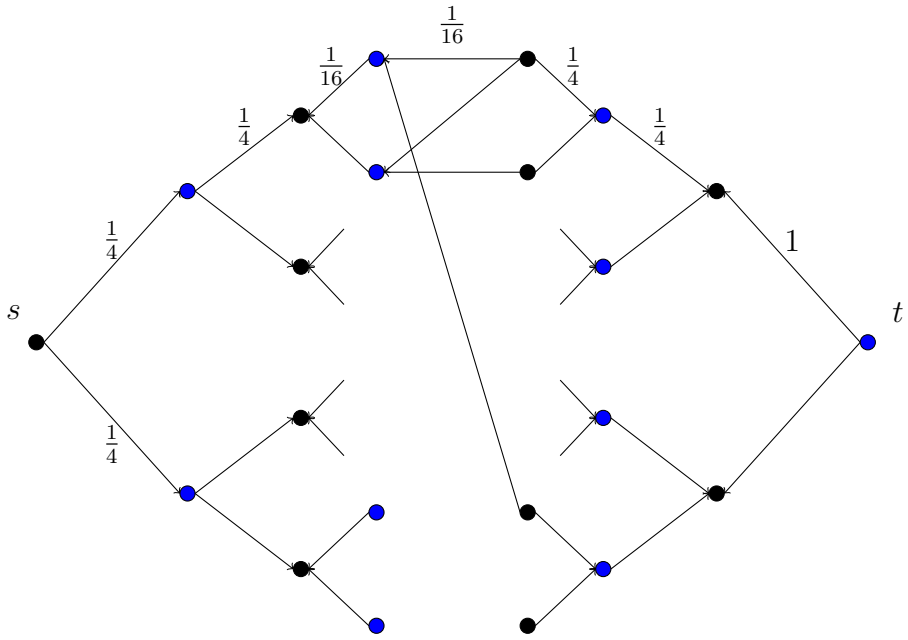

Figure 4.11. The welded tree graph with depth $h = 3$: the black vertices are the vertices in V_{even}, where the edge directions are reversed and where all adjacent edges have the same weight and direction.

$p_s = \mathcal{R}_{s,t}$ is the largest potential value, we only need to bound $\| |p\rangle \|$:

$$\| |p\rangle \|^2 = \frac{2}{\mathcal{R}_{s,t}} \sum_{k=0}^{n} \sum_{u \in S_k} p_u^2 \mathsf{w}_u \leq \mathcal{R}_{s,t} \sum_{k=0}^{n} s_k \mathsf{w}_u = O(n^2).$$

□

The result of Theorem 4.3.14 is worse than the state-of-the-art algorithm for the welded tree problem by [JZ23], which has cost $O(n)$ queries and $O(n^2)$ time. By approximating the electrical flow, we infer much more information than is actually needed to recover the bit string associated with t, but it exemplifies how sampling from the electrical flow can provide an exponential speedup.

4.3.7 Electrical flow sampling on the welded tree circuit $\mathcal{G}_C$

In this section, we show that the electrical flow in a multidimensional electrical network can also be used to show an exponential quantum-classical separation for the pathfinding problem relative to an oracle. We achieve this by constructing, and sampling from the s-t *alternative* electrical flow that we defined in Definition 4.3.3, which is the flow achieving minimal energy out of all unit s-t flows satisfying Alternative Kirchhoff's Law, and we

show that it also satisfies Alternative Ohm's Law through explicitly constructing the alternative potential p^{alt}. In all of this section, we assume that the parameter n is odd for readability, but everything can be slightly modified to also hold for even n.

Since the graph that we will try to find an s-t path for is quite large, we start by analyzing the s-t alternative flow and alternative potential for smaller graphs that will form the building blocks for the larger graph. We start with a network $G_1 = (V, E, \mathsf{w})$, whose vertex set is given by $V = \{s, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, t\}$. We have visualized G_1, with its directed edge set and weights in Fig. 4.12.

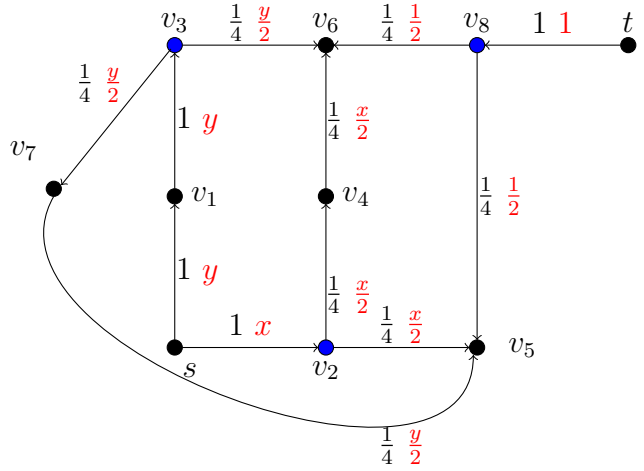

Figure 4.12. The graph G_1 with corresponding edge directions where the blue vertices have an additional alternative neighbor as defined in Eq. (4.37). For each $(u, v) \in \overrightarrow{E}$, the weights $\mathsf{w}_{u,v}$ are denoted in black, and the flow values $\theta^{\mathsf{alt}}_{u,v}$ in red for any valid unit s-t alternative flow parameterized by x and $y = 1 - x$.

These directions and weights give rise to the star state $|\psi_u\rangle$ for each $u \in V$, but we will also consider additional the following additional alternative neighborhoods for the vertices $v_2, v_3, v_8 \in V$:

$$
\begin{aligned}
|\psi^{\mathrm{alt}}_{v_2}\rangle &= \sqrt{\frac{2}{3}}\left(-|v_2, v_4\rangle + \frac{1}{2}|v_2, s\rangle + \frac{1}{2}|v_2, v_5\rangle\right), \\
|\psi^{\mathrm{alt}}_{v_3}\rangle &= \sqrt{\frac{2}{3}}\left(\frac{1}{2}|v_3, v_1\rangle - |v_3, v_6\rangle + \frac{1}{2}|v_3, v_7\rangle\right), \\
|\psi^{\mathrm{alt}}_{v_8}\rangle &= \sqrt{\frac{2}{3}}\left(\frac{1}{2}|v_8, t\rangle - |v_8, v_5\rangle + \frac{1}{2}|v_8, v_6\rangle\right).
\end{aligned}
\tag{4.37}
$$

Any s-t alternative unit flow θ^{alt} must be conserved at every vertex and satisfy

$\theta^{\mathsf{alt}}_{s,v_1} = x$, $\theta^{\mathsf{alt}}_{s,v_2} = y$ for some $x, y \in [0,1]$ such that $x + y = 1$. For θ^{alt} to also satisfy Alternative Kirchhoff's Law (see Definition 4.3.2), the flow coming into any vertex v_2, v_3, v_8 through the edge with the highest weight, must evenly be distributed along the other two neighbors. This is visualized in Fig. 4.12 and we end up with a single parameter x (because $y = 1 - x$) that parameterizes all possible s-t alternative unit flows θ^{alt} on G_1. The energy of each such θ^{alt} can be explicitly calculated to see that $\mathcal{E}(\theta^{\mathsf{alt}}) = 5y^2 + 4x^2 + 3$, and the energy is therefore minimized for $x = 5/9$, resulting in the alternative effective resistance to be $\mathcal{R}^{\mathsf{alt}}_{s,t} = 47/9$.

We now explicitly construct the alternative potential p^{alt} corresponding to this s-t alternative electrical flow, that satisfies $p^{\mathsf{alt}}_{s,v_1} = p^{\mathsf{alt}}_{s,v_2} = \mathcal{R}^{\mathsf{alt}}_{s,t} = 47/9$, $p^{\mathsf{alt}}_{t,v_8} = 0$ and Alternative Ohm's Law (see Definition 4.3.5). We do this by constructing the states $|p^{\mathsf{alt}}_{|u}\rangle \in \mathrm{span}\{\Psi_\star(u)\}$ from Eq. (4.4):

$$
\begin{aligned}
|p^{\mathsf{alt}}_{|s}\rangle &= \frac{47}{9}\,|s, v_1\rangle + \frac{47}{9}\,|s, v_2\rangle\,,\\
|p^{\mathsf{alt}}_{|v_1}\rangle &= -\frac{43}{9}\,|v_1, s\rangle + \frac{43}{9}\,|v_1, v_3\rangle\,,\\
|p^{\mathsf{alt}}_{|v_2}\rangle &= -\frac{42}{9}\,|v_2, s\rangle + \frac{38}{9}\sqrt{\frac{1}{4}}\,|v_2, v_4\rangle + \frac{46}{9}\sqrt{\frac{1}{4}}\,|v_2, v_5\rangle\,,\\
|p^{\mathsf{alt}}_{|v_3}\rangle &= -\frac{39}{9}\,|v_3, v_1\rangle + \frac{52}{9}\sqrt{\frac{1}{4}}\,|v_3, v_7\rangle + \frac{26}{9}\sqrt{\frac{1}{4}}\,|v_3, v_6\rangle\,,\\
|p^{\mathsf{alt}}_{|v_4}\rangle &= -2\sqrt{\frac{1}{4}}\,|v, u\rangle + 2\sqrt{\frac{1}{4}}\,|v, t\rangle\,,\\
|p^{\mathsf{alt}}_{|v_5}\rangle &= -4\sqrt{\frac{1}{4}}\,|v_5, v_8\rangle - 4\sqrt{\frac{1}{4}}\,|v_5, v_7\rangle - 4\sqrt{\frac{1}{4}}\,|v_5, v_2\rangle\,,\\
|p^{\mathsf{alt}}_{|v_6}\rangle &= -2\sqrt{\frac{1}{4}}\,|v_6, v_8\rangle - 2\sqrt{\frac{1}{4}}\,|v_6, v_4\rangle - 2\sqrt{\frac{1}{4}}\,|v_6, v_3\rangle\,,\\
|p^{\mathsf{alt}}_{|v_7}\rangle &= -\frac{44}{9}\sqrt{\frac{1}{4}}\,|v_7, v_3\rangle + \frac{44}{9}\sqrt{\frac{1}{4}}\,|v_7, v_5\rangle\,,\\
|p^{\mathsf{alt}}_{|v_8}\rangle &= -\,|v_8, t\rangle + 0\sqrt{\frac{1}{4}}\,|v_8, v_6\rangle) + 2\sqrt{\frac{1}{4}}\,|v_8, v_5\rangle\,,\\
|p^{\mathsf{alt}}_{|t}\rangle &= 0\,|t, v_8\rangle\,.
\end{aligned}
$$

It is straightforward to verify that these states indeed satisfy Alternative Ohm's Law as well as the equations $p^{\mathsf{alt}}_{s,v_1} = p^{\mathsf{alt}}_{s,v_2} = 47/9$, $p^{\mathsf{alt}}_{t,v_8} = 0$. It is also clear that $|p^{\mathsf{alt}}_{|u}\rangle \in \mathrm{span}\{\Psi_\star(u)\}$ for every u without additional alternative neighborhoods, i.e. $u \in \{s, v_1, v_4, v_5, v_6, v_7, t\}$, since all edge potentials are the same. For $u \in \{v_2, v_3, v_8\}$, we can

confirm that $|p^{\mathsf{alt}}_{|u}\rangle \in \text{span}\{\Psi_\star(u)\}$ by calculating that all the amplitudes of $|p^{\mathsf{alt}}_{|u}\rangle$ sum to 0.

The second example graph $G_2 = (V, E, \mathsf{w})$ (see Fig. 4.13) is build by combining the graph G_1 (see Fig. 4.12) with three welded tree graphs W_1, W_2, W_3 (see Fig. 4.11). The "starting" roots of these three welded tree graphs are w_1, w_4 and w_6 respectively.

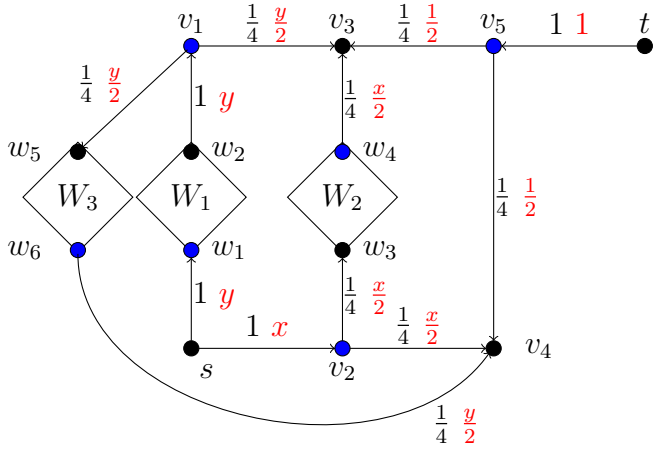

Figure 4.13. The graph G_2 with corresponding edge directions where the blue vertices are the vertices in V_{odd} and have the alternative neighborhoods $\Psi_\star(u) = \hat{\Psi}_\star(u)$ (see Definition 4.3.14). Each diamond, indexed by $i \in [3]$ represents a welded tree graph of depth n. For each $(u, v) \in \overrightarrow{E}$, the weights $\mathsf{w}_{u,v}$ are denoted in black, and the flow values $\theta^{\mathsf{alt}}_{u,v}$ in red for any valid unit s-t alternative flow parameterized by x and $y = 1 - x$. The black vertices are the vertices in V_{even}, where the edge directions are swapped and where adjacent edges have the same weight and direction.

The welded tree graph is an example of a one-dimensional random hierarchical graph with nodes $\{S_0, S_1, \cdots, S_n\}$. We additionally saw that for the weight assignments, edge directions, and alternative neighborhoods, we ended up with an s-t electrical flow that matched the s-t alternative electrical flow, as it also satisfied Alternative Kirchhoff's Law. From the perspective of electrical networks, we can therefore interpret each welded tree graph W_i as an edge of resistance $\mathcal{R}_i$, but we will formalize this intuition shortly. The weights and directions of W_1 in G_2 match those of Eq. (4.36) and Fig. 4.11, so $\mathcal{R}_i = \mathcal{R}$, where $\mathcal{R}$ is the effective resistance of a welded tree graph of depth n (see Eq. (4.34)). The weights of W_2 and W_3 have been multiplied by a factor of $1/4$, and their edge directions are reversed (because their respective roots are w_4 and w_6, so we have $\mathcal{R}_2 = \mathcal{R}_3 = 4\mathcal{R}$).

In G_2, the motivation for alternative neighborhoods, edge directions, and weight assignments in the network G_1 becomes clear. Exactly like for the one-dimensional random hierarchical graphs in Section 4.3.6, these assignments induce a partition of V

into V_{even} and V_{odd} (visualized by blue vertices in Fig. 4.13). For each vertex $u \in V_{\text{even}}$, all adjacent edges have the same weight and direction, allowing us to easily generate the star state $|\psi_u\rangle$. For each $u \in V_{\text{odd}} \backslash \{s,t\}$, we have $|\psi_u\rangle \in \Psi_\star(u) = \hat{\Psi}_\star(u)$. Like in Section 4.3.6, we can assume without loss of generality that we know for any $u \in V$ whether it belongs to V_{even} or V_{odd} by keeping track of the parity of the distance from s that is initially 0, and flips every time the algorithm takes a step.

Since the welded tree graph sends through all flow coming into one root to the other, any s-t alternative unit flow on G_2 is equivalent to a s-t alternative unit flow on G_1, with the addition that we also have flow running through each welded tree graph. By Fig. 4.12 and Fig. 4.13 we therefore see that the energy of a s-t alternative unit flow θ^{alt} can be decomposed by the energy in G_1 in addition to the energy on these welded tree graphs and is hence given by

$$\mathcal{E}(\theta^{\mathsf{alt}}) = 5y^2 + 4x^2 + 3 + \mathcal{R}_1 y^2 + \mathcal{R}_2 \left(\frac{x}{2}\right)^2 + \mathcal{R}_3 \left(\frac{y}{2}\right)^2 = (2 + 5\mathcal{R})y^2 + (4 + \mathcal{R})x^2 + 3.$$

This is minimized by taking $x = (2\mathcal{R}+5)/(3\mathcal{R}+9)$, meaning $y = 1-x = (\mathcal{R}+4)/(3\mathcal{R}+9)$. For readability, we actually keep x in the resulting alternative effective resistance, but simplify it slightly by making use of that for these values of x and y we have $(2+5\mathcal{R})y = (4 + \mathcal{R})x$:

$$\mathcal{R}^{\mathsf{alt}}_{s,t} = (2 + 5\mathcal{R})y^2 + (4 + \mathcal{R})x^2 + 3 = (4 + \mathcal{R})(x^2 + xy) + 3 = (4 + \mathcal{R})x + 3.$$

We now explicitly construct the alternative potential p^{alt} corresponding to this s-t alternative electrical flow, that satisfies $p^{\mathsf{alt}}_{s,w_1} = p^{\mathsf{alt}}_{s,v_2} = \mathcal{R}^{\mathsf{alt}}_{s,t} = (4 + \mathcal{R})x + 3$, $p^{\mathsf{alt}}_{t,v_5} = 0$ and Alternative Ohm's Law. We do this by constructing the states $|p^{\mathsf{alt}}_{|u}\rangle \in \text{span}\{\Psi_\star(u)\}$ from Eq. (4.4). We slightly abuse notation however and only show the edges visible in Fig. 4.13, meaning we will not explicitly write down the amplitudes and basis states for edges inside the welded tree graphs:

$$|p^{\mathsf{alt}}_{|s}\rangle = (3 + 5y + 2\mathcal{R}y)\,|s, w_1\rangle + (3 + 4x + \mathcal{R}x)\,|s, v_2\rangle\,, \quad |p^{\mathsf{alt}}_{|w_1}\rangle = -(3 + 4y + 2\mathcal{R}y)\,|w_1, s\rangle\,,$$

$$|p^{\mathsf{alt}}_{|w_2}\rangle = (3 + 4y + \mathcal{R}y)\,|w_2, v_1\rangle\,, \quad |p^{\mathsf{alt}}_{|w_3}\rangle = -(2 + 2x + 2\mathcal{R}x)\sqrt{\frac{1}{4}}\,|w_3, v_2\rangle\,,$$

$$|p^{\mathsf{alt}}_{|w_4}\rangle = -(2 + 2x)\sqrt{\frac{1}{4}}\,|w_4, v_3\rangle\,, \quad |p^{\mathsf{alt}}_{|w_5}\rangle = -(4 + 2y + 2\mathcal{R}y)\,|w_5, v_1\rangle\,,$$

$$|p^{\mathsf{alt}}_{|w_6}\rangle = (4 + 2y)\,|w_6, v_4\rangle\,, \quad |p^{\mathsf{alt}}_{|t}\rangle = 0\,|t, v_5\rangle\,,$$

$$
\begin{aligned}
|p^{\mathsf{alt}}_{|v_1}\rangle &= -(3+3y+\mathcal{R}y)\,|v_1,w_2\rangle + (4+4y+2\mathcal{R}y)\sqrt{\frac{1}{4}}(|v_1,w_5\rangle + (2+2y))\sqrt{\frac{1}{4}}\,|v_1,v_3\rangle\,,\\
|p^{\mathsf{alt}}_{|v_2}\rangle &= (2+4x+2\mathcal{R}x)\sqrt{\frac{1}{4}}\,|v_2,w_3\rangle + (4+2x)\sqrt{\frac{1}{4}}\,|v_2,v_4\rangle - (3+3x+\mathcal{R}x)\,|v_2,s\rangle\,,\\
|p^{\mathsf{alt}}_{|v_3}\rangle &= -2\sqrt{\frac{1}{4}}\,|v_3,v_5\rangle - 2\sqrt{\frac{1}{4}}\,|v_3,w_4\rangle - 2\sqrt{\frac{1}{4}}\,|v_3,v_1\rangle\,,\\
|p^{\mathsf{alt}}_{|v_4}\rangle &= -4\sqrt{\frac{1}{4}}\,|v_4,v_5\rangle - 4\sqrt{\frac{1}{4}}\,|v_4,v_2\rangle - 4\sqrt{\frac{1}{4}}\,|v_4,w_6\rangle\,,\\
|p^{\mathsf{alt}}_{|v_5}\rangle &= -\,|v_5,t\rangle + 0\sqrt{\frac{1}{4}}\,|v_5,v_3\rangle) + 2\sqrt{\frac{1}{4}}\,|v_5,v_4\rangle\,.
\end{aligned}
$$

It is straightforward to verify that these states indeed satisfy Alternative Ohm's Law for all edges outside the welded tree graphs as well as the equations $p^{\mathsf{alt}}_{s,w_1} = p^{\mathsf{alt}}_{s,v_2} = \mathcal{R}^{\mathsf{alt}}_{s,t}$, since $(2+5\mathcal{R})y = (4+\mathcal{R})x$ and that $p^{\mathsf{alt}}_{t,v_5} = 0$. It is also clear that $|p^{\mathsf{alt}}_{|u}\rangle \in \mathrm{span}\{\Psi_\star(u)\}$ for every $u \in \{s, v_3, v_4, t\}$, since all edge potentials. For $u \in \{v_1, v_2, v_5\}$, we can confirm that $|p^{\mathsf{alt}}_{|u}\rangle \in \mathrm{span}\{\Psi_\star(u)\}$ by calculating that all the amplitudes of $|p^{\mathsf{alt}}_{|u}\rangle$ sum to 0. For the edges in the welded tree graphs, we have seen in Section 4.3.6 that the s-t alternative electrical flow through each welded tree graph satisfies Ohm's Law. This means there exist potential values for all vertices (and hence edges), that are smaller than the potential at the root where the flows enters in at each welded tree graph that satisfy Alternative Ohm's Law. These are consistent with our potential p^{alt} since

$$
(p^{\mathsf{alt}}_{w_1,s} - p^{\mathsf{alt}}_{w_2,v_1})\frac{1}{y} = (p^{\mathsf{alt}}_{w_3,v_2} - p^{\mathsf{alt}}_{w_4,v_3})\frac{1}{x} = \left(p^{\mathsf{alt}}_{w_5,v_1} - p^{\mathsf{alt}}_{w_6,v_4}\right)\frac{1}{y} = \mathcal{R}.
$$

Recall from the proof of Theorem 4.3.16 that for a welded tree graph of depth n we have $\mathcal{R} = \Theta(n)$, meaning that $\mathcal{R}^{\mathsf{alt}}_{s,t} = \Theta(n)$. For the alternative potential, since for each edge potential we have $p^{\mathsf{alt}}_{u,v} = O(n)$, we find by Eq. (4.4) that

$$
||\,|p^{\mathsf{alt}}\rangle\,||^2 = \frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}} \sum_{(u,v)\in E} (p^{\mathsf{alt}}_{u,v})^2 \mathsf{w}_{u,v} = O(n) \sum_{(u,v)\in E} \mathsf{w}_{u,v} = O(n^2).
$$

We can now invoke Theorem 4.3.10 to approximate the state $|\theta^{\mathsf{alt}}\rangle$. Since the energy along the s-t path $(s,v_2),(v_2,v_4),(v_4,v_5),(v_5,t)$ contains a constant fraction of the energy $\mathcal{R}^{\mathsf{alt}}_{s,t}$, we could then sample from this state to recover a s-t path. However, since this path is of constant length, any classical algorithm can also recover this path by an exhaustive search of its neighbors in constant time.

Next, we construct a family welded tree circuit graph $\mathcal{G}_C$ by connecting n graphs iso-

morphic to G_2 (see Fig. 4.13) as a path as indicated in Fig. 4.14 and define a pathfinding problem for this type of graph.

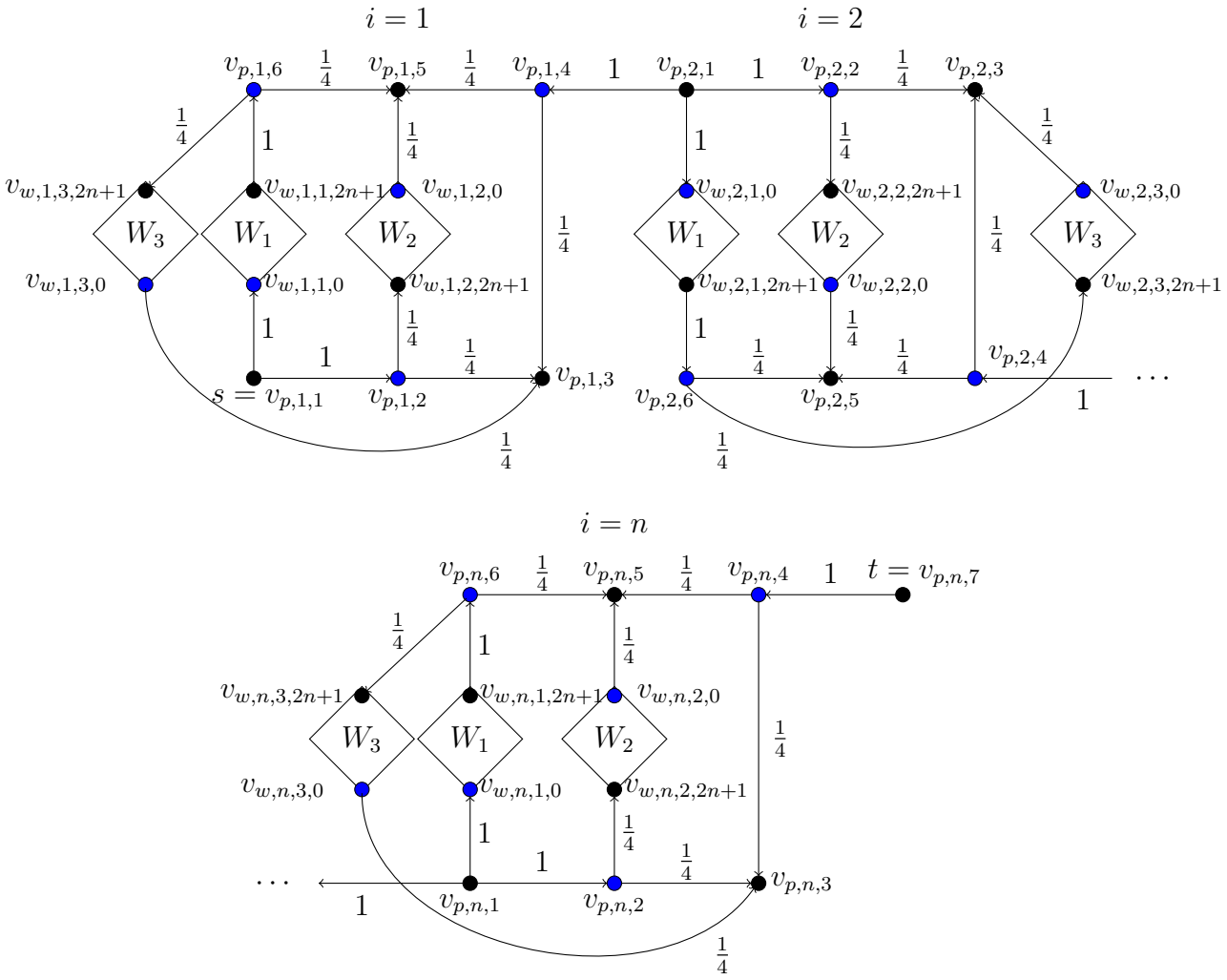

Figure 4.14. The welded tree circuit graph $\mathcal{G}_C$ showing all edge directions and edge weights. The blue vertices are the vertices in V_{odd} and have the alternative neighborhoods $\Psi_\star(u) = \hat{\Psi}_\star(u)$ (see Definition 4.3.14). The black vertices are the vertices in V_{even}, where the edge directions are swapped and where adjacent edges have the same weight and direction. Each diamond, indexed by $j \in [3]$ represents the j'-th welded tree graph in that layer. See Fig. 4.15 for a detailed overview of the welded tree graph's structure.

