University of Latvia
Faculty of Computing

Jevgēnijs Vihrovs

Limitations of Quantum Walks and Randomized Algorithms

Doctoral Thesis

Area: Computer Science
Sub-Area: Mathematical Foundations of Computer Science

Scientific Advisor:
Dr. Comp. Sci., Prof. Andris Ambainis

Riga, 2019

Abstract

In this work, we study the complexity of algorithms in different models of computation. Specifically, we investigate properties and limitations of quantum walk algorithms, as well as lower bounds on the running time of randomized algorithms in the query complexity model.

In the first part of the thesis, we study Grover's quantum walk and search. Firstly, we develop a mathematical characterization of the localization properties of quantum walk, when the state of the quantum walk gets trapped in the same position for the whole duration of the walk. We then show a variety of examples that exhibit such behaviour. Secondly, we investigate the stationary states of the quantum search that are close to the starting state of the walk. In that case Grover's search does not give any advantage over the classical algorithms. We give a complete description of such states and show the necessary and sufficient conditions on the existence of such states.

In the second part, we study lower bound methods in the randomized query complexity model. We examine a family of classical adversary lower bounds that originate from the quantum adversary bounds. We show that they all are asymptotically equivalent for all total functions, and are equal to the fractional block sensitivity lower bound. Then we show examples of separations between them for partial functions. This increases the understanding of the lower bounds and their usefulness. Finally, we investigate the relationship between fractional block sensitivity and block sensitivity for partial functions. We show a separation between these measures and prove its optimality.

Acknowledgements

First of all, I am deeply thankful to my advisor prof. Andris Ambainis for introducing me to the field of theoretical computer science and many exciting research topics, for guiding me through the scientific research process and for the continuous support and help in my work.

I am highly grateful to my friend and coauthor Krišjānis Prūsis, together with whom I learned many things during our graduate studies. I also want to thank my coauthors, Mārtiņš Kokainis and Thomas G. Wong, the latter for being a mentor on some of my very first research projects.

During the course of my graduate studies, I was interning at the National University of Singapore, Centre for Quantum Computing. I wish to express my sincere gratitude to prof. Miklos Santha for the unending hospitality and many fruitful discussions.

I want to thank my dear friends Jelena Polakova, Swagato Sanyal