Each layer contains three welded tree graphs W_1, W_2, W_3 and the following 7 vertices

$$V_{p,i} := \{v_{p,i,j} : j \in [7]\}.$$

These layers are connected through the fact that $v_{p,i,7} = v_{p,(i+1),2}$ for every $i \in [n-1]$.

The welded tree graphs structure is shown in Fig. 4.15 for $j = 1$ (the edge directions are simply reversed for $j \in \{2, 3\}$) and the weight assignment and edge directions for these welded tree graphs, as well as for the remaining edges, are the same as for the graph G_2 in Fig. 4.13. The complete graph G is shown in Fig. 4.14. Due to this construction, each vertex has degree 3 except for the vertices $s = v_{p,1,1}$ and $t = v_{p,n,7}$. It therefore induces the same partition of V into V_{even} and V_{odd} as in G_2 (visualized by blue vertices in Fig. 4.14). For each vertex $u \in V_{\text{even}}$, all adjacent edges have the same weight and direction, allowing us to easily generate the star state $|\psi_u\rangle$. For each $u \in V_{\text{odd}} \backslash \{s, t\}$, we have $|\psi_u\rangle \in \Psi_\star(u) = \hat{\Psi}_\star(u)$.

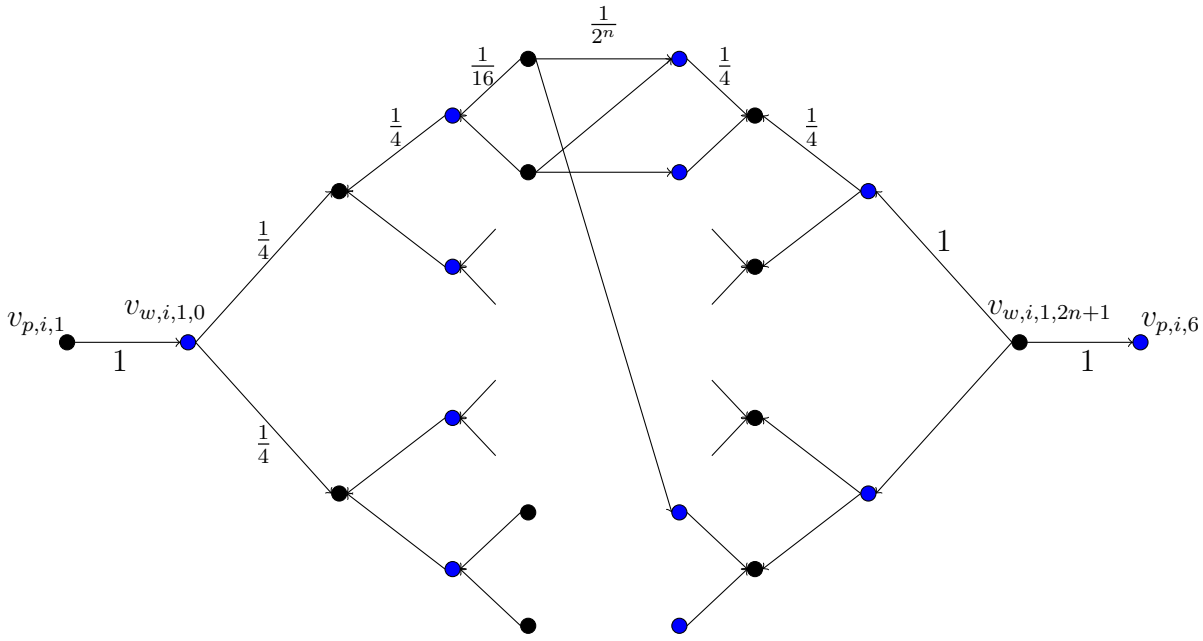

Figure 4.15. The 1st welded tree graph in the i'th layer. For $j \in \{2, 3\}$ the edge directions are simply reversed. The black vertices are the vertices in V_{even}, where the edge directions are reversed and where adjacent edges have the same weight and direction.

All these names $v_{a,b,c}$ to refer to vertices are simply for notation purposes to properly define the graph. Similar to the setting in Section 4.3.6, we assign a random name from the set $\{0, 1\}^{3n}$ to each vertex $u \in V$. To access the neighbors of a particular vertex, we have quantum access to an adjacency list oracle O_G for the graph $\mathcal{G}_C$. Given a $3n$-bit string $\sigma \in \{0, 1\}^{3n}$ corresponding to a vertex $u \in V$, the adjacency list oracle O_G provides the bit strings of the neighboring vertices in $\Gamma(u)$. If σ does not correspond to any vertex, which will most often be the case since $2^{3n} \gg |V|$, the oracle instead returns $\perp$.

As the graph $\mathcal{G}_C$ consists of n identical subgraphs isomorphic to G_2, the flow and

potential vector analysis almost directly follows from the analysis of G_2. Starting with the s-t alternative electrical flow θ^{alt}, we can obtain this flow by simply connecting the n s-t alternative electrical flows on each copy of G_2. This results in an alternative effective resistance $\mathcal{R}^{\mathsf{alt}}_{s,t} = \Theta(n^2)$. The alternative potential p^{alt} can also be obtained directly by combining all alternative potentials from each copy of G_2, where we add $((4+\mathcal{R})x+3)(n-i)$ to each edge potential obtained from the copy of G_2 in the i'th layer. This way we ensure that for every $i \in [n-1]$

$$|p^{\mathsf{alt}}_{v_{p,i+1,1}}\rangle = ((4+\mathcal{R})x+3)(n-i)\,|\psi_{v_{p,i+1,1}}\rangle\,,$$

meaning $||\,|p^{\mathsf{alt}}\rangle\,|| = O(n^2)$. We now consider the following problem on the graph G, for which we exhibit a quantum algorithm that can solve the given problem exponentially faster than any classical algorithm can.

Problem 4.3.17 (The pathfinding problem on a welded tree circuit graph $\mathcal{G}_S$)**.** *Given an adjacency list oracle O_G to the welded tree circuit graph $\mathcal{G}_C$ and the names of the starting vertex $s = 0^{3n}$, output the names of vertices of an s-t path.*

We then provide a quantum algorithm that can find the s-t shortest path in the welded tree circuit graph G and hence solves Problem 4.3.17 in polynomial time.

Algorithm 4 Quantum algorithm for solving Problem 4.3.17

Input: Oracle access of the welded tree circuit Graph $\mathcal{G}_S$, the starting vertex $s = 0^{3n}$, a success probability parameter $\delta > 0$.

Output: The labels of an s-t path on G.

1. Set $i = 1$, $S = \emptyset$, $T_1 = n^2$ and $T_2 = \Theta(n^2 \log(n/\delta))$.
2. For $j = 0$ to T_1, run phase estimation on the multidimensional quantum walk operator $U_{\mathcal{A}\mathcal{B}^{\mathsf{alt}}}$ and state $|\psi^+_s\rangle$ to precision $O(\epsilon^2/n^2)$, where $\epsilon = O(1/n^2)$, and measure the phase register. If the output is "0", return the resulting state $|\theta'\rangle$ and immediately continue to Step 3.
3. Measure $|\theta'\rangle$ to obtain an outcome $|u,v\rangle$, representing the edge $(u,v) \in E$, and add it to S. If $i < T_2$, increment i by 1 and return to Step 2.
4. Search through S using Breadth First Search for an s-t path and output the path if it is found.

Theorem 4.3.18. *Given an adjacency list oracle O_G to the welded tree circuit graph $\mathcal{G}_C$, there exists a quantum algorithm that solves Problem 4.3.17 with success probability*

$1 - O(\delta)$ *and cost*

$$O\left(n^{11}\log(n/\delta)\right) \text{ queries}, \qquad O\left(n^{12}\log(n/\delta)\right) \text{ time}.$$

Proof. The proof consists of a cost and success probability analysis of Algorithm 4, where we focus on the success probability that the algorithm outputs the path

$$\mathcal{P} = ((s, v_{p,1,2}), (v_{p,1,2}, v_{p,1,3}), (v_{p,1,3}, v_{p,1,4}), \ldots, (v_{p,n,4}, t)).$$

We apply Theorem 4.3.10, which states that each run of phase estimation in Step 2 succeeds with a probability of at least $\Theta\left(\frac{1}{\mathcal{R}^{\mathsf{alt}}_{s,t}}\right) = \Theta\left(\frac{1}{n^2}\right)$. Hence, the probability that at least a single out of the $T_1 = \Theta(n^2)$ runs succeed is constant.

Suppose that we had a perfect copy of $|\theta^{\mathsf{alt}}\rangle$, then after measuring it we would obtain an edge $(u, v) \in \mathcal{P}$ with a probability at least

$$\min_{(u,v)\in\mathcal{P}} \frac{1}{\mathcal{R}^{\mathsf{alt}}_{s,t}} \frac{(\theta^{\mathsf{alt}}_{u,v})^2}{\mathsf{w}_{u,v}} = \Omega\left(\frac{1}{n^2}\right).$$

Instead, we have access to a state $|\theta'\rangle$, which by Theorem 4.3.10 satisfies

$$\frac{1}{2}\left\| |\theta'\rangle\langle\theta'| - |\theta\rangle\langle\theta| \right\|_1 \leq \epsilon = O\left(\frac{1}{n^2}\right).$$

Hence by measuring $|\theta'\rangle$, we obtain an edge $(u, v) \in E$ that contains the vertex t with probability at least $\Omega\left(\frac{1}{n^2}\right)$. The probability that all edges in $\mathcal{P}$ are present in S after reaching Step 4 is due to the union bound therefore at least

$$1 - |\mathcal{P}|\left(1 - O\left(\frac{1}{n^2}\right)\right)^{T_2} \geq 1 - O(\delta).$$

For the cost of Step 2, each iteration of the phase estimation requires

$$O\left(\frac{\| |p^{\mathsf{alt}}\rangle \| \mathcal{R}^{\mathsf{alt}}_{s,t}}{\epsilon^2}\right) = O(n^8)$$

calls to $U_{\mathcal{AB}^{\mathsf{alt}}}$. By Lemma 4.3.12, each such call has a cost of $O(1)$ queries and $O(n)$ elementary operations. Since we can set up the initial state $|\psi_s\rangle$ in the same cost and we run at most $T_1 \cdot T_2$ iterations of phase estimation, we find that the total contribution

of Step 2 to the cost is

$$O\left(n^{11}\log(n/\delta)\right) \text{ queries,} \qquad O\left(n^{12}\log(n/\delta)\right) \text{ time.}$$

For the cost of Step 4, we must only do a Breadth First Search to search for any s-t path in the subgraph defined by the edges in S. Since identifying the vertex s and t can both be done using a single operation due to them having distinct degrees, the total cost of this step is $O(T_2) = O(n^2\log(n\delta))$ queries and other basic operations. So the cost of Step 2 dominates the total cost of the algorithm. □

4.4 Vertex Superposition

In this section, we introduce a family of multigraphs on which we can obtain an exponential quantum advantage for the pathfinding problem. We call this family of multigraphs regular sunflower graphs and denote an instance by $\mathcal{G}_S$. When the degree of the graph is greater than 7, we will show that the regular sunflower graph $\mathcal{G}_S$ is a mild expander graph with high probability in Section 4.4.5. Our quantum algorithm, described in detail in Algorithm 5, Section 4.4.6, first prepares a 0-eigenstate of the adjacency matrix A of the regular sunflower graph as a quantum state. Having access to this quantum state enables us to efficiently find the s-t path in the regular sunflower graph by measuring it to sample the vertices.

The existence of a 0-eigenvector of the adjacency matrix for certain special tree graphs has previously been used to demonstrate quantum advantages in formula evaluation [ACR$^+$10, FGG07]. More recently, the 0-eigenvector of the adjacency matrix for some random hierarchical graphs has been used to find a marked vertex [BLH23]. The result in this section is the first, to our knowledge, to use the information contained in the 0-eigenspace of an adjacency matrix to show an exponential quantum advantage for the pathfinding problem. This might provide some insight into the combination of spectral graph theory with the quantum eigenstate filtering algorithm for more applications.

In particular, we construct a family of d-regular ($d \geq 3$ is an odd integer) regular sunflower graphs $\mathcal{G}_S = (\mathcal{V}, \mathcal{E})$ (See Figure 4.3 for an example), with $|\mathcal{V}| = \exp(O(n))$, and prove the following:

1. With probability at least $1 - \exp(-\Omega(n))$, the graph $\mathcal{G}_S$ is a mild expander graph with spectral gap at least $1/\operatorname{poly}(n)$ when $d \geq 7$ (Theorem 4.4.7).

2. A quantum algorithm can find the s-t path for a specific pair of vertices s and t (given s but not t at the beginning) in time $\mathrm{poly}(n)$ (Theorem 4.4.17).

3. Any classical algorithm requires $\exp(\Omega(n))$ time to find t starting from s with a large probability (Theorem 4.5.7).

The rest of the section is organized as follows: In Section 4.4.1, we provide an overview of the quantum algorithm. In Section 4.4.2 we will introduce the notion of a (mild) expander graph, the definition of the regular sunflower graph $\mathcal{G}_S$, and the pathfinding problem in $\mathcal{G}_S$. In Section 4.4.3, we connect the adjacency matrix and the effective Hamiltonian via the invariant subspace $\mathcal{S}$. In Section 4.4.4, we prove that, with a high probability, $\mathcal{G}_S$ is a mild expander graph. In Section 4.4.5 we analyze the spectral properties of the effective Hamiltonian of $\mathcal{G}_S$. In Section 4.4.6 we provide an efficient quantum algorithm to find an s-t path in the graph.

4.4.1 Overview of the algorithm

There are two standard quantum algorithmic tools that can be used to obtain a 0-eigenstate of the adjacency matrix A. One is quantum phase estimation; the other is using quantum singular value transformation (QSVT) [GSLW18] together with a filtering polynomial [LT19a]. The input of the quantum phase estimation algorithm is $U = e^{-iAt}$ and $|s\rangle$. By post-selecting the estimated phase, one can approximately prepare a 0-eigenstate as a quantum state [GTC19, Proposition 3]. In this work, we will take the second approach and prepare the eigenstate using QSVT, which has the advantage of achieving a better dependence on the approximation error [Ton22, Theorem 6].

Using QSVT, we will first need to construct a *block encoding* of the adjacency matrix A, which is a unitary circuit encoding A as part of the unitary matrix [GSLW18]. A precise definition of the block encoding is given in Definition 2.1.1, and in Section 2.1 we show that the adjacency list oracle can be used to construct a block encoding with constant overhead.

With unitary U_A, that is, a block encoding of the adjacency matrix A, QSVT allows us to implement a minimax filtering polynomial that implements the projector Π_0 onto the 0-eigenspace of the adjacency matrix as indicated by Theorem 2.4.2 in Section 2.4. This means that, starting from a state $|s\rangle$, it yields a quantum circuit $\mathcal{V}_{\mathrm{circ}}$ such that

$$\mathcal{V}_{\mathrm{circ}} |0\rangle_\gamma |s\rangle_\beta = |0\rangle_\gamma f(A/\alpha) |s\rangle_\beta + |\perp\rangle_{\gamma\beta} \approx |0\rangle_\gamma \frac{\Pi_0 |s\rangle_\beta}{\|\Pi_0 |s\rangle_\beta\|} + |\perp\rangle_{\gamma\beta},$$

where $(\langle 0|_\gamma \otimes I_\beta)\,|\perp\rangle_{\gamma\beta} = 0$. Here $f(x)$ is a degree 2ℓ even polynomial satisfying $|f(x)| \leq 1$ for all $-1 \leq x \leq 1$, α is a normalization factor from block encoding that ensures $\|A/\alpha\| \leq 1$.

A very important fact about the adjacency matrix of the regular sunflower graph, similar to the welded tree graphs in [CCD+03] and the random hierarchical graphs in [BLH23], is that $|s\rangle$ lives in a low-dimensional invariant subspace of A. In our scenario, even though A acts on a Hilbert space of dimension $\exp(\Theta(n))$ (n is the number of trees we use to construct the regular sunflower graph), this invariant subspace is only $O(n^2)$-dimensional (assuming $m = \Theta(n)$ as mentioned above). This invariant subspace, which we call the *symmetric subspace* and denote by $\mathcal{S}$, is defined in Definition 4.4.9 (we prove that it is indeed invariant in Lemma 4.4.2). $\mathcal{S}$ is spanned by a set of $O(n^2)$ supervertex states defined in Eq. (4.43), which include $|s\rangle$ as one of them.

Because the effective Hamiltonian H (Definition 4.4.10) is the restriction of A to $\mathcal{S}$, one can readily prove that for any polynomial $p(x)$, $p(A)\,|\phi\rangle = p(H)\,|\phi\rangle$ for any $|\phi\rangle \in \mathcal{S}$ (Lemma 4.4.4). In particular, because $|s\rangle \in \mathcal{S}$, we have

$$f(A/\alpha)\,|s\rangle = f(H/\alpha)\,|s\rangle\,.$$

In other words, QSVT gives us the ability to implement matrix functions of the effective Hamiltonian H.

Recall that our goal is to prepare a 0-eigenstate of A. In fact, we will prepare a 0-eigenstate of H, which is guaranteed to also be a 0-eigenstate of A because H is the restriction of A to an invariant subspace. In other words,

$$f(A/\alpha)\,|s\rangle = f(H/\alpha)\,|s\rangle \approx \Pi_0\,|s\rangle\,. \tag{4.38}$$

Here $\Pi_0\,|s\rangle$ is an (unnormalized) 0-eigenstate of H, and is at the same time a 0-eigenstate of A.

In order to prepare the 0-eigenstate of the adjacency matrix A of the regular sunflower graph as the quantum state $\frac{\Pi_0|s\rangle}{\|\Pi_0|s\rangle\|}$ in polynomial time, it is sufficient to satisfy the following conditions:

1. The spectral gap Δ of H around the 0 eigenvalue should be at least $1/\operatorname{poly}(n)$. More precisely,
$$\Delta = \Omega(1/n^3).$$

2. The probability of getting a 0-eigenstate as a quantum state $\frac{\Pi_0|s\rangle}{\|\Pi_0|s\rangle\|}$ is at least

$1/\operatorname{poly}(n)$. More precisely,

$$\|(\langle 0|_\gamma \otimes I_\beta)\mathcal{V}_{\text{circ}} |0\rangle_\gamma |s\rangle_\beta \|^2 = \|\Pi_0 |s\rangle \|^2 = \Omega(1/n^2).$$

These two lower bounds are provided through the spectral properties of H stated in Corollary 4.4.16, and the success probability lower bound further uses Eq. (4.66).

The state we prepare through the above procedure approximates the quantum state $\frac{\Pi_0|s\rangle}{\|\Pi_0|s\rangle\|}$, which is a 0-eigenstate of H. However, the 0-eigenspace of H is two-dimensional (Corollary 4.4.16), and consequently being a 0-eigenstate does not uniquely determine $\Pi_0 |s\rangle$. We show that the normalized state $\frac{\Pi_0|s\rangle}{\|\Pi_0|s\rangle\|} = |\eta^{\text{odd}}\rangle$ in Eq. (4.65), where $|\eta^{\text{odd}}\rangle$ is a specific 0-eigenstate of H defined in Corollary 4.4.16.

The 0-eigenstate $|\eta^{\text{odd}}\rangle$ prepared in the above procedure is extremely useful in helping us find an s-t path. In fact, one s-t path consists of the roots of each tree $\mathcal{T}_i$ used to construct the regular sunflower graph $\mathcal{G}_S$. These roots are s_i (see Figure 4.3) for $i = 1, 2, \cdots, n$.. According to Corollary 4.4.16 (iii) the 0-eigenstate $|\eta^{\text{odd}}\rangle$ has at least $\Omega(1/n)$ overlap with each s_i for odd i (in Corollary 4.4.16 we write $s_i = S_{i,1}$ to be consistent with Definition 4.4.8). Therefore measuring $|\eta^{\text{odd}}\rangle$ in the computational basis yields a bit-string that represents each s_i for odd i with probability at least $\Omega(1/n^2)$. With $\tilde{O}(n^2)$ repetitions we can with large probability obtain all s_i for odd i.

Collecting all these sample vertices and their neighbors, which can be obtained through the adjacency list oracle, we then have a $\tilde{O}(n^2)$-sized subgraph that contains the s-t path, and therefore also the end vertex t, with large probability. t can be identified by the indicator function mentioned in Problem 4.4.1. The path can then be found using a Breadth First Search which runs on a classical computer in time $\operatorname{poly}(n)$. A formal statement with rigorous proof can be found in Theorem 4.4.17.

4.4.2 Graph definition and properties

In this section, we first introduce the notion of a mild expander graph. Then we give the definition of regular sunflower graph $\mathcal{G}_S$ and prove that $\mathcal{G}_S$ is a mild expander graph.

Following [Vad12, Chapter 4], we define the *neighborhood* and *edge boundary* of a set of vertices and (K, ϵ)-*vertex expander graph* as follows:

Definition 4.4.1 (Neighborhood in graph)**.** *For a graph $G = (V, E)$, we define the neighborhood of a vertex $u \in V$ to be $\Gamma(u) := \{v \in V : \{u, v\} \in E\}$. Similarly we define the neighborhood of a set $S \subset V$ to be $\Gamma(S) = \bigcup_{u \in S} \Gamma(u)$.*

The *edge boundary* of a set of vertices is defined in the following way:

Definition 4.4.2 (Edge boundary)**.** *For a graph $G = (V, E)$, we define the edge boundary of a subset $S \subset V$ to be $\partial(S) = \{\{u, v\} : u \in S, v \notin S\}$.*

With the above, we are ready to define what an expander graph is:

Definition 4.4.3 ((K, ϵ)-vertex expander graph)**.** *A d-regular, unweighted multigraph G is a* (K, ϵ)-vertex expander *if for every set $S \subset V$ such that $|S| \leq K \leq |V|/2$, we have*

$$|\Gamma(S) \backslash S| \geq \epsilon d |S|.$$

There are usually three equivalent definitions of expanders based on vertex expansion, edge expansion, and spectral expansion. A (K, ϵ)-vertex expander graph according to Definition 4.4.3 also satisfies

$$|\partial(S)| \geq |\Gamma(S) \backslash S| \geq \epsilon d |S|,$$

where the first inequality is because each element in $\Gamma(S) \backslash S$ must be connected to S by at least one edge. By Cheeger's inequalities, if $K = |V|/2$, this also implies spectral expansion, that is, the spectral gap of the graph Laplacian (for a definition see [Chu97, Chapter 1]) is lower bounded by at least $\epsilon^2 d/2$ [Vad12, Theorem 4.9].

Based on the definition of (K, ϵ)-vertex expander in Definition 4.4.3, we define *mild expander graph* as follows.

Definition 4.4.4 (Mild expander graph)**.** *A d-regular, unweighted multigraph G is a* (K, ϵ)-vertex mild expander graph *if for every set $S \subset V$ such that $|S| \leq K \leq |V|/2$ and $\epsilon d = 1/\mathrm{poly}\log(|V|)$, we have*

$$|\Gamma(S) \backslash S| \geq \epsilon d |S|.$$

We then provide the definition of the regular sunflower graph $\mathcal{G}_S$ as follows.

Definition 4.4.5 (The regular sunflower graph $\mathcal{G}_S = (\mathcal{V}, \mathcal{E})$)**.** *Let $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$ be rooted trees of height $m \geq 2$. For each $\mathcal{T}_i$, the root has degree $d-2$, and all other internal vertices have degree $d-1$. All leaves must be distance m from the root. We then link the roots $s_1, s_2, \ldots, s_n$ of trees $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$ with the edges $\{s_1, s_2\}, \{s_2, s_3\}, \ldots, \{s_{n-1}, s_n\}, \{s_n, s_1\}$. Next, we link the $(d-2)(d-1)^{m-2}$ leaves of $\mathcal{T}_i$ with the leaves of $\mathcal{T}_{i+1}$ for $1 \leq i \leq n-1$ and the leaves of $\mathcal{T}_n$ with the leaves of $\mathcal{T}_1$. For each pair of trees, select $(d-1)/2$ random*

perfect matchings between their leaves and add all their edges to the graph. The resulting multigraph is a d-regular sunflower graph $\mathcal{G}_S = (\mathcal{V}, \mathcal{E})$.

An illustration of the graph is given in Figure 4.3 for $d = 3, m = 4, n = 8$. When proving the classical query complexity lower bound in Section 4.5.3, we assume $m = \Theta(n)$ for simplicity.

All information of the multigraph is represented in its adjacency matrix, as defined below:

Definition 4.4.6. *Let* $G = (V, E)$ *be an undirected multigraph of degree* $d = O(1)$, $\mathsf{N} = |V|$. *Then its adjacency matrix* $A = (A_{v,v'})_{v,v' \in V}$ *is an* $\mathsf{N} \times \mathsf{N}$ *matrix in which* $A_{v,v'}$ *is equal to the multiplicity of the edge* (v, v') *for any* $v, v' \in V$.

We focus on finding the path between the starting vertex s and the ending vertex t. In the graph $\mathcal{G}_S$ defined above, s is the root vertex of the tree $\mathcal{T}_1$, and t is the root vertex of the tree $\mathcal{T}_{n/2+1}$. Here s is given to us directly at the beginning in the form of a bit-string representing it. We also give an indicator function for identifying t.

Definition 4.4.7. *The indicator function* f_t *satisfies:*

$$f_t(v) = \begin{cases} 1 & \text{if } v = t, \\ 0 & \text{otherwise.} \end{cases}$$

Problem 4.4.1 (Pathfinding Problem in $\mathcal{G}_S$)**.** *Given the adjacency list oracle with access to the regular sunflower graph, the starting vertex* $s = s_1$, *and an indicator function* $f_t(v)$ *(Definition 4.4.7), compute an* s-t *path. We fix* t *as* $t = s_{n/2+1}$.

It is illustrative to group some of the vertices in this multigraph together to reveal some additional structure in it.

Definition 4.4.8. *A supervertex* $S_{i,j}$ *is the set of vertices in the* j*-th layer of tree* $\mathcal{T}_i$ *in the regular sunflower graph defined in Definition 4.4.5.*

The cardinality of this set can be computed from Definition 4.4.5 to be

$$s_{i,j} = |S_{i,j}| = \begin{cases} 1, & \text{if } j = 1, \\ (d-1)^{j-2}(d-2), & \text{otherwise.} \end{cases} \tag{4.39}$$

We will call $S_{i,j}$ a *supervertex*. From this, we can compute the total number of vertices in the regular sunflower graph to be $\mathsf{N}_{\mathcal{G}_S} = n(d-1)^{m-1}$.

Let $e_{ij,kl}$ be the number of edges, counting multiplicity, between two sets of vertices $S_{i,j}, S_{k,l}$ in $\mathcal{G}_S = (\mathcal{V}, \mathcal{E})$, where $1 \le i, k \le n, 1 \le j, l \le m$. In other words,

$$e_{ij,kl} = \sum_{u \in S_{i,j}} \sum_{v \in S_{k,l}} A_{u,v}. \tag{4.40}$$

When $j = l = m$, $k = i + 1$ for $1 \le i \le n - 1$ or $k = 1, i = n$, we have $e_{ij,kl} = \frac{d-1}{2}(d-1)^{m-2}(d-2)$ because there are $\frac{d-1}{2}$ unions of random perfect matching between the vertices in $S_{i,m}$ and the vertices in $S_{i+1,m}$ for $1 \le i \le n-1$, also between $S_{1,m}$ and $S_{n,m}$.

When $j = l = 1$, $k = i + 1$ for $1 \le i \le n - 1$ or $k = 1, i = n$, we have $e_{ij,kl} = 1$ since there is only one edge between the vertex in $S_{i,1}$ and the vertex in $S_{i+1,1}$ for $1 \le i \le n-1$, and between $S_{n,1}$ and $S_{1,1}$.

When $i = k$ and $j = l + 1$ or $l - 1$, we have $e_{ij,kl} = \max\{s_{i,j}, s_{k,l}\}$ due to the tree structure of $\mathcal{T}_i$. For other cases of i, j, k, l, we have $e_{ij,kl} = 0$. To summarize:

$$e_{ij,kl} = \begin{cases} 1 & \text{if } j = l = 1 \text{ and } k = i + 1 \text{ for } 1 \le i \le n - 1; \\ 1 & \text{if } j = l = 1 \text{ and } k = 1, i = n; \\ \max\{s_{i,j}, s_{k,l}\} & \text{if } i = k \text{ and } j = l + 1 \text{ or } l - 1; \\ \frac{d-1}{2}(d-2)(d-1)^{m-2} & \text{if } j = l = m \text{ and } k = i + 1 \text{ for } 1 \le i \le n - 1; \\ \frac{d-1}{2}(d-2)(d-1)^{m-2} & \text{if } j = l = m \text{ and } k = 1, i = n; \\ 0 & \text{Otherwise.} \end{cases} \tag{4.41}$$

Because the graph is undirected, $e_{ij,kl} = e_{kl,ij}$.

By the construction of the graph is Definition 4.4.5, one can readily verify that the number of edges (counting multiplicity) connecting $u \in S_{i,j}$ to another fixed supervertex $S_{k,l}$ is the same as that of all $u \in S_{i,j}$. In other words, $\sum_{v \in S_{k,l}} A_{u,v}$ is the same for all $u \in S_{i,j}$. By Eq. (4.40), we therefore have

$$\sum_{v \in S_{k,l}} A_{u,v} = \frac{e_{ij,kl}}{s_{i,j}}, \quad \forall u \in S_{i,j}. \tag{4.42}$$

From the above discussion, we can see that the supervertices can be arranged into a graph: we link an edge between any pair of supervertices $S_{i,j}$ and $S_{k,l}$ for which $e_{ij,kl} \neq 0$.

The resulting graph is shown in Figure 4.16, and is referred to as the *supergraph*. We can see that the supergraph has a certain 2-dimensional structure that is useful for the analysis of its properties.

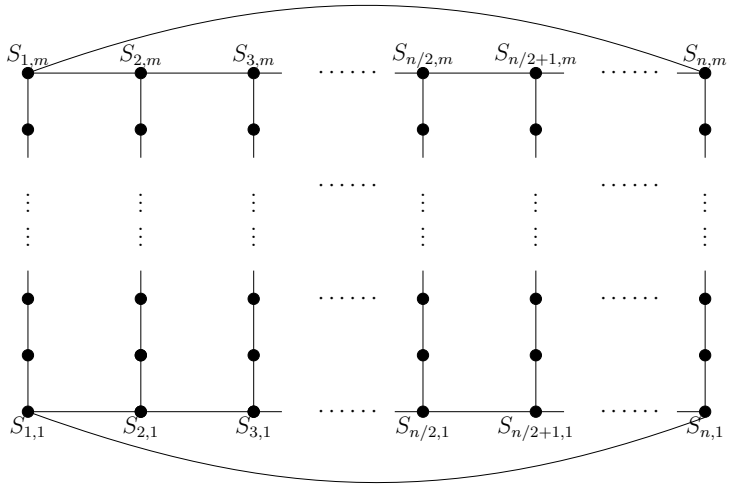

Figure 4.16. The supergraph consisting of the supervertices defined in Definition 4.4.8. Each pair of supervertices are linked by an edge if there exists an edge in the regular sunflower graph between two vertices contained in these two supervertices respectively.

4.4.3 The invariant subspace and effective Hamiltonian

In this section, we will discuss the adjacency matrix A of the regular sunflower graph $\mathcal{G}_S$ in Definition 4.4.5. In particular, certain symmetries of this graph make it possible for us to identify an invariant subspace of the adjacency matrix A. Within this invariant subspace, we can replace A with its restriction to this subspace, which we will call the *effective Hamiltonian* and denote it by H. Below we will provide a precise definition of H.

The supervertex defined in Definition 4.4.8 also corresponds to a quantum state, which we call the *supervertex state* and defined as

$$|S_{i,j}\rangle = \frac{1}{\sqrt{s_{i,j}}} \sum_{v \in S_{i,j}} |v\rangle \,, \tag{4.43}$$

where the value of $s_{i,j} = |S_{i,j}|$ is given in Eq. (4.39).

These supervertex states are all orthogonal to each other because supervertices do not overlap.

Definition 4.4.9 (The symmetric subspace $\mathcal{S}$)**.** *We define the mn-dimensional* symmetric subspace $\mathcal{S}$ *as follows*

$$\mathcal{S} = \text{span}\{|S_{i,j}\rangle : 1 \leq i \leq n, 1 \leq j \leq m\}. \tag{4.44}$$

Below, we will show that the symmetric subspace $\mathcal{S}$ is in fact an invariant subspace of the adjacency matrix A.

Lemma 4.4.2. *Let $\mathcal{G}_S$be the regular sunflower graph defined in Definition 4.4.5. Let A be the adjacency matrix of this multigraph as defined in Definition 4.4.6. Then for supervertex state $|S_{k,l}\rangle$ defined in Eq.* (4.43)*, we have*

$$A\,|S_{k,l}\rangle = \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{e_{ij,kl}}{\sqrt{s_{i,j}s_{k,l}}}\,|S_{i,j}\rangle\,,$$

where $e_{ij,kl}$ is the number of edges linking vertices in $S_{i,j}$ (defined in Definition 4.4.8) to those in $S_{k,l}$, and $s_{i,j} = |S_{i,j}|$.

Proof. We first observe that we only need to concern ourselves with the subgraph that is the regular sunflower graph $\mathcal{G}_S$, since the isolated vertices are not contained, or connected to, any supervertex.

$$\begin{aligned} A\,|S_{kl}\rangle =& \frac{1}{\sqrt{s_{kl}}}\sum_{v\in S_{kl}} A\,|v\rangle = \frac{1}{\sqrt{s_{kl}}}\sum_{v\in S_{kl}}\sum_{u} A_{u,v}\,|u\rangle \\ =& \sum_{ij}\frac{1}{\sqrt{s_{kl}}}\sum_{u\in S_{ij}}\sum_{v\in S_{kl}} A_{u,v}\,|u\rangle = \sum_{ij}\frac{1}{\sqrt{s_{kl}}}\sum_{u\in S_{ij}}\frac{e_{ij,kl}}{s_{ij}}\,|u\rangle \\ =& \sum_{ij}\frac{e_{ij,kl}}{\sqrt{s_{ij}s_{kl}}}\,|S_{ij}\rangle\,. \end{aligned}$$

where the fourth equality comes from Eq. (4.42), and the last equality comes from the definition of the supervertex state in Eq. (4.43). □

We define, with $e_{ij,kl}$ and $s_{i,j}$ as in Lemma 4.4.2,

$$\tilde{H}_{ij,kl} = \frac{e_{ij,kl}}{\sqrt{s_{i,j}s_{k,l}}}, \tag{4.45}$$

and these numbers serve as matrix elements that completely determine how A acts on the invariant subspace $\mathcal{S}$. Because they will play an important role in our analysis, we will compute them explicitly here.

Lemma 4.4.3. *$\tilde{H}_{ij,kl}$ defined in Eq.* (4.45) *takes the following value:*

$$\tilde{H}_{ij,kl} = \begin{cases} 1 & \text{if } j = l = 1 \text{ and } k = i + 1 \text{ for } 1 \le i \le n - 1; \\ 1 & \text{if } j = l = 1 \text{ and } k = 1, i = n; \\ \sqrt{d-1} & \text{if } 2 \le j \le m - 1, 2 \le l \le m \text{ and } i = k; \\ \sqrt{d-2} & \text{if } j = 1, l = 2 \text{ and } i = k; \\ \frac{d-1}{2} & \text{if } j = l = m \text{ and } k = i + 1 \text{ for } 1 \le i \le n - 1; \\ \frac{d-1}{2} & \text{if } j = l = m \text{ and } k = 1, i = n; \\ 0 & \text{otherwise;} \end{cases}$$

and $\tilde{H}_{ij,kl} = \tilde{H}_{kl,ij}$.

Proof. Recall the values of $s_{i,j}$ given in Eq. (4.39) and those of $e_{ij,kl}$ given in Eq. (4.41), we can compute $\tilde{H}_{ij,kl} = \frac{e_{ij,kl}}{\sqrt{s_{i,j} s_{k,l}}}$ directly. We have $\tilde{H}_{ij,kl} = \tilde{H}_{kl,ij}$ because the graph is undirected and $e_{ij,kl} = e_{kl,ij}$. □

Any matrix can be regarded as a linear map, and we will look at the adjacency matrix A from this viewpoint. Because the symmetric subspace $\mathcal{S}$ is an invariant subspace of A according to Lemma 4.4.2, we can then *restrict* the linear map A to this subspace to obtain a well-defined linear map.

Definition 4.4.10 (The effective Hamiltonian)**.** *The effective Hamiltonian $H : \mathcal{S} \to \mathcal{S}$ is the restriction of the adjacency matrix $A : \mathcal{H} \to \mathcal{H}$ (Definition 4.4.6) of the regular sunflower graph (Definition 4.4.5) to its invariant subspace $\mathcal{S}$ given in Eq.* (4.44)*. In other words, we define $H : \mathcal{S} \to \mathcal{S}$ to be*

$$H : \mathcal{S} \ni |\phi\rangle \mapsto A |\phi\rangle .$$

We name H the "effective Hamiltonian" because if we consider quantum dynamics described by the time evolution operator e^{-iAt}, then if the initial state is in $\mathcal{S}$ the dynamics will be completely captured by the restriction of A to $\mathcal{S}$, and therefore H serves as a Hamiltonian governing the time evolution. This was a key idea in the quantum walk algorithm in [CCD$^+$03].

We will use the following result when proving the correctness and efficiency of our algorithm:

Lemma 4.4.4. *Let A be the adjacency matrix of the regular sunflower graph and H be the effective Hamiltonian restricted to the symmetric subspace $\mathcal{S}$ as defined Definition 4.4.9. Then for any vector $|v\rangle \in \mathcal{S}$, we have $f(A)|v\rangle = f(H)|v\rangle$ where $f(\cdot)$ is a polynomial function.*

Proof. By linearity, we only need to prove for monomials $f(x) = x^k$. We do so by induction on k. When $k = 1$ we have $A|v\rangle = H|v\rangle$ by definition. If we have $A^{k-1}|v\rangle = H^{k-1}|v\rangle$, then

$$A^k|v\rangle = A(A^{k-1}|v\rangle) = A(H^{k-1}|v\rangle) = H(H^{k-1}|v\rangle) = H^k|v\rangle.$$

□

Compared to the adjacency matrix A, the effective Hamiltonian H has the advantage that it only acts on a mn-dimensional subspace, and therefore can be represented by a matrix of size $mn \times mn$. We will next proceed to construct this representation and reveal further structure through it.

We first define an isometry $V_{\mathcal{S}} : \mathbb{C}^m \otimes \mathbb{C}^n \to \mathcal{S}$ through

$$V_{\mathcal{S}} : \sum_{i=1}^{n}\sum_{j=1}^{m} \Phi_{ij}|\mathfrak{b}_j\rangle|\mathfrak{a}_i\rangle \mapsto \sum_{i=1}^{n}\sum_{j=1}^{m} \Phi_{ij}|S_{i,j}\rangle, \tag{4.46}$$

for any $\Phi_{ij} \in \mathbb{C}$, and where $|\mathfrak{a}_i\rangle \in \mathbb{C}^n$ is the n-dimensional vector with 1 on the i-th entry and 0 everywhere else, and $|\mathfrak{b}_j\rangle \in \mathbb{C}^m$ is the m-dimensional vector with 1 on the j-th entry and 0 everywhere else. Here we use $|\mathfrak{b}_j\rangle|\mathfrak{a}_i\rangle$ rather than $|\mathfrak{a}_i\rangle|\mathfrak{b}_j\rangle$ in order to reveal the block-tridiagonal structure of the matrix representation of the effective Hamiltonian that is to be introduced later. It can be easily verified that this is a bijective isometry, and that its inverse $V_{\mathcal{S}} : \mathcal{S} \to \mathbb{C}^m \otimes \mathbb{C}^n$ is

$$V_{\mathcal{S}}^{-1} : \sum_{i=1}^{n}\sum_{j=1}^{m} \Phi_{ij}|S_{i,j}\rangle \mapsto \sum_{i=1}^{n}\sum_{j=1}^{m} \Phi_{ij}|\mathfrak{b}_j\rangle|\mathfrak{a}_i\rangle. \tag{4.47}$$

With this isometry, we now consider the linear map $V_{\mathcal{S}}^{-1}HV_{\mathcal{S}}$. This linear map maps $\mathbb{C}^m \otimes \mathbb{C}^n$ to itself and therefore can be written as a $mn \times mn$ matrix. We therefore define

Definition 4.4.11. *We call $\tilde{H} = V_{\mathcal{S}}^{-1}HV_{\mathcal{S}}$ the* matrix representation *of the effective Hamiltonian H, for the effective Hamiltonian H defined in Definition 4.4.10 and $V_{\mathcal{S}}$ given in Eq.* (4.46).

Because $V_{\mathcal{S}}$ is a bijective isometry, H and $\tilde{H}$ are unitarily equivalent to each other, and therefore have the same spectrum. Their eigenvectors also have one-to-one correspondence: for any eigenvector $|\Psi\rangle$ of H, $V_{\mathcal{S}}^{-1}|\Psi\rangle$ is an eigenvector of $\tilde{H}$ corresponding to the same eigenvalue, and vice versa.

We will be able to obtain a more explicit characterization of $\tilde{H}$ compared to H, which helps us to use $\tilde{H}$ to analyze H.

Lemma 4.4.5. *The matrix entry of $\tilde{H}$ on the $((j-1)n+i)$-th row and $((l-1)n+k)$-th column is $\tilde{H}_{ij,kl}$ in Eq.* (4.45).

Proof. We only need to observe that

$$\tilde{H}\,|\mathfrak{b}_l\rangle\,|\mathfrak{a}_k\rangle = V_{\mathcal{S}}^{-1}HV_{\mathcal{S}}\,|\mathfrak{b}_l\rangle\,|\mathfrak{a}_k\rangle = V_{\mathcal{S}}^{-1}H\,|S_{k,l}\rangle = V_{\mathcal{S}}^{-1}\sum_{ij}\tilde{H}_{ij,kl}\,|S_{k,l}\rangle = \sum_{ij}\tilde{H}_{ij,kl}\,|\mathfrak{b}_j\rangle\,|\mathfrak{a}_i\rangle\,,$$

where we have used Lemma 4.4.2 and the definition of $\tilde{H}_{ij,kl}$ in Eq. (4.45). Also note that $|\mathfrak{b}_j\rangle\,|\mathfrak{a}_i\rangle$, when written as a vector in $\mathbb{C}^{nm}$, has 1 on the $((j-1)n+i)$-th entry and 0 everywhere else. □

With the matrix entries available, we can now write down the matrix $\tilde{H}$ explicitly:

$$\tilde{H} = \begin{pmatrix} D_0 & t_1 I & & & \\ t_1 I & 0 & t_2 I & & \\ & t_2 I & 0 & \dots & \\ & & \ddots & \ddots & t_{m-1}I \\ & & & t_{m-1}I & \gamma D_0 \end{pmatrix}, \tag{4.48}$$

where D_0 is the $n \times n$ adjacency matrix associated with the cycle graph C_n given in Eq. (4.51), and $t_1 = \sqrt{d-2}$, $t_2 = t_3 = \cdots = t_{m-1} = \sqrt{d-1}$, $\gamma = \frac{d-1}{2}$. One may notice that $\tilde{H}$ has the same sparsity pattern, i.e., the position of the non-zero entries, as the adjacency matrix of the supergraph in Figure 4.16. This is not a coincidence since $\tilde{H}_{ij,kl} \neq 0$ if and only if $S_{i,j}$ and $S_{k,l}$ are linked in the supergraph, as can be seen from Eq. (4.45). We observe that $\tilde{H}$ is a block tridiagonal matrix, and this is useful for analyzing its spectral properties.

4.4.4 Expansion properties of $\mathcal{G}_S$

In this section, we will investigate the expansion properties of the regular sunflower graph $\mathcal{G}_S$. First, we note that each pair of adjacent sets $S_{i,m}$ and $S_{i+1,m}$ for $1 \le i \le n-1$, also between $S_{1,m}$ and $S_{n,m}$ form a random bipartite graph, with the connectivity given by $(d-1)/2$ random perfect matchings. The following lemma gives us the expansion property of a bipartite graph with N vertices on each side whose connectivity is given through D random perfect matchings:

Lemma 4.4.6. *Let L and R be two sets of vertices with $|L| = |R| = N$. Link L and R through $D \ge 3$ random perfect matchings. Denote the resulting graph by $G_B = (V_B, E_B)$, and let $\chi = 2/3$, $\delta = 1/(2\log N)$. Then G_B has the following expansion properties with probability $1 - \Theta(1/N^{2\log(3/2)(1-\delta)})$:*

(i) For any subset $L' \subseteq L$ and $|L'| \le \chi N$, we have $|\Gamma(L')| = |\Gamma(L')\backslash L'| \ge (1+\delta)|L'|$, where $\Gamma(L')$ denotes the neighborhood of L' as defined in Definition 4.4.1.

(ii) For any subset $T \subseteq L \cup R$ and $|T| \le N$, we have $|\Gamma(T)\backslash T| \ge \frac{\delta}{2}|T|$.

In other words, with probability $1 - \Theta(1/N^{2\log(3/2)(1-\delta)})$, this bipartite regular graph is a mild expander graph.

The proof of Lemma 4.4.6 follows the proof of [Kow19, Theorem 4.1.1], so we defer the proof to Appendix B.2. In the context of subgraphs formed by two adjacent sets of vertices between $S_{i,m}$ and $S_{i+1,m}$ for $1 \le i \le n-1$, also between $S_{1,m}$ and $S_{n,m}$ of the regular sunflower graph $\mathcal{G}_S$, we have $D = (d-1)/2$, and $N = (d-2)(d-1)^{m-2}$. We will next use this lemma to study the expansion property of $\mathcal{G}_S$.

Theorem 4.4.7. *Let $m = \Theta(n)$ and $d \ge 7$ be a odd integer constant. With at least $1 - \exp(-\Omega(n))$ probability, the d-regular sunflower graph $\mathcal{G}_S = (\mathcal{V}, \mathcal{E})$ defined in Definition 4.4.5 is a mild expander graph as defined in Definition 4.4.4. More precisely, for any subset $T \subseteq \mathcal{V}$, and $|T| \le \frac{|\mathcal{V}|}{2}$, we have $|\Gamma(T)\backslash T| \ge \frac{1}{\text{poly}\log(|\mathcal{V}|)} \cdot |T|$.*

Proof. First, we will introduce some parameters to be used later. Let $\chi = \frac{2}{3}$, $\delta = 1/\Theta(n)$ and $N = (d-2)(d-1)^{m-2}$. We note that

$$|\mathcal{V}| = n(d-1)^{m-1} = \Theta(nN).$$

To prove the theorem, it suffices to show that with at least $1 - \exp(-\Omega(n))$ probability, for any subset $T \subseteq \mathcal{V}$ and $|T| \le |\mathcal{V}|/2$, we have $|\Gamma(T)\backslash T| \ge \Omega(|T|/n^3)$.

Let $S_i = \bigcup_{j=1}^{m} S_{i,j}$ be the vertices of i-th constituent tree $\mathcal{T}_i$ as defined in Definition 4.4.5 and $T_i = T \cap S_i$. Let $A_i = T_i \cap S_{i,m}$ and $B_i = T_i \backslash A_i$. In other words, A_i contains the elements of T_i (and therefore T) that are also the leaf vertices of the constituent tree $\mathcal{T}_i$, while B_i contains the elements of T_i that are internal vertices of $\mathcal{T}_i$. We want to show that T has at least $\Omega(|T|/n^3)$ adjacent vertices that are not in itself. Therefore, we assume towards contradiction that with probability at least $\exp(-o(n))$, we can find a subset T such that

$$|\Gamma(T) \setminus T| = o(|T|/n^3), \quad |T| \leq |\mathcal{V}/2|. \tag{4.49}$$

Observe that the number of edges out of the set T is at least $\big||A_i| - |A_{i+1}|\big|$ in the bipartite part between trees i and $i+1$, because the leaves of $\mathcal{T}_i$ and $\mathcal{T}_{i+1}$ are linked through $(d-1)/2$ random perfect matchings, each of which is a bijection. Therefore we have

$$\big||A_i| - |A_{i+1}|\big| = o(|T|/n^3).$$

Hence for each A_i and A_j, we have

$$\big||A_i| - |A_j|\big| = o(|T|/n^2).$$

Without loss of generality, assume that $|T_1| \geq |T_i|$ for $i \in [n]$, then we have $|T_1| \geq \frac{1}{n}|T|$.

- We first consider $|A_1| \leq \frac{1}{3}|T_1|$. Note that because B_1 is a subset of internal vertices of $\mathcal{T}_1$, whose vertices have at least $d-2$ children each. With this fact we can prove that $|\Gamma(B_1) \setminus B_1| \geq (d-2)|B_1| \geq |B_1| \geq \frac{2}{3}|T_1|$.

  This can be proven by considering each connected component of B_1. By induction on the size of a connected component S we can show that it has at least $(d-2)|S|$ child vertices that are not contained in S. Because different connected components cannot share child vertices due to the tree structure of $\mathcal{T}_1$, we can show that $|\Gamma(B_1) \setminus B_1| \geq (d-2)|B_1|$.

  Again because the vertices in B_1 are all internal vertices of $\mathcal{T}_1$, we have $(\Gamma(B_1) \setminus B_1) \cap T = (\Gamma(B_1) \setminus B_1) \cap A_1$. Thus there are at least $\frac{1}{3}|T_1|$ vertices inside $\Gamma(B_1)$ that are also outside the set T, that is, $|\Gamma(T) \setminus T| \geq \frac{1}{3}|T_1| \geq \frac{1}{3n}|T|$. This contradicts the assumption in Eq. (4.49).

- If $|A_1| \geq \frac{1}{3}|T_1| \geq \frac{1}{3n}|T|$, we have $|A_1| \leq \frac{10}{9}\frac{1}{n}|T|$. This is true by the following

argument: Note that $|A_1| + |A_2| + \ldots + |A_n| \leq |T|$ and $||A_i| - |A_j|| = o(|T|/n^2)$, so $n|A_1| \leq |T| + o(|T|/n)$. Hence $|A_1| \leq \frac{1}{n}|T| + o(|T|/n^2) \leq \frac{10}{9} \cdot \frac{1}{n}|T|$.

Since $|T| \leq |\mathcal{V}|/2$, $|\mathcal{V}|/n = \frac{d-1}{d-2}N$ and $\frac{d-1}{d-2} \leq \frac{6}{5}$, we have $|A_1| \leq \frac{10}{9} \cdot \frac{1}{n}|T| \leq \frac{10}{9} \cdot \frac{1}{n}\frac{|\mathcal{V}|}{2} = \frac{5}{9}\frac{d-1}{d-2}N \leq \chi N$. That is $|A_1| \leq \chi N$. This tells us that $|A_1|$ is small enough for the expansion property in Lemma 4.4.6 (i) to hold. By Lemma 4.4.6 (i), and because A_1 is contained in one side of the bipartite regular graph between $S_{1,m}$ and $S_{2,m}$, we know that

$$|(S_{2,m} \cap \Gamma(A_1)) \setminus A_1| \geq (1+\delta)|A_1| \geq \frac{1}{3n}(1+\delta)|T|, \tag{4.50}$$

with probability at least $1 - \exp(-\Omega(n))$. Note that in the above equation, it is redundant to exclude the set A_1, because $S_{2,m} \cap \Gamma(A_1)$ does not intersect with $A_1 \subset S_{1,m}$. We keep it there to keep the notation consistent with Lemma 4.4.6. The number of vertices in $S_{2,m}$ that are adjacent to A_1 and at the same time not in T is

$$|(S_{2,m} \cap \Gamma(A_1))\setminus T| \geq |(S_{2,m} \cap \Gamma(A_1))\setminus A_1| - |A_2| \geq (1+\delta)|A_1| - |A_2| \geq \Omega(|T|/n^2),$$

where in the second inequality we have used $||A_1| - |A_2|| = o(|T|/n^3)$. Because $S_{2,m} \cap \Gamma(A_1) \subset \Gamma(T)$, the above inequalities imply that $|\Gamma(T) \setminus T| \geq \Omega(|T|/n^2)$. Therefore for the assumption in Eq. (4.49) to hold with probability at least $\exp(-o(n))$, we will need Eq. (4.50) to fail with probability at least $\exp(-o(n))$. But Eq. (4.50) holds with probability at least $1 - \exp(-\Omega(n))$ by Lemma 4.4.6 (i), and therefore we have reached a contradiction.

Therefore, for any set T with $|T| \leq |\mathcal{V}|/2$, we have $|\Gamma(T) \setminus T| = \Omega(|T|/n^3)$. □

4.4.5 Spectral properties of the effective Hamiltonian

In this section, we analyze the spectral properties of the effective Hamiltonian H (defined in Definition 4.4.10) associated with the regular sunflower graph $\mathcal{G}_S$ in Definition 4.4.5. Specifically, we will show that there is a unique 0-eigenvector of H that overlaps the starting state $|s\rangle = |S_{1,1}\rangle$, and the spectral gap of H around 0 is bounded from below by $1/\operatorname{poly}(m,n)$.

Our quantum algorithm is built on the observation that the 0-eigenvector of the adjacency matrix of the graph sometimes yields useful information. This is also the underlying idea of the quantum algorithm for exit-finding in [BLH23]. In the following, we

first look at some example graphs and the corresponding 0-eigenvectors of the adjacency matrices. These examples will be useful for our analysis of the regular sunflower graph we construct. Unless otherwise specified, we assume that all graphs are unweighted, which means that all edges have weight 1.

Definition 4.4.12 (Adjacency matrix of a cycle graph C_n)**.** *A* cycle graph *consists of n vertices $\{v_1, v_2, \ldots, v_n\}$ with edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$. The adjacency matrix D_0 of C_n is defined as*

$$D_0 = \begin{pmatrix} 0 & 1 & & & 1 \\ 1 & 0 & 1 & & \\ & 1 & 0 & \ddots & \\ & & \ddots & \ddots & 1 \\ 1 & & & 1 & 0 \end{pmatrix}. \tag{4.51}$$

Definition 4.4.13 (Adjacency matrix of weighted path graph P_m)**.** *A* path graph *consists of n vertices $\{v_1, v_2, \ldots, v_m\}$ with edges $\{v_i, v_{i+1}\}$ for $i = 1, 2, \ldots, n-1$. The* path graph *P_m is weighted if each edge $\{v_i, v_{i+1}\}$ has weight $t_i \in \mathbb{R}$. The adjacency matrix D_1 of P_m is defined as*

$$D_1 = \begin{pmatrix} 0 & t_1 & & & \\ t_1 & 0 & t_2 & & \\ & t_2 & 0 & \ddots & \\ & & \ddots & \ddots & t_{m-1} \\ & & & t_{m-1} & 0 \end{pmatrix}_{m \times m}. \tag{4.52}$$

Example 4.4.8 (0-eigenvector of adjacency matrix D_0 of a cycle graph C_n [Spi19, Lemma 6.5.1])**.** *Let n be an integer that is a multiple of 4 and D_0 be the adjacency matrix of a cycle graph C_n, then eigenvalues of D_0 are $2\cos(\frac{2\pi l}{n})$ for $l = 1, \ldots, n$ and the two orthogonal 0-eigenvectors x, y of D_0 are the following:*

$$x_l = \cos(l\pi/2) \text{ where } l = 1, 2, \ldots, n. \tag{4.53}$$

$$y_l = \sin(l\pi/2) \text{ where } l = 1, 2, \ldots, n. \tag{4.54}$$

When it comes to the 0-eigenvector of the adjacency matrix of a weighted path graph of odd length, one can also compute the unique 0-eigenvector implicitly as in the proof of [BLH23, Lemma 3.2]. We formulate this result as follows:

Example 4.4.9 (0-eigenvector of adjacency matrix D_1 of an weighted path graph P_m)**.** *Let m be an odd integer and d be a positive integer. Let D_1 be the adjacency matrix of the path graph P_m with edges weights $t_1, t_2, \ldots, t_{m-1}$, then the unique 0-eigenvector of D_1 is $|x\rangle = (x_1, x_2, \cdots, x_m)^\top$, where*

$$x_l = \begin{cases} 0 & \text{for even } l = 2, 4, \ldots, m-1. \\ \prod_{k=1}^{(l-1)/2} \left(-\frac{t_{2k-1}}{t_{2k}}\right) x_1 & \text{for odd } l = 1, 3, \ldots, m. \end{cases} \tag{4.55}$$

In particular, we will use the case where $t_1 = \sqrt{d-2}$, $t_2 = \ldots = t_{m-1} = \sqrt{d-1}$. In this case, for even $l \geq 2$, we have

$$|x_l| = \sqrt{\frac{d-2}{d-1}} |x_1|. \tag{4.56}$$

To achieve the above objectives it suffices to study the matrix $\tilde{H}$. Because of the definition $\tilde{H} = V_{\mathcal{S}}^{-1} H V_{\mathcal{S}}$ in Definition 4.4.11, we know that $\tilde{H}$ and H share the same spectrum, and there is a bijective correspondence between their respective eigenstates. We have already written down $\tilde{H}$ as a block tridiagonal matrix. There is a more compact way to express $\tilde{H}$ which can help us understand the structure of its eigenvalues and eigenvectors. We let

$$D_1 = \begin{pmatrix} 0 & t_1 & & & \\ t_1 & 0 & t_2 & & \\ & t_2 & 0 & \ddots & \\ & & \ddots & \ddots & t_{m-1} \\ & & & t_{m-1} & 0 \end{pmatrix}_{m \times m} . \tag{4.57}$$

Then we can also rewrite $\tilde{H}$ as follows:

$$\tilde{H} = (|\mathfrak{b}_1\rangle\langle\mathfrak{b}_1| + \gamma\,|\mathfrak{b}_m\rangle\langle\mathfrak{b}_m|) \otimes D_0 + D_1 \otimes I, \tag{4.58}$$

where $\mathfrak{b}_j$ is the m-dimensional vector with 1 on the j-th entry and 0 everywhere else.

From the above, we can see that, to understand the properties of H, we only need to study $\tilde{H}$, which we will do next. Using the compact form representation of $\tilde{H}$ in Eq. (4.58), the following Lemma 4.4.10 shows that the eigenvalues and eigenvectors of $\tilde{H}$ have a particular structure.

Lemma 4.4.10. *Let $\mu_l = 2\cos(2l\pi/n)$ and let $|\phi_l\rangle$ be the corresponding eigenvector of D_0 (defined in Eq. (4.51)), with $l = 1, 2, \ldots, n$. Let $\lambda_j(a, b)$ be the j-th smallest eigenvalue of $H_1(a, b)$ with $j = 1, 2, \ldots, m$, where*

$$H_1(a,b) = \begin{pmatrix} a & t_1 & & & \\ t_1 & 0 & t_2 & & \\ & t_2 & 0 & \ddots & \\ & & \ddots & \ddots & t_{m-1} \\ & & & t_{m-1} & b \end{pmatrix}_{m\times m}. \tag{4.59}$$

Let $\gamma = \frac{d-1}{2}$ and $|\psi_j^l\rangle$ be the $\lambda_j(\mu_l, \gamma\mu_l)$-eigenvector of $H_1(\mu_l, \gamma\mu_l)$ with $1 \le l \le n, 1 \le j \le m$. The eigenvalues and eigenvectors of $\tilde{H}$ in Eq. (4.48) are

$$\lambda_j(\mu_l, \gamma\mu_l) \text{ and } |\psi_j^l\rangle\,|\phi_l\rangle \text{ for } l = 1, 2, \ldots, n, j = 1, 2, \ldots, m.$$

Proof. We prove this lemma by examining each eigenvalue and eigenvector of $\tilde{H}$. That is, for each $l = 1, 2, \ldots, n, j = 1, 2, \ldots, m$,

$$\begin{aligned} \tilde{H}\,|\psi_j^l\rangle\,|\phi_l\rangle &= \left[\left(|\mathfrak{b}_1\rangle\langle\mathfrak{b}_1| + \gamma\,|\mathfrak{b}_m\rangle\langle\mathfrak{b}_m|\right) \otimes D_0 + D_1 \otimes I\right] |\psi_j^l\rangle\,|\phi_l\rangle \\ &= \left[\left(\mu_l\,|\mathfrak{b}_1\rangle\langle\mathfrak{b}_1| + \gamma\mu_l\,|\mathfrak{b}_m\rangle\langle\mathfrak{b}_m| + D_1\right)|\psi_j^l\rangle\right] \otimes |\phi_l\rangle \\ &= \lambda_j(\mu_l, \gamma\mu_l)\,|\psi_j^l\rangle\,|\phi_l\rangle, \end{aligned} \tag{4.60}$$

where in the second equation we have used $D_0\,|\phi_l\rangle = \mu_l\,|\phi_l\rangle$, and in the third equation

we have used

$$H_1(\mu_l, \gamma\mu_l)\,|\psi_j^l\rangle = \left(\mu_l\,|\mathfrak{b}_1\rangle\,\langle\mathfrak{b}_1| + \gamma\mu_l\,|\mathfrak{b}_m\rangle\,\langle\mathfrak{b}_m| + D_1\right)|\psi_j^l\rangle = \lambda_j(\mu_l, \gamma\mu_l)\,|\psi_j^l\rangle\,. \tag{4.61}$$

Eq. (4.60) then provides us with $n \times m$ eigenpairs, which are all eigenpairs of $\tilde{H}$ because the dimension of the vector space is also $n \times m$. □

Using this lemma, we will identify the 0-eigenspace of $\tilde{H}$, and lower bound spectral gap around the eigenvalue 0. Since the eigenvalues of D_0 are readily available as $\mu_l = 2\cos(2\pi l/n)$, we can compute the spectrum of $\tilde{H}$ by computing the eigenvalues $\lambda_j(\mu_l, \gamma\mu_l)$ of the tridiagonal matrices $H_1(\mu_l, \gamma\mu_l)$ for $j = 1, 2, \ldots, m$, $l = 1, 2, \ldots, n$. Using this fact, we next show that $\lambda_j(\mu_l, \gamma\mu_l) = 0$ if and only if $\mu_l = 0$ and $\lambda_j(\mu_l, \gamma\mu_l) = \Omega(1/(mn))$ for $\mu_l \neq 0$. Specifically, we prove the first result by computing the determinant of H and the latter by computing the inverse of $H_1(\mu_l, \gamma\mu_l)$ for the case of $\mu_l \neq 0$.

We note that the determinant of a general tridiagonal matrix can be computed through [EMK06, Theorem 2.1]. We apply this result to the special class of tridiagonal matrices $H_1(a, b)$ as defined in Eq. (4.59).

Lemma 4.4.11 (Determinant of tridiagonal matrix $H_1(a, b)$ [EMK06, Theorem 2.1])**.** *Let $\beta = (\beta_0, \ldots, \beta_m)$ be a $m + 1$ dimensional vector defined as follows,*

$$\beta_i = \begin{cases} 1 & \text{if } i = 0 \\ a & \text{if } i = 1 \\ -t_{i-1}^2\beta_{i-2} & \text{if } i = 2, \ldots, m-1; \\ b - t_{m-1}^2\beta_{m-2} & \text{if } i = m. \end{cases} \tag{4.62}$$

The determinant of $H_1(a, b)$ is equal to β_m.

Corollary 4.4.12. *Let $t_1 = \sqrt{d-2}$, $t_2 = t_3 = \cdots = t_{m-1} = \sqrt{d-1}$, $\gamma = \frac{d-1}{2}$ and let m be an odd integer. For every $\mu_l = 2\cos(2\pi l/n)$ with $l = 1, 2 \ldots, n$, the eigenvalue $\lambda_j(\mu_l, \gamma\mu_l)$ of $H_1(\mu_l, \gamma\mu_l)$ is equal to 0 if and only if $\mu_l = 0$.*

Proof. By Lemma 4.4.11 and m being an odd integer, the determinant of $H_1(a, b)$ is equal to $\beta_m = b + \prod_{k=1}^{\frac{m-1}{2}} t_{2k}^2 a$ when $m = 1 \bmod 4$ or $\beta_m = b - \prod_{k=1}^{\frac{m-1}{2}} t_{2k}^2 a$ when $m = 3 \bmod 4$.

For the matrix $H_1(\mu_l, \gamma\mu_l)$, we have

$$\beta_m = \gamma\mu_l \pm \prod_{k=1}^{\frac{m-1}{2}} t_{2k}^2 \mu_l = \left(\frac{d-1}{2} \pm (d-1)^{m-1}\right)\mu_l.$$

Since $\frac{d-1}{2} \pm (d-1)^{m-1} \neq 0$, we have $\beta_m = 0$ if and only if $\mu_l = 0$. □

When $\mu_l \neq 0$, we next show that all the eigenvalues of $H_1(\mu_l, \gamma\mu_l)$ are far away from 0, that is, $|\lambda_j(\mu_l, \gamma\mu_l)| = \Omega(1/(mn^2))$. The idea is to compute the inverse of the tridiagonal matrix $H_1(\mu_l, \gamma\mu_l)$, for which we have the following lemma:

Lemma 4.4.13. *Let $H_1(a, b)$ be as defined in Eq. (4.59), with odd m, $t_1 = \sqrt{d-2}$, $t_2 = t_3 = \cdots = t_{m-1} = \sqrt{d-1}$, $\gamma = \frac{d-1}{2}$, for integer $d \geq 3$. Let $\mu_l = 2\cos(2\pi l/n)$ with $l = 0, 1 \ldots, n-1$ satisfying $\mu_l \neq 0$. Let $a = \mu_l$ and $b = \gamma\mu_l$. Then $\|H_1^{-1}(a, b)\| = O(mn^2)$.*

The proof of this lemma can be found in Appendix B.3. Because the eigenvalues of $H_1^{-1}(a, b)$ are exactly $1/\lambda_j(\mu_l, \gamma\mu_l)$, for $j = 1, 2, \cdots, m$, the above lemma tells us that $1/|\lambda_j(\mu_l, \gamma\mu_l)| = O(mn^2)$, and hence $|\lambda_j(\mu_l, \gamma\mu_l)| = \Omega(1/(mn^2))$ for l such that $\mu_l \neq 0$.

When $\mu_l = 0$, the corresponding $\lambda_j(\mu_l, \gamma\mu_l) = \lambda_j(0, 0)$ can still be non-zero, and the following lemma helps us bound these eigenvalues away from 0:

Lemma 4.4.14. *Let D_1 be as defined in Eq. (4.57), in which $t_1 = \sqrt{d-2}$, $t_2 = \cdots = t_{m-1} = \sqrt{d-1}$. Then D_1 has a non-degenerate 0-eigenstate $|\Psi\rangle = (\Psi_1, \Psi_2, \cdots, \Psi_m)^\top$ where*

$$\Psi_j = \prod_{k=1}^{(j-1)/2} \left(-\frac{t_{2k-1}}{t_{2k}}\right)\Psi_1 = (-1)^{(j-1)/2}\sqrt{\frac{d-2}{d-1}}\Psi_1, \quad \text{for all odd } j \geq 2,$$

and $\Psi_j = 0$ for even j. Moreover, 0 is separated from the rest of the spectrum of D_1 by a gap of at least $2\sqrt{d-2}/(m-1)$.

The proof can be found in Appendix B.3. Because $H_1(0, 0) = D_1$, the above lemma tells us that $|\lambda_j(0, 0)| \geq 2\sqrt{d-2}/(m-1)$ for all j such that $\lambda_j(0, 0) \neq 0$. Moreover, this lemma also provides an explicit formula for $|\Psi\rangle$, i.e., the 0-eigenvector of D_1, which is needed for constructing the 0-eigenvector of $\tilde{H}$.

We will then summarize the above discussions into the proof of the following Theorem 4.4.15.

Theorem 4.4.15. *For the matrix $\tilde{H}$ in Eq. (4.48), the following statements are true:*

(i) The non-zero eigenvalues of $\tilde{H}$ are bounded away from zero by $\Omega(1/(mn^2))$.

(ii) Let $|\Psi\rangle$ be the normalized 0-eigenvector of matrix D_1, and let $|\Phi^{\text{even}}\rangle$ and $|\Phi^{\text{odd}}\rangle$ be the two orthogonal 0-eigenvectors of matrix D_0 defined as $|\Phi^{\text{even}}\rangle = (\Phi_1^{\text{even}}, \Phi_2^{\text{even}}, \ldots, \Phi_n^{\text{even}})^\top$ and $|\Phi^{\text{odd}}\rangle = (\Phi_1^{\text{odd}}, \Phi_2^{\text{odd}}, \ldots, \Phi_n^{\text{odd}})^\top$, where

$$\Phi_l^{\text{even}} = \frac{1}{\sqrt{n/2}} \cos(l\pi/2), \quad \Phi_l^{\text{odd}} = \frac{1}{\sqrt{n/2}} \sin(l\pi/2). \tag{4.63}$$

The 0-eigenspace of $\tilde{H}$ is 2-dimensional and is spanned by the orthonormal basis

$$|\chi^{\text{even}}\rangle = |\Psi\rangle\,|\Phi^{\text{odd}}\rangle, \quad |\chi^{\text{odd}}\rangle = |\Psi\rangle\,|\Phi^{\text{even}}\rangle.$$

(iii) The two quantum states $|\chi^{\text{even}}\rangle$ and $|\chi^{\text{odd}}\rangle$ satisfy the following conditions: for all even $1 \le i \le n$, we have $|\langle\chi^{\text{even}}|\mathfrak{b}_1, \mathfrak{a}_i\rangle| = \Omega(1/\sqrt{mn})$, $|\langle\chi^{\text{odd}}|\mathfrak{b}_1, \mathfrak{a}_i\rangle| = 0$.[1] For all odd $1 \le i \le n$, we have $|\langle\chi^{\text{odd}}|\mathfrak{b}_1, \mathfrak{a}_i\rangle| = \Omega(1/\sqrt{mn})$, $|\langle\chi^{\text{even}}|\mathfrak{b}_1, \mathfrak{a}_i\rangle| = 0$.

Proof of Theorem 4.4.15. From Lemma 4.4.10 we know that all eigenvalues of $\tilde{H}$ are of the form $\lambda_j(\mu_l, \gamma\mu_l)$, for $\mu_l = 2\cos(2l\pi/n)$. We divide these eigenvalues into two categories: those with $\mu_l \neq 0$ or those with $\mu_l = 0$. For those with $\mu_l \neq 0$, they are all bounded away from 0 by at least $\Omega(1/(mn^2))$ as a result of Lemma 4.4.13. For those with $\mu_l = 0$, they are either 0 or bounded away from 0 by at least $\Omega(1/m)$ as a result of Lemma 4.4.14. Combining these two cases we have shown that all non-zero eigenvalues are bounded away from 0 by at least $\Omega(1/(mn^2))$. This proves (i).

For $\lambda_j(\mu_l, \gamma\mu_l) = 0$, we need $\mu_l = 0$ by Corollary 4.4.12. There are two values for l that achieve this: $l = n/4$ and $l = 3n/4$. From Example 4.4.8, the corresponding 0-eigenvectors of D_0 are $|\Phi^{\text{even}}\rangle$ and $|\Phi^{\text{odd}}\rangle$ given in Eq. (4.63). It is easy to check that $\langle\Phi^{\text{even}}|\Phi^{\text{odd}}\rangle = 0$ and these two eigenvectors are both normalized. For $\lambda_j(0,0) = \lambda_j(\mu_l, \gamma\mu_l)$ with these two values of l, we know from Lemma 4.4.14 that (again observe $H_1(0,0) = D_1$) there is only one j that satisfies $\lambda_j(0,0) = 0$, corresponding to the eigenvector $|\Psi\rangle$ given in that lemma. Therefore Lemma 4.4.10 tells us that the 0-eigenspace of $\tilde{H}$ is 2-dimensional and spanned by $|\Psi\rangle\,|\Phi^{\text{even}}\rangle$ and $|\Psi\rangle\,|\Phi^{\text{odd}}\rangle$, and these two vectors are normalized and orthogonal to each other. Therefore we have (ii).

[1] Here we use the notation that $|\mathfrak{b}_j, \mathfrak{a}_i\rangle = |\mathfrak{b}_j\rangle\,|\mathfrak{a}_i\rangle$.

For (iii), note that

$$\langle \chi^{\text{even}} | \mathfrak{b}_j, \mathfrak{a}_i \rangle = \langle \Psi | \mathfrak{b}_j \rangle \langle \Phi^{\text{even}} | \mathfrak{a}_i \rangle = \frac{1}{\sqrt{n/2}} \Psi_j^* \cos(i\pi/2),$$
$$\langle \chi^{\text{odd}} | \mathfrak{b}_j, \mathfrak{a}_i \rangle = \langle \Psi | \mathfrak{b}_j \rangle \langle \Phi^{\text{odd}} | \mathfrak{a}_i \rangle = \frac{1}{\sqrt{n/2}} \Psi_j^* \sin(i\pi/2).$$

When i is even, $|\cos(i\pi/2)| = 1$ and $\sin(i\pi/2) = 0$. Therefore $|\langle \chi^{\text{even}} | \mathfrak{b}_j, \mathfrak{a}_i \rangle| = |\Psi_j|/\sqrt{n/2}$ while $\langle \chi^{\text{odd}} | \mathfrak{b}_j, \mathfrak{a}_i \rangle = 0$. Because $|\Psi\rangle$ is normalized, we have

$$1 = \sum_{j=1}^{m} |\Psi_j|^2 = |\Psi_1|^2 \left(1 + \frac{d-2}{d-1}\frac{m-1}{2}\right).$$

Therefore $|\Psi_1| = \Omega(1/\sqrt{m})$. Consequently $|\langle \chi^{\text{even}} | \mathfrak{b}_j, \mathfrak{a}_i \rangle| = \Omega(1/\sqrt{mn})$. When i is odd, the corresponding statements can be proved in the same way. □

This theorem implies the following about the eigenvalues and eigenvectors of the effective Hamiltonian H:

Corollary 4.4.16. *For the effective Hamiltonian defined in Definition 4.4.10, the following statements are true*

(i) The non-zero eigenvalues of H are bounded away from zero by $\Omega(1/(mn^2))$.

(ii) The 0-eigenspace of H is 2-dimensional, and is spanned by an orthonormal basis

$$|\eta^{\text{even}}\rangle = V_{\mathcal{S}} |\chi^{\text{even}}\rangle, \quad |\eta^{\text{odd}}\rangle = V_{\mathcal{S}} |\chi^{\text{odd}}\rangle,$$

where $|\chi^{\text{even}}\rangle$ and $|\chi^{\text{odd}}\rangle$ are from Theorem 4.4.15 (ii), and $V_{\mathcal{S}}$ is the isometry defined in Eq. (4.46)*.*

(iii) The two quantum states $|\chi^{\text{even}}\rangle$ and $|\eta^{\text{odd}}\rangle$ satisfy the following conditions: for all even $1 \le i \le n$, we have $|\langle \eta^{\text{even}} | S_{i,1} \rangle| = \Omega(1/\sqrt{mn})$, $|\langle \eta^{\text{odd}} | S_{i,1} \rangle| = 0$. For all odd $1 \le i \le n$, we have $|\langle \eta^{\text{odd}} | S_{i,1} \rangle| = \Omega(1/\sqrt{mn})$, $|\langle \eta^{\text{even}} | S_{i,1} \rangle| = 0$.

Proof. Because by Definition 4.4.11, $\tilde{H} = V_{\mathcal{S}}^{-1} H V_{\mathcal{S}}$, H and $\tilde{H}$ share the same spectrum. Therefore (i) is a direct consequence of Theorem 4.4.15 (i), and this fact also implies that 0 is an eigenvalue of H with two-fold degeneracy. Since $|\chi^{\text{even}}\rangle$ and $|\chi^{\text{odd}}\rangle$ are eigenvectors of $\tilde{H}$, $|\eta^{\text{even}}\rangle$ and $|\eta^{\text{odd}}\rangle$ must be eigenvectors corresponding to the same eigenvalue, i.e., 0. Because $|\chi^{\text{even}}\rangle$ and $|\chi^{\text{odd}}\rangle$ form an orthonormal basis, $|\eta^{\text{even}}\rangle$ and $|\eta^{\text{odd}}\rangle$ must also form

an orthonormal basis since $V_{\mathcal{S}}$ is an isometry and thus preserves the inner product. We therefore have (ii). For (iii), we only need to note the fact that because $|S_{i,1}\rangle = V_{\mathcal{S}}\,|\mathfrak{b}_1, \mathfrak{a}_i\rangle$, we have $\langle \eta^{\text{even}}|S_{i,1}\rangle = \langle \chi^{\text{even}}|\mathfrak{b}_1, \mathfrak{a}_i\rangle$ and $\langle \eta^{\text{odd}}|S_{i,1}\rangle = \langle \chi^{\text{odd}}|\mathfrak{b}_1, \mathfrak{a}_i\rangle$ for all i. □

4.4.6 The algorithm

In this section, we provide a quantum algorithm to find an s-t path in the regular sunflower graph $\mathcal{G}_S$ defined in Definition 4.4.5 and show that this quantum algorithm requires only polynomial queries to the adjacency list oracle in Definition 2.1.3 and the indicator function in Definition 4.4.7. The algorithm we will present is reminiscent of the two-measurement algorithm in [CDF$^+$02]. Here, we first present the high-level idea.

First, we assume that the input of the algorithm is a unitary U_A, which is a block encoding of the adjacency matrix A of the regular sunflower graph $\mathcal{G}_S$. This assumption is without loss of generality because, given the adjacency list access of the graph, we can implement a unitary U_A in polynomial time as indicated by Lemma 2.1.2 in Section 2.1. The vertex t could also be given as input but it is not necessary to know it in advance for the algorithm. The vertex t could also be given as input but it is not necessary to know it in advance for the algorithm.

The algorithm works as follows: Starting from the initial state $|s\rangle$ we apply a polynomial function of the adjacency matrix A that has the effect of filtering out all eigenvectors corresponding to non-zero eigenvalues as indicated by Theorem 2.4.2 in Section 2.4. Because this is a non-unitary operation, it only succeeds with probability approximately the overlap between $|s\rangle$ and the 0-eigenspace of the effective Hamiltonian H (defined in Definition 4.4.10). Upon successful preparation of the projected state, we then measure it using the computational basis. This returns the bit string representing the vertex $S_{i,1}$, that is, the root of the tree $\mathcal{T}_i$, for odd i, each with probability at least $\Omega(1/\operatorname{poly}(m,n))$. Therefore, repeating this process gives samples that cover all the vertices $S_{i,1}$ for odd i along the target s-t path.[2] To fill in the remaining even-numbered vertices, we query all neighbors of the vertices in the previous step. t will then be included among these vertices with large probability, which we identify through querying f_t $\operatorname{poly}(m,n)$ times. Then a Breadth First Search of this graph gives a path from s to t.

Theorem 4.4.17. *Let $\mathcal{G}_S$ be the regular sunflower graph as defined in Definition 4.4.5 with $s = S_{1,1}$ known, m odd, and n an integer multiple of 4. Let $t = S_{n/2+1,1}$. Then with probability at least $2/3$, Algorithm 5 finds an s-t path in polynomial time.*

[2]Strictly speaking $S_{i,1}$ is a supervertex defined in Definition 4.4.8 rather than a vertex. However, given that $S_{i,1}$ contains only one vertex, we will use $S_{i,1}$ to refer to the vertex contained in it.

Algorithm 5 Finding an s-t path in the enlarged regular sunflower graph $\mathcal{G}_S$

Input: The adjacency list oracle access of a regular sunflower graph $\mathcal{G}_S = (\mathcal{V}, \mathcal{E})$, $s \in \mathcal{V}$ and an indicator function f_t (Definition 4.4.7).

Output: The set of vertices of an s-t path.

1: Construct circuit unitary $\mathcal{V}_{\text{circ}}$ acting on registers α, β_1, β_2 using Theorem 2.4.2 to get

$$\mathcal{V}_{\text{circ}} |0\rangle_{\alpha\beta_1\beta_2} = |0\rangle_{\alpha\beta_1} \Pi_0 |s\rangle_{\beta_2} + |\perp\rangle ,$$

where $(\langle 0|_{\alpha\beta_1} \otimes I_{\beta_2}) |\perp\rangle = 0$.

2: Measure the ancilla qubits, if the measurement result is 0, we obtain one copy of the quantum state

$$\frac{\Pi_0 |s\rangle}{\|\Pi_0 |s\rangle\|} = |\eta^{\text{odd}}\rangle .$$

Repeat until we have N_s copies of the quantum states $|\Psi^{H,\text{even}}\rangle$.

3: Let $\mathcal{M} = \emptyset$. Measure all the N_s copies of the quantum states $|\Psi^{H,\text{even}}\rangle$ in the computational basis to obtain outcomes $u \in \mathcal{V}$, and add them to $\mathcal{M}$. Using the adjacency list oracle to find all the neighbors of vertices in $\mathcal{M}$, and add them to $\mathcal{M}$. Construct a subgraph $\mathcal{G}_{\text{samp}}$ that contains all the vertices in $\mathcal{M}$.

4: Search through the subgraph $\mathcal{G}_{\text{samp}}$ using Breadth First Search for an s-t path and output the path if it is found.

Proof. In the first step of Algorithm 5, we first construct a block encoding U_A using Lemma 2.1.2 in Section 2.1, which will be a $(d, \text{poly}(m,n), 0)$-block-encoding of A. Next, the goal is to apply Theorem 2.4.2. In our setting, A is the adjacency matrix defined in Definition 4.4.6. $\mathcal{S}$ is the symmetric subspace defined in Eq. (4.44). The restriction of A to $\mathcal{S}$ is defined to be the effective Hamiltonian in Definition 4.4.10. The spectral gap of H around the eigenvalue 0 is guaranteed by Corollary 4.4.16 (i) to be at least $\Omega(1/(mn^2))$. Therefore, the first part of the conditions are all satisfied, with $\Delta = \Omega(1/(mn^2))$. Apply Theorem 2.4.2 with $\alpha = d$, $\varepsilon_{\text{BE}} = \Theta(1/(m^2n^2))$, and $\Delta = \Omega(1/(mn^2))$ to obtain the unitary $\mathcal{V}_{\text{circ}}$, which is an $(1, \text{poly}(m,n), \epsilon_{\text{BE}})$-block-encoding of Π_0 using

$$O((\alpha/\Delta)\log(1/\epsilon_{\text{BE}})) = O(mn^2 \log(nm)) = O(mn^2 \log(mn)), \tag{4.64}$$

applications of (controlled-) U_A and $U_A^\dagger$ and

$$\mathcal{V}_{\text{circ}} |0\rangle_{\alpha\beta_1\beta_2} = |0\rangle_{\alpha\beta_1} \Pi_0 |s\rangle_{\beta_2} + |\perp\rangle .$$

In the second step of Algorithm 5, because the 0-eigenspace of H is spanned by the

orthogonal vectors $|\eta^{\text{even}}\rangle$ and $|\eta^{\text{odd}}\rangle$ from Corollary 4.4.16 (ii), we have

$$\Pi_0 = |\eta^{\text{even}}\rangle \langle\eta^{\text{even}}| + |\eta^{\text{odd}}\rangle \langle\eta^{\text{odd}}| .$$

Therefore

$$\Pi_0 |s\rangle = |\eta^{\text{even}}\rangle \langle\eta^{\text{even}}|s\rangle + |\eta^{\text{odd}}\rangle \langle\eta^{\text{odd}}|s\rangle = |\eta^{\text{odd}}\rangle \langle\eta^{\text{odd}}|s\rangle ,$$

where we have used the fact that $|s\rangle = |S_{1,1}\rangle$ and therefore have no overlap with $|\eta^{\text{even}}\rangle$ according to Corollary 4.4.16 (iii). Consequently

$$\frac{\Pi_0 |s\rangle}{\|\Pi_0 |s\rangle \|} = |\eta^{\text{odd}}\rangle . \tag{4.65}$$

The probability of getting the quantum state $|\eta^{\text{odd}}\rangle$ by measuring all 0 state in the first register is therefore

$$\|\Pi_0 |s\rangle \|^2 = |\langle\eta^{\text{odd}}|S_{1,1}\rangle|^2 = \Omega(1/mn), \tag{4.66}$$

where we have used Corollary 4.4.16 (iii) for $i = 1$.

The number of queries to obtain one copy of $|\eta^{\text{odd}}\rangle$ with probability at least 2/3 is therefore

$$O(mn) \times O(mn^2 \log(1/\epsilon_{\text{BE}})) = O(m^2 n^3 \log(1/\epsilon_{\text{BE}})), \tag{4.67}$$

where the query complexity for one application of the projection operator is given by (4.64). Since the projection operator is implemented with block-encoding error $\epsilon_{\text{BE}} = \Theta(1/(m^2 n^2)$, the quantum state we get in the end will also differ from $|\eta^{\text{odd}}\rangle$ by

$$O(mn) \times \epsilon_{\text{BE}} = O(mn\epsilon_{\text{BE}}) = O(1/mn) \tag{4.68}$$

in trace distance. Repeat until we have $N_s = \Theta(mn \log(n))$ copies of the quantum states.

In the third step of Algorithm 5, we measure all N_s copies of the approximate state $|\Psi^{H,\text{even}}\rangle$ on the computational basis to get the name of N_s vertices of the graph. From these vertices, we first find all their neighbors in the graph $\mathcal{G}$. There are at most dN_s of them, and finding all the neighbors takes at most $dN_s = O(mn \log(n))$ queries to the oracle $O_{\mathcal{G}}$. From these vertices and their neighbors, we generate a subgraph $\mathcal{G}_{\text{samp}}$ of the original graph $\mathcal{G}$ with them as vertices. It turns out that the constructed subgraph will contain all vertices of the target s-t path with probability at least 2/3. The reason is as follows.

For a perfect copy of $|\eta^{\text{odd}}\rangle$, the probability of the measurement outcome being

an even vertex v along the target s-t path is at least $\Omega(1/(mn))$. Note that the trace distance between a perfect copy of $|\eta^{\text{odd}}\rangle$ and an approximate copy of $|\eta^{\text{odd}}\rangle$ is $O(1/(mn))$. Therefore, measure an approximate state of $|\eta^{\text{odd}}\rangle$ will return an even vertex vertex v along the target s-t path with probability $\mathfrak{p} = \Omega(1/(mn))$. Then the probability that an even vertex v does not appear in the samples is $(1-\mathfrak{p})^{N_s}$, where $\mathfrak{p} = \Omega(1/(mn))$. By the union bound, the probability of at least one of the even vertices not showing up among the samples is at most

$$(n/2)(1-\mathfrak{p})^{N_s}. \tag{4.69}$$

Therefore, the choice of $N_s = \Theta(mn\log(n))$ ensures that we obtain all even vertices along the target s-t path with probability at least $2/3$.

In the final step of Algorithm 5, the subgraph $\mathcal{G}_{\text{samp}}$ contains $dN_s = O(mn\log(n))$ number of vertices. Therefore, the classical algorithm can find the s-t path in $O(mn\log(n))$ time. Therefore with probability at least $2/3$ we will be able to find an s-t path in the original graph $\mathcal{G}$. Since each step takes at most the polynomial number of queries of U_A and each query of the unitary U_A takes polynomial time by Lemma 2.1.2, therefore Algorithm 5 takes polynomial time finds an s-t path. □

4.5 Classical Lower Bounds

In this section, we show that there are no classical algorithms for finding an s-t path in the welded tree path graph $\mathcal{G}_P$, the welded tree circuit graph $\mathcal{G}_C$, and the regular sunflower graph $\mathcal{G}_S$, in subexponential time. To prove these lower bounds for finding an s-t path, that is, a set of edges or vertices, in these graphs, we follow the lower bound proof in [CCD$^+$03] to show exponential quantum-classical separations to find a marked vertex in the welded tree graph.

The high-level idea is to first show that classical algorithms can only explore graphs by generating a connected subgraph of them. Then one can show that before encountering a cycle in these graphs, the behavior of the classical algorithm can be described by a *random embedding* π of a rooted $(d-1)$-ary tree $\mathcal{T}$ into these graph as defined in [CCD$^+$03], which we restate with slight modification here:

Definition 4.5.1 (Random embedding)**.** *Let $\mathcal{T}$ be a $(d-1)$-ary tree. A mapping π from $\mathcal{T}$ to G is a random embedding of $\mathcal{T}$ into G if it satisfies the following:*

1. *The root of $\mathcal{T}$ is mapped to s.*

2. *Let $v_1, \ldots, v_{d-1}$ be the children of the root, and $a_1, \ldots, a_{d-1}$ be the neighbors of s. Then $(\pi(v_1), \ldots, \pi(v_{d-1}))$ is an unbiased random permutation of $(a_1, \ldots, a_{d-1})$.*

3. *Let v be an internal vertex in $\mathcal{T}$ other than the root, let $v_1, \ldots, v_{d-1}$ be its children and u its parent, and let $a_1, \ldots, a_{d-1}$ be the neighbors of $\pi(v)$ except for $\pi(u)$. Then $(\pi(v_1), \ldots, \pi(v_{d-1}))$ is an unbiased random permutation of $(a_1, \ldots, a_{d-1})$.*

We also say that the embedding π is proper *if it is injective, and say it* exits *if $t \in \pi(\mathcal{T})$.*

To show that a classical algorithm takes exponential time, it is therefore sufficient to show that a random embedding has to be exponentially sized in order to find a cycle or exit t.

4.5.1 Classical lower bound for the pathfinding problem in $\mathcal{G}_P$

In this section, we show that no classical randomized algorithm $\mathcal{R}$ can solve the pathfinding problem in the welded tree path graph $\mathcal{G}_P$ in subexponential time. To prove the lower bound, we analyze the difficulty of $\mathcal{R}$ winning a game in which $\mathcal{R}$ needs to output the vertices of an s-t path in the graph $\mathcal{G}_P$. Note that $\mathcal{R}$ winning the game is equivalent to solving the pathfinding problem. Since every s-t path has to go through a vertex of degree at least 4 or pass through the vertex $p_{n/2}$ as indicated in Figure 4.1, we can define a new game that is easier to win, where $\mathcal{R}$ needs to output the name of a vertex of degree at least 4 or the name of the vertex $p_{n/2}$, or a cycle. Finally, using the classical lower bound result of the welded tree problem, we show that $\mathcal{R}$ cannot win this new game in subexponential time.

Game A The adjacency list oracle O contains a random set of names for the vertices of $\mathcal{G}_P$ such that each vertex has a distinct $2n$- bit string as its name and the starting vertex s has the name 0^{2n} and the ending vertex t has the name 1^{2n}. At each step, $\mathcal{R}$ sends a $2n$-bit string to O, and the oracle O returns the names of the neighbors of that vertex if the given vertex name is valid. $\mathcal{R}$ wins if it outputs the names of the vertices of an s-t path.

The total number of vertices in the graph G is $n2^{n+2} + \frac{n(n+1)}{2}$ and the total number of potential names is 2^{2n}. If $\mathcal{R}$ makes at most $2^{n/6}$ queries, the probability of querying the name of a vertex that has not been previously returned by the oracle O is at most $2^{n/6}(n2^{n+2} + \frac{n(n+1)}{2})/2^{2n} = O(2^{-n/6})$. Therefore, we can restrict $\mathcal{R}$ to traverse two connected subgraphs starting with vertices s and t, respectively.

To obtain an upper bound on the success probability of Game A, we compare it with a simpler game by relaxing the condition needed for $\mathcal{R}$ to win. Every s-t path contains

at least one vertex of degree at least 4 passing through some t_i or the vertex $p_{n/2}$. Thus, the following Game B is easier to win and it suffices to show that $\mathcal{R}$ cannot win this easier game in subexponential time.

Game B Let Game B be the same as Game A, except that $\mathcal{R}$ wins if it outputs the name of a vertex of degree at least 4 or the name of the vertex $p_{n/2}$, or if the vertices visited by $\mathcal{R}$ contain a cycle.

Following [CCD+03], the additional cycle condition that allows $\mathcal{R}$ to win in Game B allows us to analyze the success probability of R winning. This analysis involves determining whether a random embedding of a random rooted binary tree into the random graph G contains a cycle, a vertex of degree at least 4, or the vertex $p_{n/2}$.

Lemma 4.5.1. *If $\mathcal{R}$ uses $2^{n/6}$ queries to the oracle O, then its probability of winning Game A is at most $2(n+1)\cdot 4\cdot 2^{-n/6}$.*

Proof. To obtain an upper bound of the winning probability of $\mathcal{R}$ for Game A, it suffices to show that the probability of $\mathcal{R}$ winning Game B is at most $2(n+1)\cdot 4\cdot 2^{-n/6}$.

Let T be a random rooted binary tree with $2^{n/6}$ vertices and $\pi(T)$ be the image in the graph G under the random embedding π. Given the name of the starting vertex s, similar to [CCD+03], the probability of $\mathcal{R}$ winning Game B can be expressed as the probability that $\pi(T)$ contains a cycle, or a vertex of degree at least 4, or the vertex $p_{n/2}$.

First, $\mathcal{R}$ has to enter a welded tree graph to find a vertex with a degree of at least 4 or to find a cycle. This is true because, as indicated in Figure 4.1, the vertices that have a degree at least 4 are the roots t_i, and the path P_n and the n welded trees form a tree structure. Lemma 4.2.3 states that, in a welded tree graph, any classical algorithm that makes at most $2^{n/6}$ queries to the oracle and finds the other root or a cycle with probability at most $4\cdot 2^{-n/6}$. There are n welded trees and n vertices of degree at least 4. Therefore, by the union bound, the probability of $\mathcal{R}$ finding a cycle or a vertex with a degree at least 4 using $2^{n/6}$ queries to the oracle is at most $n\cdot 4\cdot 2^{-n/6}$.

Second, the probability that $\mathcal{R}$ finds the name of the vertex $p_{n/2}$ can be expressed as the probability that a random embedding π of a random rooted binary tree T with root s into $\mathcal{G}_P$ contains the vertex $p_{n/2}$. Consider a path in T from the root to a leaf. To reach the vertex $p_{n/2}$, π must follow the path P_n $\frac{n}{2}$ times, which has probability $2^{-n/2}$. Since there are at most $2^{n/6}$ tries on each path of T and there are at most $2^{n/6}$ paths. The probability of finding the name of the vertex $p_{n/2}$ is at most $2^{-n/6}$. Therefore, given the name of the starting vertex s, the probability of $\mathcal{R}$ finding the vertex $p_{n/2}$, a cycle, or a vertex of degree at least 4 is at most $(n+1)\cdot 4\cdot 2^{-n/6}$

The same result also holds if the only given name is the ending vertex t. Hence, given the names of s and t, the probability of $\mathcal{R}$ uses $2^{n/6}$ queries to the oracle to find a cycle, the vertex $p_{n/2}$, or a vertex with a degree at least 4 is at most $2(n+1) \cdot 4 \cdot 2^{-n/6}$. Therefore, $\mathcal{R}$ cannot win Game A in subexponential time.

□

4.5.2 Classical lower bound for the pathfinding problem in $\mathcal{G}_C$

In this section, we show that our Algorithm 4 actually provides an exponential speedup compared to any classical algorithm under the assumption that the following welded tree pathfinding problem is classically hard. To simplify the proof of our lower bound for the pathfinding problem Problem 4.3.17, we use the following assumption and the known classical lower bound of the welded tree problem.

Problem 4.5.2 (The welded tree pathfinding problem)**.** *Given an adjacency list oracle O_G to the welded tree graph G and the names of the starting vertex s and the ending vertex t, output the names of the vertices of an s-t path.*

It is folklore that the welded tree pathfinding problem is classically difficult, however, there is no formal statement as far as we are aware.

Assumption 4.5.3. *There exist constants $c_1 > 0$ and $c_2 \in (0, 2)$ such that any classical algorithm that makes at most $2^{n/6}$ number of queries to O_G to the welded tree graph G solves Problem 4.5.2 with probability at most $c_1 \cdot 2^{-c_2 n}$.*

Lemma 4.5.4 (Theorem 9 in [CCD$^+$03])**.** *For the welded tree problem Problem 4.3.15, any classical algorithm that makes at most $2^{n/6}$ queries to the oracle O_G finds the ending vertex or a cycle with probability at most $4 \cdot 2^{-n/6}$.*

To prove the lower bound, we analyze the difficulty of any classical algorithm $\mathcal{A}$ winning a simpler game:

Game A Let n be odd in the welded circuit graph $\mathcal{G}_C$. Let Game A be the game where any classical algorithm $\mathcal{A}$ wins if it outputs the name of one of the vertex $v_{p,(n+1)/2,1}$, or if the vertices visited by $\mathcal{A}$ contain a cycle. Following [CCD$^+$03], the additional cycle condition that allows $\mathcal{A}$ to win in Game A allows us to analyze the success probability of $\mathcal{A}$ winning. This analysis involves determining whether a random embedding of a random rooted binary tree into the random graph $\mathcal{G}_C$contains a cycle or the vertex $v_{p,(n+1)/2,1}$.

Theorem 4.5.5. *Let G be the graph defined in Section 4.3.7. Let c_1, c_2 be the constants from Assumption 4.5.3 and assume that this assumption is true. Then any classical algorithm that makes at most $2^{n/6}$ queries to O_G solves Problem 4.3.17 with probability at most $(5 + c_1) \cdot 2^{-\min\{c_2,1/6\}n}$.*

Proof. Let T be a random rooted binary tree with $2^{n/6}$ vertices and $\pi(T)$ be the image in the graph G under the random embedding π. Given the name of the starting vertex s, similar to [CCD+03], the probability of $\mathcal{A}$ winning Game A can be expressed as the probability that $\pi(T)$ contains a cycle or the vertex $v_{p,(n+1)/2,1}$.

First, $\mathcal{A}$ has to enter a welded tree subgraph to find a cycle, as seen in Fig. 4.14. There are two possibilities to get a cycle in a welded tree subgraph. One is to find a cycle that contains only one root in one of the welded tree subgraphs. In this case, Lemma 4.2.3 states that, in one of the welded tree subgraphs, starting from one root, any classical algorithm that makes at most $2^{n/6}$ queries to the oracle and finds the other root or a cycle with probability at most $4 \cdot 2^{-n/6}$. The other is to find a cycle that contains two roots of a welded tree subgraph. By Assumption 4.5.3, any classical algorithm that makes at most $2^{n/6}$ queries to the oracle and finds such a cycle with probability at most $c_1 \cdot 2^{-c_2 n}$.

We can now assume that $\mathcal{A}$ will not encounter any cycle. Conditioned on this fact, the probability that $\mathcal{A}$ finds the name of the vertex $v_{p,(n+1)/2,1}$ can be expressed as the probability that $\pi(T)$ contains the vertex $v_{p,(n+1)/2,1}$, for which π must follow the corresponding path $2n$ times, which has probability 2^{-2n}. Since there are at most $2^{n/6}$ tries on each path of T and there are at most $2^{n/6}$ paths, the probability of finding the name of the vertex $v_{p,(n+1)/2,1}$ is by the union bound at most $2^{n/3}2^{-2n} \leq 2^{-5n/3}$. We have the same result if the given name is t. Therefore, given the name of the starting vertex s and t, the probability of $\mathcal{A}$ finding the vertex $v_{p,(n+1)/2,1}$ is $2 \cdot 2^{n/3}2^{-2n} \leq 2^{-5n/3}$.

By combining the two cases with the union bound, we find that the probability of $\mathcal{A}$ winning Game A is at most $2^{-5n/3} + (4 + c_1) \cdot 2^{-\min\{c_2,1/6\}n} \leq (5 + c_1) \cdot 2^{-\min\{c_2,1/6\}n}$. Since solving Problem 4.3.17 automatically wins Game A, the theorem follows. □

4.5.3 Classical lower bound for the pathfinding problem in $\mathcal{G}_S$

In this section, we show that no efficient classical algorithm can find an s-t path in the regular sunflower graph $\mathcal{G}_S = (\mathcal{V}, \mathcal{E})$ defined in Definition 4.4.5. Without loss of generality, we assume that the classical algorithm can only traverse a connected subgraph starting with vertex s since the name space of the vertices of the graph could be much

larger, as indicated in [CCD+03].

For this, we have the following lemma:

Lemma 4.5.6. *Let $\mathcal{T}$ be a rooted $d-1$-ary tree with q vertices, with $q = (d-1)^{cn}$ and $c < 1/4$. Assuming $m = n+1$, then a random embedding π of this tree into $\mathcal{G}_S$ as defined in Definition 4.5.1 is improper or exits with probability at most $O((d-1)^{-(1/2-2c)n})$.*

Proof. First, we note that in order for π to exit, there must exist a path in $\mathcal{T}$ leading from the root to leaves that, when mapped to $\mathcal{G}_S$ through π, moves right in $\mathcal{G}_S$ along the bottom path for $n-1$ consecutive steps, or moves downward from the top part of the graph to the bottom for $m-1$ consecutive steps. Both involve probability at most $O((d-1)^{-n})$. Because there are $\binom{q}{2} = O(q^2)$ paths in $\mathcal{T}$ the probability of it exiting is therefore $O((d-1)^{-(1-2c)n})$.

Next, we consider the probability of the embedding being improper, i.e., there existing vertices $a \neq b$ of $\mathcal{T}$ such that $\pi(a) = \pi(b)$, which also means the presence of a cycle in the subgraph $\pi(\mathcal{T})$ of $\mathcal{G}_S$. Note that with the exception of the cycle at the bottom of the graph consisting of $S_{1,1}, S_{2,1}, \ldots, S_{n,1}$, a cycle must involve the topmost level, the only part of the graph aside from the bottom cycle that does not consist of trees. We can exclude the bottom cycle from our consideration since finding it already involves finding the exit t, which we have established is unlikely in a random embedding. Now we will focus on a cycle that passes through the topmost level. We will show that in such a cycle, it is very unlikely for lower levels $1, 2 \ldots, n/2$ of $\mathcal{G}_S$ to be involved in the cycle in $\pi(\mathcal{T})$, because that would require a path in $\mathcal{T}$ that, if we take the direction away from the root, when mapped through π, travels downwards consecutively for distance at least $n/2+1$. Given the tree structure in the levels from $n/2+1$ to n, for each path, this happens with probability most $(d-1)^{-n/2}$. Because there are $\binom{q}{2}$ paths in $\mathcal{T}$, the probability of having a cycle in $\pi(\mathcal{T})$ that involves a vertex on levels $1, 2 \ldots, n/2$ is therefore at most

$$\binom{q}{2}(d-1)^{-n/2} = O((d-1)^{-(1/2-2c)n}). \tag{4.70}$$

Hereafter we will only consider the scenario where the cycle in $\pi(\mathcal{T})$ does not involve a vertex on levels $1, 2 \ldots, n/2$.

We consider each of the $\binom{q}{2} = O(q^2)$ pairs of vertices a and b in $\mathcal{T}$ separately. We will show that it is exponentially unlikely for $\pi(a) = \pi(b)$. We will introduce some notations: let P be the path linking a and b, and let c be the vertex on this path that is closest to the root. Let P_1 be the path leading from c to a, and P_2 the path leading from c to b.

We then divide all vertices in the supervertices $S_{i,n/2+1}, S_{i,n/2+2}, \ldots, S_{i,n+1}$, i.e., all

vertices in $\mathcal{T}_i$ that are at least $n/2+1$-distance from the root of $\mathcal{T}_i$ (defined in Definition 4.4.5 and illustrated in Figure 4.3), into $(d-2)(d-1)^{n/2}$ complete d-ary subtrees of height $n/2$, for $i = 1, 2, \ldots, n$.

Since if $\pi(a) = \pi(b)$ then $\pi(a)$ will be involved in a cycle in $\mathcal{T}$, this tells us that we only need to consider the situation where $\pi(a)$ is among these complete d-ary subtrees. Otherwise, we will have the situation of a cycle going to lower levels, which we have established is exponentially unlikely. Then we consider two possible scenarios: c being an ancestor of a and b, or $c = a$ or $c = b$. We know that one of these two possibilities must be true because c is at least as close to the root as a and b.

In the first scenario where c is an ancestor of a and b, the path $\pi(P_1)$ must visit a sequence of these subtrees, which we denote by $S_{k_1}, S_{k_2}, \ldots S_{k_u}$, and similarly $\pi(P_2)$ visits $S_{l_1}, S_{l_2}, \ldots S_{l_v}$. For any choice of $S_{k_1}, S_{k_2}, \ldots S_{k_{u-1}}$, suppose that $S_{k_{u-1}}$ is in the tree $\mathcal{T}_i$ (as defined in Definition 4.4.5), the next subtree S_{k_u} will be uniformly randomly chosen among the subtrees that are within the adjacent trees $\mathcal{T}_{i-1 \pmod n}$ and $\mathcal{T}_{i+1 \pmod n}$ due to the random connectivity between the trees $\{\mathcal{T}_{i'}\}$ and the random embedding. There are $2(d-2)(d-1)^{n/2}$ such subtrees, and therefore the probability of S_{k_u} being any one of them is $1/(2(d-2)(d-1)^{n/2})$. The same is also true for S_{l_v}. Because of the Markovian nature of the random embedding, S_{k_u} and S_{l_v} are independent of each other, and therefore the probability of $S_{k_u} = S_{l_v}$ is at most

$$2(d-2)(d-1)^{n/2} \times \frac{1}{(2(d-2)(d-1)^{n/2})^2} = \frac{1}{2(d-2)(d-1)^{n/2}}. \tag{4.71}$$

In the second scenario, without loss of generality, we assume $c = a$. Then we denote the subtrees visited on the path from c to b as $S_{k_1}, S_{k_2}, \ldots S_{k_u}$. Conditional on $S_{k_1}, S_{k_2}, \ldots S_{k_{u-1}}$, again because the random embedding is Markovian, S_{k_u} can be any one of the subtrees with probability at most $1/(2(d-2)(d-1)^{n/2})$. S_{k_1} is one of these trees, and therefore Eq. (4.71) is also an upper bound for the probability of $S_{k_u} = S_{k_1}$.

From the upper bound of probability Eq. (4.71), we can see that for any pair of a and b the probability of $\pi(a) = \pi(b)$ is at most $O((d-1)^{-n/2})$, unless there is a cycle going through levels $1, 2, \ldots, n/2$. There are $\binom{q}{2}$ pairs of a and b, and therefore the probability of π being an improper embedding is at most

$$\binom{q}{2} \times O((d-1)^{-n/2}) = O((d-1)^{-(1/2-2c)n}), \tag{4.72}$$

excluding the scenario of a cycle going through lower levels. The probability of such a

cycle existing is upper bound by Eq. (4.70), and therefore the total probability of π being an improper embedding is at most $O((d-1)^{-(1/2-2c)n})$ accounting for all situations. $\square$

With the above reasoning, the classical lower bound is stated in the following theorem:

Theorem 4.5.7. *For the regular sunflower graph $\mathcal{G}_S$ defined in Definition 4.4.5, with $m = n + 1$, using $q = (d-1)^{cn}$ queries of the adjacency list oracle $O_{\mathcal{G}_S}$ where $c < 1/4$, the probability of a classical randomized algorithm finding a cycle or finding the vertex t is at most $O((d-1)^{-(1/2-2c)n})$.*

Chapter 5

Conclusion

Problem of Solving Polynomial Systems: The Boolean Macaulay linear system approach is an interesting framework to study giving insights to the limitations and capabilities of quantum computation. On the one hand, a lot of problems such as Factoring, Graph isomorphism, and Learning with binary errors can be put into this single framework. On the other hand, the QLS algorithm used for the (Boolean) Macaulay linear system is BQP-complete and the Factoring problem is known to be inside BQP by Shor's algorithm, therefore, if we can find an approach to get around the curse of the condition number of the Boolean Macaulay linear system derived from the Factoring problem, then we might be able to extend the result to other problems, such as Graph Isomorphism and Learning with binary errors, revealing new capabilities of quantum computation.

Our analytical lower bound on the condition number decreases when there are multiple solutions of the polynomial systems, but the polynomial systems used for cryptography usually have one or few solutions [CG17], so our result gives strong evidence that the QLS algorithm cannot be used for attacking cryptosystems via the Macaulay matrix approach. Also, we suspect that having many solutions will not make the QLS algorithm work substantially better. For example, consider adding l new field equations $y_i^2 - y_i = 0$, then the number of solutions of the new polynomial system will increase by a factor of 2^l, however, the length of the shortest vector stays the same – indeed one can see that the shortest vector is an affine combination of solutions where all the new variables are set to 0.[1]

Given an ill-conditioned QLSP, two main approaches have been proposed, one is

[1]To see this consider the coordinates corresponding to monomials does not including the new variables.

the truncated QLS algorithm while the other is the preconditioned QLS algorithm [HHL09]. Our lower bound on the tQLScn prohibits further speedup by the truncated QLS algorithm, however further investigation is needed regarding the possibility of using preconditioned QLS algorithms, such as parallel sparse approximate inverse preconditioner [CJS13], circulant preconditioner [SX18], fast inversion [TAWL20], or develop new preconditioned QLS algorithms for the Boolean Macaulay linear system. A promising feature of the Boolean Macaulay linear system is that it cannot be dequantized by known classical techniques [CGL$^+$20] since the Boolean Macaulay linear system has high or full column rank.

The introduced variant of the quantum coupon collector problem provides an example of how to extract the solution efficiently if the values in the solution vector of a QLSP are correlated in a nice pattern. Since applications of the QLS algorithm usually gain restricted access to the solution vector, a generalization of our extraction method, utilizing more general correlated patterns, could have interesting applications.

When the Hamming weight of the solution of a Boolean polynomial system is logarithmic in the number of variables, our lower bound does not rule out a superpolynomial speedup over exhaustive search. Finding a real application that exhibits such a superpolynomial speedup would be very interesting. However, for applications like polynomial systems over a finite field, the employed reductions usually increase the number of variables in the corresponding polynomial system significantly – likely compromising the potential speedup.

Pathfinding Problem: We summarize the result of exponential speedups with respect to the pathfinding problem and potential future directions as in Fig. 5.1

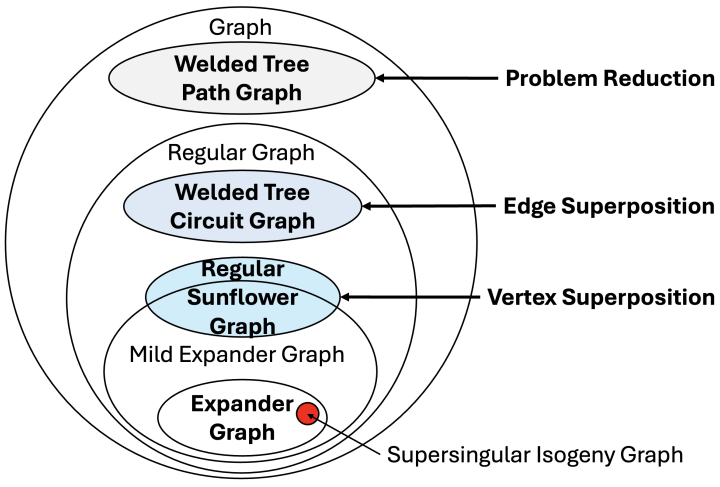

Figure 5.1. Exponential Speedups Landscape for Pathfinding Problems

In particular, for the edge superposition approach and vertex superposition approach, we have developed a new multidimensional electrical network framework and found a new connection between eigenspaces of the adjacency matrix and pathfinding problem within a graph, these new algorithmic tools could potentially have more applications.

Although our applications all consider adjacency list access to the underlying graph, the multidimensional electrical network can also find applications in the non-oracle setting. For example, in the case of finding an s-t path in exponentially sized isogeny graphs [CLG09], the adjacency list oracle can be instantiated by computing isogenies between elliptic curves. This problem is believed to be hard even for quantum algorithms, but our new multidimensional electrical network framework gives some new hope, as our works present an exponential speedup for this pathfinding problem on a regular graph.

The other potential application of the multidimensional electrical network framework is to find a killer application of quantum algorithms based on the QLS algorithm. Arguably, the most promising killer applications are believed to be based on quantum linear algebra with classical input and classical output. However, due to the recent dequantization results [Tan19, Tan21, CGL$^+$20] and caveats of the QLS algorithm [Aar15], in order for a quantum algorithm to show a superpolynomial speedup in this context, the rank of the matrix in the linear system needs to be high and the condition number of the matrix should be polylogarithmic with respect to the size of the matrix. Moreover, the only barrier to showing superpolynomial speedup to use the quantum algorithm in Chapter 3 for solving systems of polynomial equations is because the lower bound of the condition number of their associated matrix is exponentially large. It is unlikely that a standard preconditioning technique will overcome this barrier. The multidimensional electrical network framework allows us to generate the s-t alternative electrical flow state in polynomial time in our pathfinding example graph from Fig. 4.14, whereas the cost of generating the s-t electrical flow state is exponentially large. Since the alternative electrical flow state is a solution to a linear system of equations, as shown in Theorem 4.3.1 and Theorem 4.3.2, our framework may provide a new way to circumvent the condition number barrier for quantum algorithms to solve polynomial systems. By modifying the system of linear equations, like we change the incidence matrix B to the alternative incidence matrix $\mathcal{B}^{\mathsf{alt}}$, one could obtain an efficiently solvable linear system whose solution contains information about the solution to the original linear system.

By reformulating Alternative Kirchhoff's Law and Alternative Ohm's Law in terms of the alternative incidence matrix, our multidimensional electrical network framework could also have applications in the classical algorithm design for graph problems. Elec-

trical networks have a strong duality to both classical random walks, as well as quantum random walks. In this work, we study and recover the connection between (alternative) electrical networks and the generalized quantum walk paradigm of multidimensional quantum walks. It would be interesting to see if there is a similar classical analog of this multidimensional quantum walk by trying to map the (alternative) electrical network back to some classical Markovian process. In addition, Kirchhoff's Law and Ohm's Law were originally derived from real physical observations. It would be worthwhile to study the physical intuition behind our new Alternative Kirchhoff's Law and Alternative Ohm's Law.

Finding an s-t path in the welded-tree graph is one of the top open problems in the field of quantum query complexity [Aar21]. Our new algorithm approaches for generating quantum superposition states over edges or vertices cannot solve this problem, but may provide some helpful insights when considered from the perspective of recent work [CCG22]. Our new quantum algorithms achieve exponential quantum-classical separations and are not rooted as defined in [CCG22]. A rooted quantum algorithm needs to maintain all the paths from the starting vertex s for the vertices within a memory space. This means that our algorithm also has the surprising property that it returns to us the s-t path even though it does not remember the paths that it explores. This fact may imply that these non-rooted algorithms are more plausible than previously thought.

Another future direction to consider is extending our results to expander graphs with constant spectral gaps, and even to certain classes of supersingular isogeny graphs. Since the access oracle in this case can be efficiently instantiated by computing isogenies of the elliptic curves, such extension will result in one of the rare examples of exponential quantum speedup in a non-oracular setting [Aar22] aside from Shor's algorithm [Sho97].

Many big open problems in the field of quantum computing can be reduced to the task of generating a specific type of quantum state, such as the graph isomorphism problem [AT07], lattice-related problems [EH22], and the problem of computing the ground state of local Hamiltonians [GHLS15]. However, limited progress has been made on how to generate these quantum states and exhibit exponential speedup on these types of problems. Alternatively, an equally important direction is to design new quantum algorithms to generate certain types of quantum states and use those states to exhibit, hopefully superpolynomial, speedups for certain other problems. Hopefully, our new flow state generation techniques may also shed some light on those big open problems related to quantum state generation.

Appendix A

A.1 Simple proof of the unique solution case

Here we present a simple proof of the correctness of Algorithm 1 for Problem 3.2.2 when it has a unique solution. Let $a = (a_1, a_2, \dots a_n) \in \{0,1\}^n$ be the unique solution of a set of polynomials $\mathcal{F}$. Let $\hat{y} = [a_1, a_2, \dots, a_i a_j, \dots, \prod_{i=1}^n a_i]^\top$ be the 0/1 solution vector labeled by the multilinear monomials under the assignment a.

Next, we will show that the Boolean Macaulay linear system $M\vec{y} = \vec{b}$ has the unique solution $\hat{y}$ when $\mathcal{F}$ has the unique solution a. In this case, we have $\hat{y} = M^+\vec{b}$ because the matrix M has linearly independent columns. When $\mathcal{F}$ has more than one solution, the columns of the matrix M are not linearly independent and the solutions of $M\vec{y} = \vec{b}$ form a multidimensional affine subspace.

Lemma A.1.1. *[AFI+04, Theorem 2] If a set of polynomials $\mathcal{F} \subseteq \mathbb{C}[x_1, \dots, x_n]$ has a unique solution $a = (a_1, a_2, \dots, a_n)$, then the following two polynomial ideals coincide*

$$\langle \mathcal{F} \rangle = \langle x_1 - a_1, x_2 - a_2, \dots, x_n - a_n \rangle .$$

Theorem A.1.2. *Given a set of polynomials $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2 \subseteq \mathbb{C}[x_1, \dots, x_n]$, where $\mathcal{F}_1 = \{f_1, f_2, \dots, f_m\}$ and $\mathcal{F}_2 = \{x_1^2 - x_1, x_2^2 - x_2, \dots, x_n^2 - x_n\}$. Suppose $\mathcal{F}$ has a unique solution $a = (a_1, \dots, a_n)$, where $\mathcal{F}_2$ forces the root of the set of polynomials $\mathcal{F}_1$ to be Boolean. Let $\hat{y}$ be the multilinear monomial solution vector corresponding to the solution a, then the Boolean Macaulay linear system $M\vec{y} = \vec{b}$ of total degree n has the unique solution $\hat{y} = M^+\vec{b}$.*

Proof. First, we prove that for all the nontrivial multilinear monomials X^β, the polynomial $X^\beta - \prod_{i=1}^n a_i^{\beta_i} = \prod_{i=1}^n x_i^{\beta_i} - \prod_{i=1}^n a_i^{\beta_i}$ is in $\langle \mathcal{F} \rangle$, where $\beta \in \{0,1\}^n \backslash \{0^n\}$. The proof is by induction on the degree d. For the base case $d = 1$, by Lemma A.1.1, for all

$1 \leq k \leq n$, $x_k - a_k \in \langle \mathcal{F} \rangle$. That is, for each k, there exist p_i, q_j such that $x_k - a_k = \sum_{i=1}^{m} p_i f_i + \sum_{j=1}^{n} q_j(x_j^2 - x_j)$. Let $B_d = \{\text{all multilinear monomials with degree } d\}$, and suppose the claim is true for d, that is, for any $X_d^{\beta'} \in B_d$, $X_d^{\beta'} - \prod_{i=1}^{n} a_i^{\beta_i'} \in \langle \mathcal{F} \rangle$. For any $X_{d+1}^{\beta} \in B_{d+1}$, there exists some x_k and $X_d^{\beta'}$ such that $X_{d+1}^{\beta} = x_k \cdot X_d^{\beta'}$ and $\prod_{i=1}^{n} a_i^{\beta_i} = a_k \cdot \prod_{i=1}^{n} a_i^{\beta_i'}$, then

$$X_{d+1}^{\beta} - \prod_{i=1}^{n} a_i^{\beta_i} = x_k(X_d^{\beta'} - \prod_{i=1}^{n} a_i^{\beta_i'}) + (x_k - a_k) \prod_{i=1}^{n} a_i^{\beta_i'},$$

which implies that $X_{d+1}^{\beta} - \prod_{i=1}^{n} a_i^{\beta_i} \in \langle \mathcal{F} \rangle$. Therefore, for all nontrivial multilinear monomials X^{β}, there exists $p_{i\beta}, q_{j\beta} \in \mathbb{C}[x_1, \ldots, x_n]$ such that $X^{\beta} - \prod_{i=1}^{n} a_i^{\beta_i} = \sum_{i=1}^{m} p_{i\beta} f_i + \sum_{j=1}^{n} q_{j\beta}(x_j^2 - x_j) \in \langle \mathcal{F} \rangle$, where $i \in [m], j \in [n]$.

The Boolean Macaulay matrix $\begin{bmatrix} M & -\vec{b} \end{bmatrix}$ is the augmented matrix of the Boolean Macaulay linear system $M\vec{y} = \vec{b}$ of the set of polynomials $\mathcal{F}$. Since $X^{\beta} - \prod_{i=1}^{n} a_i^{\beta_i} = \psi\left(X^{\beta} - \prod_{i=1}^{n} a_i^{\beta_i}\right) = \sum_{i=1}^{m} \psi(p_{i\beta} f_i)$, and polynomial addition, subtraction, and multiplication in the polynomial ideal $\langle \mathcal{F} \rangle$ correspond to row operations of the Boolean Macaulay matrix $\begin{bmatrix} M & -\vec{b} \end{bmatrix}$, we can perform row operations on the Boolean Macaulay matrix according to $\sum_{i=1}^{m} \psi(p_{i\beta} f_i)$. By these row operations, we can obtain an extended matrix of form $\begin{bmatrix} M & -\vec{b} \\ I & -\vec{y} \end{bmatrix}$, where the columns of the identity matrix I are indexed by the nontrivial multilinear monomials and entries of $\vec{y}$ are values of $\prod_{i=1}^{n} a_i^{\beta_i}$. As performing row operations on a matrix does not change the matrix rank, the matrix M must have full column rank. Therefore, the Boolean Macaulay linear system has the unique solution $\hat{y} = M^{+}\vec{b}$. □

A.2 Bounds on binomial coefficients

In this appendix, we derive some standard bounds on binomial coefficients for completeness. First, we show that for $h \leq n/2$

$$\sum_{j=0}^{h} \binom{n}{j} \leq 3\sqrt{h}\binom{n}{h}. \tag{A.1}$$

We use the following upper bound [Juk11, Corollary 22.9] on binomial coefficients

$$\begin{aligned}\forall\, 0 < h \leq n/2 :\ \sum_{j=0}^{h} \binom{n}{j} &\leq 2^{n \cdot H(h/n)} \\ &= 2^{n\left(-\frac{h}{n}\log_2\left(\frac{h}{n}\right) - \frac{n-h}{n}\log_2\left(\frac{n-h}{n}\right)\right)} \\ &= \left(\frac{n}{h}\right)^h \left(\frac{n}{n-h}\right)^{n-h},\end{aligned}$$

in combination with Stirling's approximation [Rob55] $\sqrt{2\pi}\sqrt{n}\left(\frac{n}{e}\right)^n \leq n! \leq e\sqrt{n}\left(\frac{n}{e}\right)^n$, yielding

$$\begin{aligned}\binom{n}{h} = \frac{n!}{(n-h)!h!} &\geq \frac{\sqrt{2\pi}\sqrt{n}\left(\frac{n}{e}\right)^n}{e\sqrt{n-h}\left(\frac{n-h}{e}\right)^{n-h} e\sqrt{h}\left(\frac{h}{e}\right)^h} \\ &= \frac{\sqrt{2\pi}}{e^2}\sqrt{\frac{n}{n-h}}\frac{1}{\sqrt{h}}\left(\frac{n}{h}\right)^h\left(\frac{n}{n-h}\right)^{n-h} \\ &\geq \frac{1}{3\sqrt{h}}\sum_{j=0}^{h}\binom{n}{j}.\end{aligned}$$

Another bound that we use is

$$\sum_{j=1}^{h}\binom{n}{i} \leq \binom{n}{h}\frac{n-h+1}{n-2h+1}, \tag{A.2}$$

which can be shown by the summation of an upper bound by a geometric series, i.e.,

$$\begin{aligned}\sum_{i=1}^{h}\binom{n}{i} &= \binom{n}{h}(1 + \binom{n}{h-1}/\binom{n}{h} + \binom{n}{h-2}/\binom{n}{h} \\ &\quad + \cdots + \binom{n}{1}/\binom{n}{h}) \\ &\leq \binom{n}{h}(1 + h/(n-h+1) + h^2/(n-h+1)^2 \\ &\quad + \cdots + h^{h-1}/(n-h+1)^{h-1}) \\ &\leq \binom{n}{h}\frac{1}{1-h/(n-h+1)} \\ &= \binom{n}{h}\frac{n-h+1}{n-2h+1}.\end{aligned}$$

Appendix B

B.1 Proof of Lemma 2.5.1

Our analysis of the phase estimation algorithm, as in [Kit96], will use elements of the analyzes in [JZ23] and [Pid19], as well as the following lemma:

Lemma B.1.1 (Effective Spectral Gap Lemma [LMR+11]). *Fix $\epsilon \in [0, \pi)$, and let Λ_ϵ be the orthogonal projector onto the $e^{i\theta}$-eigenspaces of $U_{\mathcal{AB}}$ with $|\theta| \leq \epsilon$. If $|\phi\rangle \in \mathcal{B}$, then*

$$\|\Lambda_\epsilon(I - \Pi_{\mathcal{A}})|\phi\rangle\| \leq \frac{\epsilon}{2}\||\phi\rangle\|.$$

Lemma B.1.2. *Define the unitary $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - 1)(2\Pi_{\mathcal{B}} - 1)$ acting on a Hilbert space $\mathcal{H}$ for projectors $\Pi_{\mathcal{A}}, \Pi_{\mathcal{B}}$ onto some subspaces $\mathcal{A}$ and $\mathcal{B}$ of $\mathcal{H}$ respectively. Let $|\psi\rangle = \sqrt{p}|\varphi\rangle + (I - \Pi_{\mathcal{A}})|\phi\rangle$ be a normalized quantum state such that $U_{\mathcal{AB}}|\varphi\rangle = |\varphi\rangle$ and $|\phi\rangle$ is a (unnormalized) vector satisfying $\Pi_{\mathcal{B}}|\phi\rangle = |\phi\rangle$. Then performing phase estimation on the state $|\psi\rangle$ with operator $U_{\mathcal{AB}}$ and precision δ outputs "0" with probability $p' \in [\frac{4}{\pi^2}p, p + \frac{17\pi^2\||\phi\rangle\|}{16T}]$, leaving a state $|\psi'\rangle$ satisfying*

$$\frac{1}{2}\||\psi'\rangle\langle\psi'| - |\varphi\rangle\langle\varphi|\|_1 \leq \sqrt{\frac{17\pi^4\delta\||\phi\rangle\|}{64p}}.$$

Consequently, when the precision is $O\left(\frac{p\epsilon^2}{\||\phi\rangle\|}\right)$, the resulting state $|\psi'\rangle$ satisfies

$$\frac{1}{2}\||\psi'\rangle\langle\psi'| - |\varphi\rangle\langle\varphi|\|_1 \leq \epsilon.$$

Proof. By the promise that $|\psi\rangle = \sqrt{p}|\varphi\rangle + (I - \Pi_{\mathcal{A}})|\phi\rangle$ with $\Pi_{\mathcal{B}}|\phi\rangle = |\phi\rangle$, we can apply

Lemma B.1.1 to obtain

$$\|\Lambda_\epsilon\left(|\psi\rangle - \sqrt{p}\,|\varphi\rangle\right)\| = \|\Lambda_\epsilon(I - \Pi_{\mathcal{A}})\,|\phi\rangle\| \leq \frac{\epsilon}{2}\,\||\phi\rangle\|\,. \tag{B.1}$$

Let $\{\theta_j\}_{j\in J} \subset (-\pi, \pi]$ be the set of phases of $U_{\mathcal{AB}}$, and let Π_j be the orthogonal projector onto the $e^{i\theta_j}$-eigenspace of $U_{\mathcal{AB}}$, so we can write

$$U_{\mathcal{AB}} = \sum_{j\in J} e^{i\theta_j}\Pi_j.$$

Phase estimation starts by making a superposition over t from 0 to $T-1$ in the phase register and conditioned on this register we apply $U^t_{\mathcal{AB}}$ to $|\psi\rangle$, creating

$$\sum_{t=0}^{T-1} \frac{1}{\sqrt{T}}\,|t\rangle\,U^t_{\mathcal{AB}}\,|\psi\rangle = \sum_{j\in J}\sum_{t=0}^{T-1}\frac{1}{\sqrt{T}}\,|t\rangle\,e^{it\theta_j}\Pi_j\,|\psi\rangle\,.$$

The phase estimation algorithm then proceeds by applying an inverse Fourier transform, $F_T^\dagger$, to the first register and then measuring the result. The probability p' of measuring 0 is

$$\begin{aligned}
p' &:= \left\| \langle 0|\,F_T^\dagger \otimes I \left(\sum_{j\in J}\sum_{t=0}^{T-1}\frac{1}{\sqrt{T}}\,|t\rangle\,e^{it\theta_j}\Pi_j\,|\psi\rangle\right)\right\|^2 \\
&= \left\|\sum_{t=0}^{T-1}\frac{1}{\sqrt{T}}\,\langle t|\otimes I\left(\sum_{j\in J}\sum_{t=0}^{T-1}\frac{1}{\sqrt{T}}\,|t\rangle\,e^{it\theta_j}\Pi_j\,|\psi\rangle\right)\right\|^2 \\
&= \frac{1}{T^2}\left\|\sum_{j\in J}\sum_{t=0}^{T-1} e^{it\theta_j}\Pi_j\,|\psi\rangle\right\|^2 = \frac{1}{T^2}\sum_{j\in J:\theta_j\neq 0}\left|\frac{1-e^{i\theta_j T}}{1-e^{i\theta_j}}\right|^2\|\Pi_j\,|\psi\rangle\|^2 + \|\Lambda_0\,|\psi\rangle\|^2 \\
&= \frac{1}{T^2}\sum_{j\in J:\theta_j\neq 0}\frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)}\,\|\Pi_j\,|\psi\rangle\|^2 + \|\Lambda_0\,|\psi\rangle\|^2\,,
\end{aligned} \tag{B.2}$$

since $\left|\sum_{t=0}^{T-1} e^{it\theta}\right| = \left|\frac{1-e^{i\theta T}}{1-e^{i\theta}}\right|$, and $|1-e^{i\theta}|^2 = 4\sin^2\frac{\theta}{2}$ for any $\theta\in\mathbb{R}$.

For the lower bound on p', we will use the identities $\sin^2\theta \leq \theta^2$ for all θ, and $\sin^2\theta \geq 4\theta^2/\pi^2$ whenever $|\theta| \leq \pi/2$. Let $\Phi = \frac{\pi}{T}$. If we apply this to Eq. (B.2), we find

that

$$p' \geq \frac{1}{T^2} \sum_{j\in J:0<|\theta_j|\leq\Phi} \frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)} \left\|\Pi_j \left|\psi\right\rangle\right\|^2 + \left\|\Lambda_0 \left|\psi\right\rangle\right\|^2$$
$$\geq \frac{1}{T^2} \sum_{j\in J:0<|\theta_j|\leq\Phi} \frac{4(T\theta_j/2)^2/\pi^2}{(\theta_j/2)^2} \left\|\Pi_j \left|\psi\right\rangle\right\|^2 + \left\|\Lambda_0 \left|\psi\right\rangle\right\|^2 \geq \frac{4}{\pi^2} \left\|\Lambda_\Phi \left|\psi\right\rangle\right\|^2.$$

By applying Eq. (B.1) with $\epsilon = 0$ and the triangle inequality, we obtain

$$\left\|\Lambda_\Phi \left|\psi\right\rangle\right\| \geq \left\|\Lambda_0 \sqrt{p} \left|\varphi\right\rangle\right\| - \left\|\Lambda_0 \left(\left|\psi\right\rangle - \sqrt{p} \left|\varphi\right\rangle\right)\right\| = \sqrt{p},$$

since $\left|\varphi\right\rangle$ is an 1-eigenvector of U, thus concluding the lower bound.

For the upper bound, we make use of the identity $\sin^2\theta \leq \min\{1, \theta^2\}$ for all θ. In combination with Eq. (B.1), this allows us to upper bound p' from where we left off in Eq. (B.2) as

$$\begin{aligned}
p' &\leq \frac{1}{T^2} \sum_{j\in J:\theta_j\neq 0} \frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)} \left(\left\|\Pi_j \left(\left|\psi\right\rangle - \sqrt{p} \left|\varphi\right\rangle\right)\right\|^2 + \left\|\Pi_j \sqrt{p} \left|\varphi\right\rangle\right\|^2\right) + \left\|\Lambda_0 \sqrt{p} \left|\psi\right\rangle\right\|^2 \\
&= \frac{1}{T^2} \sum_{j\in J:|\theta_j|<\sqrt{\frac{1}{\||\phi\rangle\|T}}} \frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)} \left\|\Pi_j \left(\left|\psi\right\rangle - \sqrt{p} \left|\varphi\right\rangle\right)\right\|^2 \\
&+ \frac{1}{T^2} \sum_{j\in J:|\theta_j|\geq\sqrt{\frac{1}{\||\phi\rangle\|T}}} \frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)} \left\|\Pi_j \left(\left|\psi\right\rangle - \sqrt{p} \left|\varphi\right\rangle\right)\right\|^2 + \frac{p}{T^2} \left\|\sum_{j\in J} \sum_{t=0}^{T-1} e^{it\theta_j} \Pi_j \left|\theta\right\rangle\right\|^2 \\
&\leq \frac{1}{T^2} \sum_{j\in J:|\theta_j|<\sqrt{\frac{1}{\||\phi\rangle\|T}}} \frac{\pi^2 T^2}{4} \frac{\left\|\left|\phi\right\rangle\right\|}{4T} + \frac{1}{T^2} \sum_{j\in J:|\theta_j|\geq\sqrt{\frac{1}{\||\phi\rangle\|T}}} \pi^2 \left\|\left|\phi\right\rangle\right\| T \left\|\Pi_j \left(\left|\psi\right\rangle - \sqrt{p} \left|\varphi\right\rangle\right)\right\|^2 + p \\
&\leq p + \frac{17\pi^2 \left\|\left|\phi\right\rangle\right\|}{16T}.
\end{aligned}$$

Finally, let $\left|\psi'\right\rangle$ be the (normalized) post-measurement state after measuring 0. We abbreviate PE for the phase estimation algorithm followed by the projection onto measuring 0, as described in Eq. (B.2), such that $\left|\psi'\right\rangle = \frac{1}{\sqrt{p'}} \mathsf{PE} \left|\psi\right\rangle$. Note that since $\left|\varphi\right\rangle$ is an 1-eigenvector of U, we have $\left|\varphi\right\rangle = \mathsf{PE} \left|\varphi\right\rangle$, meaning we can conclude the lemma via the inequality

$$\frac{1}{2} \left\|\left|\psi'\right\rangle\left\langle\psi'\right| - \left|\varphi\right\rangle\left\langle\varphi\right|\right\|_1 \leq \sqrt{1 - \left|\left\langle\psi'|\varphi\right\rangle\right|^2} = \sqrt{1 - \frac{\left|\left\langle\psi\right| \mathsf{PE} \left|\varphi\right\rangle\right|^2}{p'}} = \sqrt{1 - \frac{p}{p'}} \leq \sqrt{\frac{17\pi^4 \left\|\left|\phi\right\rangle\right\|}{64Tp}}.$$

□

B.2 Expansion properties of a random bipartite graph

We will first present a technical lemma that we need to use.

Lemma B.2.1. *Let* $1 \leq s \leq \frac{2}{3}N$ *and* $\delta = 1/(2\log N)$ *Let* $p_s = \frac{(N-s)!((1+\delta)s)!)^2}{s!(N-\delta s)!((\delta s)!)^3}$, *then*

$$\log p_s \lesssim \log(3/2)(\delta - 1)s.$$

We omit the proof of this lemma because it follows almost exactly the analysis of the proof in [Kow19, Theorem 4.1.1].

Next, we show D-regular random bipartite graph is a mild expander graph as stated in Lemma 4.4.6. We restate the lemma here:

Lemma (Lemma 4.4.6 in Section 4.4.4)**.** *Let* L *and* R *be two sets of vertices with* $|L| = |R| = N$. *Link* L *and* R *through* $D \geq 3$ *random perfect matchings. Denote the resulting graph by* $G_B = (V_B, E_B)$, *and let* $\chi = 2/3$, $\delta = 1/(2\log N)$. *Then* G_B *has the following expansion properties with probability* $1 - \Theta(1/N^{2\log(3/2)(1-\delta)})$*:*

(i) For any subset $L' \subseteq L$ *and* $|L'| \leq \chi N$, *we have* $|\Gamma(L')\backslash L'| \geq (1+\delta)|L'|$, *where* $\Gamma(L')$ *denotes the neighborhood of* L' *as defined in Definition 4.4.1.*

(ii) For any subset $T \subseteq L \cup R$ *and* $|T| \leq N$, *we have* $|\Gamma(T)\backslash T| \geq \frac{\delta}{2}|T|$.

In other words, with probability $1 - \Theta(1/N^{2\log(3/2)(1-\delta)})$, *this bipartite regular graph is a mild expander graph.*

Proof. To prove (i), it suffices to show that the probability of there existing $L' \subset L$ such that $|L'| \leq \chi N$ and $|\Gamma(L')\backslash L'| < (1+\delta)|L'|$ is small. For simplicity and slight abuse of the notation, in the rest of the proof, we use $\delta|L|$ to represent $\lfloor \delta|L| \rfloor$.

In one of the random perfect matchings, let the neighbors of L' in R be I_1, then $|I_1| = |L'|$. Therefore $|\Gamma(L')\backslash L'| \geq |L'|$. If $|\Gamma(L')\backslash L'| \leq (1+\delta)|L'|$ with the D random perfect matchings, it is necessary that at most $\delta|L'|$ of the values of the other $D-1$ random perfect matchings are outside the set I_1. Each random perfect match is chosen independently with probability $1/N!$. For a specific set I_δ of size $\delta|L'|$, the probability

that $\Gamma(L')\backslash L' \subseteq I_1 \cup I_\delta$ is bounded as follows:

$$Pr[\Gamma(L')\backslash L' \subseteq I_1 \cup I_\delta] \leq \left(\frac{((1+\delta)|L'|)((1+\delta)|L'|-1)\cdots((1+\delta)|L'|-|L'|+1)!}{N!}\right)^{D-1}$$
$$= \left(\frac{((1+\delta)|L'|)!}{(\delta|L'|)!N!}\right)^{D-1}.$$

Note that this probability decreases with D. For the choice of the additional set of I_δ in R of size $\delta|L'|$, there are in total $\binom{N}{\delta|L'|}$ options. Therefore,

$$\begin{aligned}
\Pr[\text{ exists } L' \subseteq L \text{ with } |\Gamma(L')\backslash L'| < (1+\delta)|L'| \,] &\leq \sum_{L'\subset L, |L'|\leq \chi N} \binom{N}{\delta|L'|} \Pr[\Gamma(L') \subseteq I_1 \cup I_\delta] \\
&= \sum_{s=1}^{\chi N} \binom{N}{s}\binom{N}{\delta s}\left(\frac{(N-s)!((1+\delta)s)!}{(\delta s)!N!}\right)^{D-1} \\
&= \sum_{s=1}^{\chi N} \left(\frac{(N-s)!}{N!}\right)^{D-3} \frac{(N-s)!\left((1+\delta)s\right)!)^{D-1}}{s!(N-\delta s)!\left((\delta s)!\right)^{D}} \\
&\leq \sum_{s=1}^{\chi N} \frac{(N-s)!\left((1+\delta)s\right)!)^{2}}{s!(N-\delta s)!\left((\delta s)!\right)^{3}}.
\end{aligned}$$

To upper bound the probability, we will bound the value of each term $p_s = \frac{(N-s)!((1+\delta)s)!)^2}{s!(N-\delta s)!((\delta s)!)^3}$. Let $\delta = 1/(2\log N)$, we consider the following two cases of s,

1. $1 \leq s \leq 2\log N$, we have $p_s = \frac{(N-s)!s!}{N!} = \frac{s(s-1)\cdots 2\cdot 1}{N(N-1)\cdots(N-s+1)}$ and
$$\sum_{s=1}^{2\log N} p_s = \Theta(1/N).$$

2. $2\log N \leq s \leq \chi N$, using the Stirling Formula $\log(k!) = k\log k - k + \frac{1}{2}\log(2\pi k) + O(1/k)$ and following Lemma B.2.1, we know that $\log p_s \lesssim \log(3/2)(\delta-1)s$. That is $p_s \approx 2^{-\log(3/2)(1-\delta)s} = a^s$, where $a = 2^{-\log(3/2)(1-\delta)} < 1$ is a constant. Thus
$$\sum_{s=2\log N}^{\chi N} a^s = a^{2\log N}\left(\frac{1-a^{\chi N - 2\log N+1}}{1-a}\right) = \Theta(N^{2\log a}) = \Theta(1/N^{2\log(3/2)(1-\delta)}).$$

Combine the two cases, we have

$$\Pr[\text{There exists } L' \subseteq L \text{ with } |\Gamma(L')\backslash L'| < (1+\delta)|L'| \,] = \Theta(1/N^{2\log(3/2)(1-\delta)}).$$

In other words, with probability at least $1 - \Theta(1/N^{2\log(3/2)(1-\delta)})$, for any subset $L' \subset L$ and $|L'| \leq \chi N$, we have $|\Gamma(L')\backslash L'| \geq (1+\delta)|L'|$.

Next, we prove (ii), that is, for any subset $T \subseteq L \cup R$ with $|T| \leq N$, with high probability, we have $\Gamma(T) \geq \delta|T|/2$. This provides us with the expansion property of an arbitrary subgraph of G_B of size at most N. Let $T = L' \cup R'$ with $L' = T \cap L$ and $R' = T \cap R$. Without loss of generality, assume that $|L'| \geq |R'|$, then $|L'| \geq |T|/2$.

- If $\chi N \leq |L'| \leq N$, by the injection property of a perfect matching and $|T| \leq N$, we know that $|\Gamma(L')\backslash L'| \geq |L'| \geq \chi N$ and $|R'| \leq (1-\chi)N$. Therefore, there are at least $|L'|-|R'| \geq (2\chi-1)N$ neighbors of L' out of T, that is $|\Gamma(T)\backslash T| \geq (2\chi-1)N$.

- If $|L'| \leq \chi N$, we know that $|\Gamma(L')\backslash L'| \geq (1+\delta)|L'|$. In addition to the vertices in the set L', there are at most $|R'|$ neighbors of L' that are in T. Therefore L' has at least $|L'| - |R'| \geq \delta|L'| \geq \frac{\delta}{2}|T|$ neighbors that are not contained in T. As a result, for any subset $T \subseteq L \cup R$ and $|T| \leq N$, we have$|\Gamma(T)\backslash T| \geq \frac{\delta}{2}|T|$. In other words, for any subset $T \subseteq L \cup R$, we have

$$|\Gamma(T)\backslash T| \geq \frac{\delta}{2}|T|.$$

Thus, for any subset $T \subseteq L \cup R$ with $|T| \leq N$, with probability $1 - 1/\Theta(N^{2\log(3/2)(1-\delta)})$, we have $|\Gamma(T)\backslash T| \geq \delta|T|/2$. □

B.3 Spectrum estimates

We will use the following lemma to upper bound the matrix spectral norm.

Lemma B.3.1. *Let $A = (A_{ij})_{n\times n}$ be a Hermitian matrix. Then*

$$\|A\| \leq \max_{|x\rangle : \||x\rangle\|_\infty \leq 1} \|A\,|x\rangle\,\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |A_{ij}|,$$

where $\|\cdot\|_\infty$ denotes the infinity norm.

This result in fact holds for any general matrix. To prove that we only need to dilate the matrix A to be $\begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}$. We will not discuss this because we are only going to use the Hermitian version.

Proof. Because A is Hermitian, either $\|A\|$ or $-\|A\|$ must be an eigenvalue of A. Therefore there exists $|\phi\rangle$ such that $A|\phi\rangle = \pm\|A\||\phi\rangle$ and $\||\phi\rangle\|_\infty = 1$. Consequently $\|A|\phi\rangle\|_\infty = \|A\|$, and this proves the inequality. The equality can be easily checked and we omit the proof. □

We will then prove Lemma 4.4.14 in the main text, which we restate below:

Lemma B.3.2. *Let*

$$D_1 = \begin{pmatrix} 0 & t_1 & & & \\ t_1 & 0 & t_2 & & \\ & t_2 & 0 & \ddots & \\ & & \ddots & \ddots & t_{m-1} \\ & & & t_{m-1} & 0 \end{pmatrix}_{m\times m} \tag{B.3}$$

as defined in Eq. (4.57)*, in which* $t_1 = \sqrt{d-2}$*,* $t_2 = \cdots = t_{m-1} = \sqrt{d-1}$*. Then* D_1 *has a non-degenerate* 0*-eigenstate* $|\Psi\rangle = (\Psi_1, \Psi_2, \cdots, \Psi_m)^\top$ *where*

$$\Psi_j = \prod_{k=1}^{(j-1)/2} \left(-\frac{t_{2k-1}}{t_{2k}}\right) \Psi_1 = (-1)^{(j-1)/2}\sqrt{\frac{d-2}{d-1}}\Psi_1, \quad \textit{for all odd } j \geq 2,$$

and $\Psi_j = 0$ *for even* j*. Moreover,* 0 *is separated from the rest of the spectrum of* D_1 *by a gap of at least* $2\sqrt{d-2}/(m-1)$*.*

Proof. First, we prove that 0 is an eigenvalue of D_1. This can be done by verifying $D_1|\Psi\rangle = 0$ for the $|\Psi\rangle$ given above.

We then prove the non-degeneracy of the eigenvalue 0 and the spectral gap. We do this through the eigenvalue interlacing theorem, which tells us that the eigenvalues of D_1 must interlace those of D_1', where D_1' is the $(m-1)\times(m-1)$ sub-matrix of D_1 on

the upper-left corner, i.e.,

$$D_1' = \begin{pmatrix} 0 & t_1 & & & \\ t_1 & 0 & t_2 & & \\ & t_2 & 0 & \ddots & \\ & & \ddots & \ddots & t_{m-2} \\ & & & t_{m-2} & 0 \end{pmatrix}_{(m-1)\times(m-1)} . \tag{B.4}$$

We will therefore first study the spectrum of D_1'.

We observe that D_1' is in fact an invertible matrix. Consider the linear system $D_1' |x\rangle = |y\rangle$, where $|x\rangle = (x_1, \cdots, x_{m-1})^\top$ and $|y\rangle = (y_1, \cdots, y_{m-1})^\top$, then we can compute the solution $|x\rangle$ using the following recursion:

$$\begin{aligned} x_{2k+2} &= \frac{y_{2k+1}}{t_{2k+1}} - \frac{t_{2k}}{t_{2k+1}} x_{2k}, \quad \forall k \geq 1, \\ x_2 &= \frac{y_1}{t_1}, \\ x_{m-2k} &= \frac{y_{m-2k+1}}{t_{m-2k}} - \frac{t_{m-2k+1}}{t_{m-2k}} x_{m-2k+2}, \quad \forall k \geq 1 \\ x_{m-2} &= \frac{y_{m-1}}{t_{m-2}}. \end{aligned} \tag{B.5}$$

Given the values of t_j, we observe that

$$\frac{t_{2k}}{t_{2k+1}} = 1, \quad \frac{t_{m-2k+1}}{t_{m-2k}} = \begin{cases} 1, & \text{if } k < (m-1)/2, \\ \sqrt{\frac{d-1}{d-2}}, & \text{if } k = (m-1)/2. \end{cases}$$

Therefore, for even entries we have

$$|x_{2k+2}| \leq \left| \frac{y_{2k+1}}{t_{2k+1}} \right| + |x_{2k}| \leq \frac{(k+1) \| |y\rangle \|_\infty}{\sqrt{d-2}},$$

For odd entries we have

$$|x_{m-2k}| \leq \left| \frac{y_{m-2k+1}}{t_{m-2k}} \right| + |x_{m-2k+2}| \leq \frac{k \| |y\rangle \|_\infty}{\sqrt{d-1}},$$

for $k < (m-1)/2$. For $k = (m-1)/2$, we have

$$|x_1| \leq \left|\frac{y_2}{t_1}\right| + \sqrt{\frac{d-1}{d-2}}|x_3| \leq \frac{\| |y\rangle \|_\infty}{\sqrt{d-2}} + \frac{\| |y\rangle \|_\infty (m-3)/2}{\sqrt{d-2}} = \frac{\| |y\rangle \|_\infty (m-1)/2}{\sqrt{d-2}}.$$

Therefore we have

$$|x_j| \leq \frac{(m-1)\| |y\rangle \|_\infty}{2\sqrt{d-2}}$$

for all j. From this we can see that if $\| |y\rangle \|_\infty \leq 1$, then

$$\|(D_1')^{-1} |y\rangle \|_\infty = \max_j |x_j| \leq \frac{(m-1)}{2\sqrt{d-2}}.$$

Therefore by Lemma B.3.1 we have $\|(D_1')^{-1}\| \leq (m-1)/(2\sqrt{d-2})$. As a result the eigenvalues of D_1' must be bounded away from 0by at least $2\sqrt{d-2}/(m-1)$.

Because by the eigenvalue interlacing theorem the eigenvalues of D_1 interlace those of D_1', if 0 is a degenerate eigenvalue of D_1, then there must exist an eigenvalue of D_1' between two 0's, i.e., this eigenvalue of D_1' must also be 0. This is impossible because we have just shown that D_1' is invertible. This proves the non-degeneracy of 0 as an eigenvalue of D_1. If there is an eigenvalue λ of D_1 such that $|\lambda| < 2\sqrt{d-2}/(m-1)$, then there must exist an eigenvalue λ' of D_1' between 0 and λ, and therefore $|\lambda'| < 2\sqrt{d-2}/(m-1)$. This is again impossible because we have just shown that all eigenvalues of D_1' must be bounded away from 0 by at least $2\sqrt{d-2}/(m-1)$. Therefore all non-zero eigenvalues of D_1 must be bounded away from 0by at least $2\sqrt{d-2}/(m-1)$. □

Lemma B.3.3 (Inverse of a nonsingular tridiagonal matrix [DFP01, Theorem 2.1])**.** *Let $H_1(a,b)$ be as defined in Eq.* (4.59)*. Let σ, δ be the two m dimensional vectors defined as follows:*

1. *$\sigma_m = b$, $\sigma_i = -t_i^2/\sigma_{i+1}$ for $i = m-1, \ldots, 2$, $\sigma_1 = a - t_1^2/\sigma_2$.*
2. *$\delta_1 = a$, $\delta_i = -t_{i-1}^2/\delta_{i-1}$ for $i = 2, \ldots, m-1$, $\delta_m = b - t_{m-1}^2/\sigma_{m-1}$.*

Then the matrix element of the inverse of $H_1(a,b))$ is

$$(H_1^{-1}(a,b))_{i,j} = \begin{cases} (-1)^{i+j} t_i \cdots t_{j-1} \frac{\sigma_{j+1} \cdots \sigma_m}{\delta_i \cdots \delta_m} & \text{if } i \leq j \\ (-1)^{i+j} t_j \cdots t_{i-1} \frac{\sigma_{i+1} \cdots \sigma_m}{\delta_j \cdots \delta_m} & \text{if } i > j \end{cases}, \tag{B.6}$$

with the convention that the empty product equals 1.

Corollary B.3.4. *Let $t_1 = \sqrt{d-2}$, $t_2 = t_3 = \cdots = t_{m-1} = \sqrt{d-1}$, $\gamma = \frac{d-1}{2}$, for integer $d \geq 3$, and let m be an odd integer. Let $\mu_l = 2\cos(2\pi l/n)$ with $l = 0, 1 \ldots, n-1$ satisfying $\mu_l \neq 0$. Let $a = \mu_l$ and $b = \gamma\mu_l$. Then we have*

$$\max_{i,j}(H_1^{-1}(a,b))_{i,j} = O(1/|a|^2) = O(n^2). \tag{B.7}$$

Proof. Since $t_1 = \sqrt{d-2}$, $t_2 = t_3 = \ldots = t_{m-1} = \sqrt{d-1}$, $\gamma = \frac{d-1}{2}$ and m is an odd integer, we have $t_1 t_2 = \sqrt{(d-2)(d-1)}$, $t_i t_{i+1} = (d-1)$ for $2 \leq i \leq m-1$, $\sigma_1 = a\left(1 + \frac{\gamma(d-1)}{d-2}\right)$, $|\sigma_i \sigma_{i+1}| = (d-1)$ for $i = 2, \ldots, m-1$ and $\delta_1 = a$, $|\delta_i \delta_{i+1}| = (d-1)$ for $2 \leq i \leq m-2$, $\delta_m = a(1 + \frac{\gamma(d-1)}{d-2})$. We can then compute the two m dimensional vectors σ, δ in Lemma B.3.3 to be:

1. $\sigma_i = -\frac{(d-1)}{b}, \sigma_{i+1} = b$, for i to be an even integer from $m-1$ to 2, $\sigma_1 = a + \frac{b(d-1)}{(d-2)}$.
2. $\delta_1 = a$, $\delta_i = -(d-2)/a$, $\delta_{i+1} = \frac{a(d-1)}{(d-2)}$ for i to be an even integer from 2 to $m-1$, $\delta_m = b + \frac{a(d-1)}{(d-2)}$.

Using the results above we will compute upper bounds for the matrix entries $(H_1^{-1}(a,b))_{i,j}$. Note that because $H_1(a,b)$ is Hermitian, we only need to consider the case of $i \leq j$. By Lemma B.3.3 we have

$$|(H_1^{-1}(a,b))_{i,j}| = \frac{t_i t_{i+1} \cdots t_{j-1}}{\delta_i \delta_{i+1} \cdots \delta_{j-1}} \cdot \frac{1}{\delta_j} \cdot \frac{\sigma_{j+1}\sigma_{j+2}\cdots\sigma_m}{\delta_{j+1}\delta_{j+2}\cdots\delta_m}. \tag{B.8}$$

We will next deal with the three parts on the right-hand side separately.

For the first part $\frac{t_i t_{i+1}\cdots t_{j-1}}{\delta_i\delta_{i+1}\cdots\delta_{j-1}}$, we consider two different cases. If $j-i$ is even, then

$$\begin{aligned}
\left|\frac{t_i t_{i+1}\cdots t_{j-1}}{\delta_i\delta_{i+1}\cdots\delta_{j-1}}\right| &= \left|\frac{(t_i t_{i+1})(t_{i+2}t_{i+3})\cdots(t_{j-2}t_{j-1})}{(\delta_i\delta_{i+1})(\delta_{i+2}\delta_{i+3})\cdots(\delta_{j-2}\delta_{j-1})}\right| \\
&= \left|\frac{t_i t_{i+1}}{t_i^2}\frac{t_{i+2}t_{i+3}}{t_{i+2}^2}\cdots\frac{t_{j-2}t_{j-1}}{t_{j-2}^2}\right| \\
&= \left|\frac{t_{i+1}}{t_i}\frac{t_{i+3}}{t_{i+2}}\cdots\frac{t_{j-1}}{t_{j-2}}\right| \\
&= \begin{cases} \sqrt{\frac{d-1}{d-2}} & \text{if } i = 1, \\ 1 & \text{if } i \geq 2. \end{cases}
\end{aligned} \tag{B.9}$$

Therefore

$$\left|\frac{t_i t_{i+1}\cdots t_{j-1}}{\delta_i\delta_{i+1}\cdots\delta_{j-1}}\right| \leq \sqrt{\frac{d-1}{d-2}}, \tag{B.10}$$

when $j-i$ is even.

When $j-i$ is odd, then

$$\left|\frac{t_i t_{i+1}\cdots t_{j-1}}{\delta_i\delta_{i+1}\cdots\delta_{j-1}}\right| = \left|\frac{t_i t_{i+1}\cdots t_{j-2}}{\delta_i\delta_{i+1}\cdots\delta_{j-2}}\right|\left|\frac{t_{j-1}}{\delta_{j-1}}\right| \le \sqrt{\frac{d-1}{d-2}}\left|\frac{t_{j-1}}{\delta_{j-1}}\right|,$$

where we have used Eq. (B.10). Note that

$$\left|\frac{t_{j-1}}{\delta_{j-1}}\right| = \begin{cases} \frac{a\sqrt{d-1}}{d-2}, & \text{if } j \text{ is odd,} \\ \frac{\sqrt{d-2}}{a}, & \text{if } j=2, \\ \frac{d-2}{a\sqrt{d-1}}, & \text{otherwise.} \end{cases} \tag{B.11}$$

Therefore

$$\left|\frac{t_i t_{i+1}\cdots t_{j-1}}{\delta_i\delta_{i+1}\cdots\delta_{j-1}}\right| \le \max\left\{\frac{a(d-2)}{(d-2)^{3/2}}, \frac{\sqrt{d-1}}{a}\right\}, \tag{B.12}$$

when $j-i$ is odd. Combining the above inequality with Eq. (B.10), we have

$$\left|\frac{t_i t_{i+1}\cdots t_{j-1}}{\delta_i\delta_{i+1}\cdots\delta_{j-1}}\right| = O(1/a), \tag{B.13}$$

for all pairs of $i \le j$.

For the third part $\frac{\sigma_{j+1}\sigma_{j+2}\cdots\sigma_m}{\delta_{j+1}\delta_{j+2}\cdots\delta_m}$ in Eq. (B.8), when $m-j$ is even, which implies that j is odd, we use the fact that $\sigma_l\sigma_{l+1} = \delta_l\delta_{l+1}$ for $l \ge 2$ to show that

$$\frac{\sigma_{j+1}\sigma_{j+2}\cdots\sigma_m}{\delta_{j+1}\delta_{j+2}\cdots\delta_m} = 1.$$

When $m-j$ is odd, which implies that j is even,

$$\frac{\sigma_{j+1}\sigma_{j+2}\cdots\sigma_m}{\delta_{j+1}\delta_{j+2}\cdots\delta_m} = \frac{\sigma_{j+1}}{\delta_{j+1}}\frac{\sigma_{j+2}\sigma_{j+2}\cdots\sigma_m}{\delta_{j+2}\delta_{j+2}\cdots\delta_m} = \frac{\sigma_{j+1}}{\delta_{j+1}}.$$

Note that

$$\left|\frac{\sigma_{j+1}}{\delta_{j+1}}\right| = \begin{cases} \frac{b(d-2)}{a(d-1)} = \frac{\gamma(d-2)}{d-1}, & \text{if } j < m-1, \\ \frac{b(d-2)}{b(d-2)+a(d-a)} \le 1, & \text{if } j = m-1. \end{cases}$$

Therefore $\left|\frac{\sigma_{j+1}}{\delta_{j+1}}\right| = O(1)$. Consequently

$$\left|\frac{\sigma_{j+1}\sigma_{j+2}\cdots\sigma_m}{\delta_{j+1}\delta_{j+2}\cdots\delta_m}\right| = O(1). \tag{B.14}$$

For the second part in Eq. (B.8), we readily have $1/|\delta_j| = O(1/a)$. Combining this with Eq. (B.13) and Eq. (B.14), we have through Eq. (B.8),

$$|(H_1^{-1}(a,b))_{i,j}| = O(1/|a|^2). \tag{B.15}$$

Note that because $a = 2\cos(2l\pi/n)$, the smallest possible $|a|$ is

$$|a| = 2\sin(2\pi/n) = \Omega(1/n).$$

Therefore we have the result as stated in this corollary. □

We will then prove the Lemma 4.4.13 in the main text, which we restate here:

Lemma. *Let $H_1(a,b)$ be as defined in Eq. (4.59), with odd m, $t_1 = \sqrt{d-2}$, $t_2 = t_3 = \cdots = t_{m-1} = \sqrt{d-1}$, $\gamma = \frac{d-1}{2}$, for integer $d \geq 3$. Let $\mu_l = 2\cos(2\pi l/n)$ with $l = 0, 1 \ldots, n-1$ satisfying $\mu_l \neq 0$. Let $a = \mu_l$ and $b = \gamma\mu_l$. Then $\|H_1^{-1}(a,b)\| = O(mn^2)$.*

Proof. This lemma is directly proved by combining Corollary B.3.4 with Lemma B.3.1. □

Bibliography

[Aar15] Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291–293, 2015. URL: https://scottaaronson.com/papers/qml.pdf, doi:10.1038/nphys3272.

[Aar21] Scott Aaronson. Open problems related to quantum query complexity. *ACM Transactions on Quantum Computing*, 2(4):1–9, 2021. arXiv: 2109.06917

[Aar22] Scott Aaronson. How much structure is needed for huge quantum speedups? *arXiv preprint arXiv:2209.06930*, 2022. arXiv: 2209.06930

[ABC+20a] Srinivasan Arunachalam, Aleksandrs Belovs, Andrew M Childs, Robin Kothari, Ansis Rosmanis, and Ronald de Wolf. Quantum coupon collector. *arXiv preprint arXiv:2002.07688*, 2020.

[ABC+20b] Srinivasan Arunachalam, Aleksandrs Belovs, Andrew M. Childs, Robin Kothari, Ansis Rosmanis, and Ronald de Wolf. Quantum coupon collector. In *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*, pages 10:1–10:17, 2020. arXiv: 2002.07688 doi:10.4230/LIPIcs.TQC.2020.10.

[ACL+24] Sarah Arpin, Mingjie Chen, Kristin E Lauter, Renate Scheidler, Katherine E Stange, and Ha TN Tran. Orientations and cycles in supersingular isogeny graphs. In *Research Directions in Number Theory: Women in Numbers V*, pages 25–86. Springer, 2024.

[ACR+10] Andris Ambainis, Andrew M Childs, Ben W Reichardt, Robert Špalek, and Shengyu Zhang. Any and-or formula of size n can be evaluated in time nˆ1/2+o(1) on a quantum computer. *SIAM Journal on Computing*, 39(6):2513–2530, 2010.

[AFI+04] Gwénolé Ars, Jean-Charles Faugere, Hideki Imai, Mitsuru Kawazoe, and Makoto Sugita. Comparison between xl and gröbner basis algorithms. pages 338–353, 2004.

[AJL06] Dorit Aharonov, Vaughan Jones, and Zeph Landau. A polynomial quantum algorithm for approximating the Jones polynomial. In *Proceedings of the*

thirty-eighth annual ACM symposium on Theory of computing, pages 427–436, 2006.

[Amb10] Andris Ambainis. Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations. *arXiv preprint arXiv:1010.4458*, 2010.

[Amb12] Andris Ambainis. Variable time amplitude amplification and quantum algorithms for linear algebra problems. In *Proceedings of the 29th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 636–647, 2012. arXiv: 1010.4458 doi:10.4230/LIPIcs.STACS.2012.636.

[AOAGC18] Eric R Anschuetz, Jonathan P Olson, Alán Aspuru-Guzik, and Yudong Cao. Variational quantum factoring. *arXiv preprint arXiv:1808.08927*, 2018.

[AP22] Simon Apers and Stephen Piddock. Elfs, trees and quantum walks. *arXiv preprint arXiv:2211.16379*, 2022. arXiv: 2211.16379

[AT07] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation. *SIAM Journal on Computing*, 37(1):47–82, 2007. Earlier version in STOC'03, arXiv: quant-ph/0301023 doi:10.1137/060648829.

[Bat13] Kim Batselier. A numerical linear algebra framework for solving problems with multivariate polynomials, 2013.

[BBHT98] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4–5):493–505, 1998. arXiv: quant-ph/9605034 doi:10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P.

[BBK+23] Ryan Babbush, Dominic W Berry, Robin Kothari, Rolando D Somma, and Nathan Wiebe. Exponential quantum speedup in simulating coupled classical oscillators. *Physical Review X*, 13(4):041041, 2023.

[BDCG+20] Shalev Ben-David, Andrew M. Childs, András Gilyén, William Kretschmer, Supartha Podder, and Daochen Wang. Symmetries, graph properties, and quantum speedups. In *Proceedings of the 61st IEEE Symposium on Foundations of Computer Science (FOCS)*, 2020. To appear. arXiv: 2006.12760

[Bel13] Aleksandrs Belovs. Quantum walks and electric networks. arXiv: 1302.3143, 2013.

[BFSS13] Magali Bardet, Jean-Charles Faugère, Bruno Salvy, and Pierre-Jean Spaenlehauer. On the complexity of solving quadratic boolean systems. *Journal of Complexity*, 29:53–75, 2013.

[BKW19] Andreas Björklund, Petteri Kaski, and Ryan Williams. Solving systems of polynomial equations over gf(2) by a parity-counting self-reduction. In Ioannis Chatzigiannakis, Christel Baier, Stefano Leonardi, and Paola Flocchini, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019*, Leibniz international proceedings in informatics, pages 1–13, Germany, July 2019. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. International Colloquium on Automata, Languages and Programming, ICALP ; Conference date: 08-07-2019 Through 12-07-2019. doi:10.4230/LIPIcs.ICALP.2019.26.

[BLH23] Shankar Balasubramanian, Tongyang Li, and Aram Harrow. Exponential speedups for quantum walks in random hierarchical graphs. *arXiv preprint arXiv:2307.15062*, 2023. arXiv: 2307.15062

[BS16] Jean-François Biasse and Fang Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 893–902. SIAM, 2016.

[Buc18] Bruno Buchberger. Gröbner bases computation by triangularizing macaulay matrices. In *The 50th Anniversary of Gröbner Bases*, volume 77, pages 25–34. Mathematical Society of Japan, 2018.

[Bur02] Christopher J C Burges. Factoring as optimization. *Microsoft Research MSR-TR-200*, 2002.

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on computing*, 26(5):1411–1473, 1997. doi:10.1137/S0097539796300921.

[BY18] Daniel J Bernstein and Bo-Yin Yang. Asymptotically faster quantum algorithms to solve multivariate quadratic equations. pages 487–506, 2018.

[CAS+22] Pedro CS Costa, Dong An, Yuval R Sanders, Yuan Su, Ryan Babbush, and Dominic W Berry. Optimal scaling quantum linear-systems solver via discrete adiabatic theorem. *PRX quantum*, 3(4):040303, 2022.

[CCD+03] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the 35th ACM Symposium on the Theory of Computing (STOC)*, pages 59–68, 2003. arXiv: quant-ph/0209131 doi:10.1145/780542.780552.

[CCG22] Andrew M Childs, Matthew Coudron, and Amin Shiraz Gilani. Quantum algorithms and the power of forgetting. *arXiv preprint arXiv:2211.12447*, 2022.

[CD23] Wouter Castryck and Thomas Decru. An efficient key recovery attack on sidh. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 423–447. Springer, 2023.

[CDF+02] Andrew M Childs, Enrico Deotto, Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Andrew J Landahl. Quantum search by measurement. *Physical Review A*, 66(3):032314, 2002.

[CFL+18] Anamaria Costache, Brooke Feigon, Kristin Lauter, Maike Massierer, and Anna Puskás. Ramanujan graphs in cryptography, 2018.

[CG17] Alessio Caminata and Elisa Gorla. Solving multivariate polynomial systems and an invariant from commutative algebra. *arXiv preprint arXiv:1706.06319*, 2017.

[CG22] Yu-Ao Chen and Xiao-Shan Gao. Quantum algorithm for boolean equation solving and quantum algebraic attack on cryptosystems. *Journal of Systems Science and Complexity*, 35(1):373–412, 2022.

[CGJ19] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 33:1–33:14, 2019. arXiv: 1804.01973 doi:10.4230/LIPIcs.ICALP.2019.33.

[CGL+20] Nai-Hui Chia, András Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang. Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning. In *Proceedings of the 52nd ACM Symposium on the Theory of Computing (STOC)*, page 387–400, 2020. arXiv: 1910.06151 doi:10.1145/3357713.3384314.

[CGY18] Yu-Ao Chen, Xiao-Shan Gao, and Chun-Ming Yuan. Quantum algorithm for optimization and polynomial system solving over finite field and application to cryptanalysis. *arXiv preprint arXiv:1802.03856*, 2018.

[Chu97] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997. https://mathweb.ucsd.edu/~fan/research/revised.html.

[CJS13] B David Clader, Bryan C Jacobs, and Chad R Sprouse. Preconditioned quantum linear system algorithm. *Physical review letters*, 110:250504, 2013.

[CKM+11] Paul Christiano, Jonathan A Kelner, Aleksander Madry, Daniel A Spielman, and Shang-Hua Teng. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 273–282, 2011.

[CKPS] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. pages 392–407. \{ http://www.minrank.org/xlfull.pdf)}.

[CKS17] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, 46(6):1920–1950, 2017. arXiv: 1511.02306 doi:10.1137/16M1087072.

[CLG09] Denis X. Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22:93–113, 1 2009. doi:10.1007/s00145-007-9002-x.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. Csidh: an efficient post-quantum commutative group action. In *Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III 24*, pages 395–427. Springer, 2018.

[DFKL+20] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. Sqisign: compact post-quantum signatures from quaternions and isogenies. In *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26*, pages 64–93. Springer, 2020.

[DFP01] CM Da Fonseca and J Petronilho. Explicit inverses of some tridiagonal matrices. *Linear Algebra and its Applications*, 325(1-3):7–21, 2001.

[DGG+23] Jintai Ding, Vlad Gheorghiu, András Gilyén, Sean Hallgren, and Jianqiang Li. Limitations of the macaulay matrix approach for using the hhl algorithm to solve multivariate polynomial systems. *Quantum*, 7:1069, 2023.

[DHHM06] Christoph Dürr, Mark Heiligman, Peter HOyer, and Mehdi Mhalla. Quantum query complexity of some graph problems. *SIAM Journal on Computing*, 35(6):1310–1328, 2006.

[Die04] Claus Diem. The {XL}-algorithm and a conjecture from commutative algebra. pages 323–337, 2004.

[DS84] Peter G. Doyle and J. Laurie Snell. *Random walks and electric networks*. Mathematical Association of America, 1984. arXiv: math/0001057 doi:10.5948/UPO9781614440222.

[DS13] Jintai Ding and Dieter Schmidt. Solving degree and degree of regularity for polynomial systems over a finite fields, 2013.

[EH10] Kirsten Eisenträger and Sean Hallgren. Algorithms for ray class groups and Hilbert class fields. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 471–483. SIAM, 2010.

[EH22] Lior Eldar and Sean Hallgren. An efficient quantum algorithm for lattice problems achieving subexponential approximation factor. *arXiv preprint arXiv:2201.13450*, 2022.

[EHKS14] Kirsten Eisenträger, Sean Hallgren, Alexei Kitaev, and Fang Song. A quantum algorithm for computing the unit group of an arbitrary degree number field. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 293–302, 2014.

[EHL+18] Kirsten Eisenträger, Sean Hallgren, Kristin Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: reductions and solutions. In *Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part III 37*, pages 329–368. Springer, 2018.

[EMK06] Moawwad El-Mikkawy and Abdelrahman Karawia. Inversion of general tridiagonal matrices. *Applied Mathematics Letters*, 19(8):712–720, 2006.

[FGG07] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum algorithm for the hamiltonian nand tree. *arXiv preprint quant-ph/0702144*, 2007.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. arXiv: `1411.4028`, 2014.

[FGGS00] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. arXiv: `quant-ph/0001106`, 2000.

[FHK+17] Jean-Charles Faugere, Kelsey Horan, Delaram Kahrobaei, Marc Kaplan, Elham Kashefi, and Ludovic Perret. Fast quantum algorithm for solving multivariate quadratic equations. *arXiv preprint arXiv:1712.07211*, 2017.

[Gal22] Steven Galbraith. Breaking supersingular isogeny diffie-hellman (sidh), Aug 2022. URL: `https://ellipticnews.wordpress.com/2022/07/31/breaking-supersingular-isogeny-diffie-hellman-sidh/`.

[GHLS15] Sevag Gharibian, Yichen Huang, Zeph Landau, and Seung Woo Shin. Quantum hamiltonian complexity. *Foundations and Trends® in Theoretical Computer Science*, 10(3):159–282, 2015.

[GHV21] András Gilyén, Matthew B Hastings, and Umesh Vazirani. (sub)Exponential advantage of adiabatic quantum computation with no sign problem. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1357–1369, 2021.

[Gil19] András Gilyén. *Quantum singular value transformation & its algorithmic applications*. PhD thesis, University of Amsterdam, 2019.

[GLP23] Yu Gao, Yang Liu, and Richard Peng. Fully dynamic electrical flows: Sparse maxflow faster than goldberg–rao. *SIAM Journal on Computing*, (0):FOCS21–85, 2023.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th ACM Symposium on the Theory of Computing (STOC)*, pages 212–219, 1996. arXiv: `quant-ph/9605043` `doi:10.1145/237814.237866`.

[GS21] Shirshendu Ganguly and Nikhil Srivastava. On non-localization of eigenvectors of high girth graphs. *International Mathematics Research Notices*, 2021(8):5766–5790, 2021.

[GSLW18] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics [full version], 2018. arXiv: `1806.01838`

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st ACM Symposium on the Theory of Computing (STOC)*, pages 193–204, 2019. arXiv: `1806.01838` `doi:10.1145/3313276.3316366`.

[GTC19] Yimin Ge, Jordi Tura, and J. Ignacio Cirac. Faster ground state preparation and high-precision ground energy estimation with fewer qubits. *Journal of Mathematical Physics*, 60(2):022202, 2019. arXiv: `1712.03193` `doi:10.1063/1.5027484`.

[Hal05] Sean Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 468–474, 2005.

[Hal07] Sean Hallgren. Polynomial-time quantum algorithms for pell's equation and the principal ideal problem. *Journal of the ACM (JACM)*, 54(1):1–19, 2007.

[HH00] Lisa Hales and Sean Hallgren. An improved quantum Fourier transform algorithm and applications. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 515–525. IEEE, 2000.

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103, 10 2009. doi:10.1103/PhysRevLett.103.150502.

[Hil21] Mark Hillery. Finding more than one path through a simple maze with a quantum walk. *Journal of Physics A: Mathematical and Theoretical*, 54(9):095301, 2021.

[HL] Sean Hallgren and Jianqiang Li. A quantum algorithm for the pathfinding problem via the quantum electrical flow. *Under preparation.*

[JDF11] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011. Proceedings 4*, pages 19–34. Springer, 2011.

[JKP23] Stacey Jeffery, Shelby Kimmel, and Alvaro Piedrafita. Quantum algorithm for path-edge sampling. *arXiv preprint arXiv:2303.03319*, 2023. arXiv: 2303.03319

[JS19] Samuel Jaques and John M Schanck. Quantum cryptanalysis in the ram model: Claw-finding attacks on sike. In *Annual International Cryptology Conference*, pages 32–61. Springer, 2019.

[Juk11] Stasys Jukna. *Extremal Combinatorics - With Applications in Computer Science (2nd ed.)*. Texts in Theoretical Computer Science. Springer, 2011. doi:10.1007/978-3-642-17364-6.

[JZ23] Stacey Jeffery and Sebastian Zur. Multidimensional quantum walks. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1125–1130, 2023. arXiv: 2208.13492

[KH18] Daniel Koch and Mark Hillery. Finding paths in tree graphs with a quantum walk. *Physical Review A*, 97(1):012308, 2018.

[Kit96] Alexei Y. Kitaev. Quantum measurements and the Abelian stabilizer problem. *ECCC*, TR96-003, 1996. arXiv: quant-ph/9511026

[Kow19] Emmanuel Kowalski. *An introduction to expander graphs.* Société mathématique de France Paris, 2019.

[KP17] Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 49:1–49:21, 2017. arXiv: 1603.08675 doi:10.4230/LIPIcs.ITCS.2017.49.

[LC17] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian simulation by quantum signal processing. *Physical Review Letters*, 118(1):010501, 2017. arXiv: 1606.02685 doi:10.1103/PhysRevLett.118.010501.

[Li23] Jianqiang Li. Exponential speedup of quantum algorithms for the pathfinding problem. *arXiv preprint arXiv:2307.12492*, 2023. arXiv: 2307.12492

[LLL23] Guanzhong Li, Jingquan Luo, and Lvzhou Li. Recover the original simplicity: concise and deterministic quantum algorithm for the welded tree problem. *arXiv preprint arXiv:2304.08395*, 2023.

[LLL24] Guanzhong Li, Lvzhou Li, and Jingquan Luo. Recovering the original simplicity: succinct and deterministic quantum algorithm for the welded tree problem. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2454–2480. SIAM, 2024.

[LMR+11] Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mário Szegedy. Quantum query complexity of state conversion. In *Proceedings of the 52nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 344–353, 2011. arXiv: 1011.3020 doi:10.1109/FOCS.2011.75.

[LMR14] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10:631–633, 2014. arXiv: 1307.0401 doi:10.1038/nphys3029.

[Lov96] László Lovász. Random walks on graphs: A survey. In Dezső Miklós, Tamás Szőnyi, and Vera T. Sós, editors, *Combinatorics, Paul Erdős is Eighty (Vol. 2)*, Bolyai Society Mathematical Studies, pages 1–46. János Bolyai Mathematical Society, 1996. URL: http://web.cs.elte.hu/~lovasz/erdos.pdf.

[LP16] Russell Lyons and Yuval Peres. *Probability on Trees and Networks*, volume 42 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, New York, 2016. Available at https://rdlyons.pages.iu.edu/. URL: http://dx.doi.org/10.1017/9781316672815, doi:10.1017/9781316672815.

[LT19a] Lin Lin and Yu Tong. Optimal quantum eigenstate filtering with application to solving quantum linear systems. arXiv: 1910.14596, 2019.

[LT19b] Lin Lin and Yu Tong. Solving quantum linear system problem with near-optimal complexity. *arXiv preprint arXiv:1910.14596*, 2019.

[LT24] Jianqiang Li and Yu Tong. Exponential quantum advantage for pathfinding in regular sunflower graphs. *arXiv preprint arXiv:2407.14398*, 2024.

[LZ23] Jianqiang Li and Sebastian Zur. Multidimensional electrical networks and their application to exponential speedups for graph problems. *arXiv preprint arXiv:2311.07372*, 2023.

[Mad13] Aleksander Madry. Navigating central path with electrical flows: From flows to matchings, and back. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 253–262. IEEE, 2013.

[Per16] Ludovic Perret. Bases de gröbner en cryptographie post-quantique, 2016.

[Pid19] Stephen Piddock. Quantum walk search algorithms and effective resistance. arXiv: `1912.04196`, 2019.

[RHK17] Daniel Reitzner, Mark Hillery, and Daniel Koch. Finding paths with quantum walks or quantum walking through a maze. *Physical Review A*, 96(3):032323, 2017.

[Rob55] Herbert Robbins. A remark on Stirling's formula. *The American Mathematical Monthly*, 62(1):26–29, 1955. `doi:10.2307/2308012`.

[S+94] Jonathan Richard Shewchuk et al. An introduction to the conjugate gradient method without the agonizing pain. 1994.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. arXiv: `quant-ph/9508027` `doi:10.1137/S0097539795293172`.

[Sie86] William McC Siebert. *Circuits, signals, and systems*. MIT press, 1986.

[Sim97] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. Earlier version in FOCS'94. `doi:10.1137/S0097539796298637`.

[Spi19] Daniel Spielman. Spectral and algebraic graph theory. *Yale lecture notes, draft of December*, 4:47, 2019.

[Src22] The Mathematica source code is also available at the Woflram Notebook Archive: `https://notebookarchive.org/2022-02-1ec5yyv`, 2022. URL: `https://notebookarchive.org/2022-02-1ec5yyv`.

[SS08] Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 563–568, 2008.

[SSO19] Yiğit Subaşi, Rolando D Somma, and Davide Orsucci. Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing. *Physical review letters*, 122:60504, 2019.

[SX18] Changpeng Shao and Hua Xiang. Quantum circulant preconditioner for a linear system of equations. *Physical Review A*, 98:62321, 2018.

[Sze04] Mario Szegedy. Quantum speed-up of Markov chain based algorithms. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 32–41, 2004. arXiv: quant-ph/0401053 doi: 10.1109/FOCS.2004.53.

[Tan09] Seiichiro Tani. Claw finding algorithms using quantum walk. *Theoretical Computer Science*, 410(50):5285–5297, 2009.

[Tan19] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st ACM Symposium on the Theory of Computing (STOC)*, pages 217–228, 2019. arXiv: 1807.04271 doi:10.1145/3313276.3316310.

[Tan21] Ewin Tang. Quantum principal component analysis only achieves an exponential speedup because of its state preparation assumptions. *Physical Review Letters*, 127(6):060503, 2021. arXiv: 1811.00414 doi: 10.1103/PhysRevLett.127.060503.

[TAWL20] Yu Tong, Dong An, Nathan Wiebe, and Lin Lin. Fast inversion, preconditioned quantum linear system solvers, and fast evaluation of matrix functions. arXiv: 2008.13295, 2020. doi:10.1103/PhysRevA.104.032422.

[Ton22] Yu Tong. *Quantum Eigenstate Filtering and Its Applications*. University of California, Berkeley, 2022.

[Vad12] Salil P Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

[Vis13] Nisheeth K Vishnoi. Lx= b. *Foundations and Trends® in Theoretical Computer Science*, 8(1–2):1–141, 2013. https://www.cs.yale.edu/homes/vishnoi/Lxb-Web.pdf.

[VV86] Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986. Earlier version in STOC'85. doi:10.1016/0304-3975(86)90135-0.

[Wan17] Guoming Wang. Efficient quantum algorithms for analyzing large sparse electrical networks. *Quantum Information & Computation*, 17(11-12):987–1026, 2017. arXiv: 1311.1851

[Wes22] Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1100–1111. IEEE, 2022.

[dW19] Ronald de Wolf. Quantum computing: Lecture notes, 2019. arXiv: 1907.09415

[WW15] Manuela Wiesinger-Widi. Gröbner bases and generalized sylvester matrices, 2015.

[YZ24] Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. *Journal of the ACM*, 71(3):1–50, 2024.

Jianqiang Li

Education

Pennsylvania State University (PSU) State College, PA, USA
Ph.D. Candidate in Computer Science and Engineering *Aug.2018 - present*
- Advisor: Dr. Sean Hallgren
- Research Focus: Quantum Algorithms and Hamiltonian Complexity

Virginia Commonwealth University (VCU) Richmond, VA, USA
Ph.D. Candidate in Computer Science *Jan.2017 - Aug. 2018*
- Advisor: Dr. Sevag Gharibian
- Research Focus: Approximation Algorithms on Local Hamiltonian Problem

Beijing University of Posts and Telecommunications (BUPT) Beijing, China
Master of Science and Computer Technology, School of Computer Science *Aug.2013 - May. 2016*
- Thesis: Quantum State Representation Based on Combinatorial Laplacian Matrix of Star-Relevant Graph

Yunnan University Kunming, Yunnan, China
Bachelor of Science in Information and Computer, School of Mathematics and Statistics *Aug.2009 - May. 2013*
- Thesis: Classical Simulation of Shor's Algorithm

Research

Publications and preprints:

The authors of the following papers are listed in alphabetical order unless mentioned explicitly otherwise.

- Exponential Quantum Advantage for Pathfinding in Regular Sunflower Graphs. Joint work with Yu Tong, 2024. https://arxiv.org/pdf/2407.14398
- Multidimensional Electrical Networks and their Application to Exponential Speedups for Graph Problems, Joint work with Sebastian Zur, 2023. https://arxiv.org/abs/2311.07372.
- Exponential speedup of quantum algorithms for the pathfinding problem, 2023. https://arxiv.org/abs/2307.12492.
- A quantum algorithm for the pathfinding problem via the quantum electrical flow. Under preparation, Joint work with Sean Hallgren, 2023.
- Limitations of the Macaulay matrix approach for using the HHL algorithm to solve multivariate polynomial systems, Quantum, 7:1069, 2023. Joint work with Jintai Ding, Vlad Gheorghiu, András Gilyén, Sean Hallgren https://arxiv.org/abs/2111.00405.
- Quantum state representation based on combinatorial Laplacian matrix of star-relevant graph, Quantum Information Processing, 2015-14(12), 4691-4713. https://arxiv.org/abs/1507.05491, (By contribution) Joint work with Xiubo Chen, Yixian Yang.

Services:

- Journal reviewer: Quantum Information Processing, Quantum, Physical Review A, Physical Review Letter.
- Sub-reviewer (Conferences): QIP 2021, 2022, ISAAC 2023, SODA 2025, SOSA 2025, STACS 2025.
- Volunteer: VCU Computer Science Day Teaching.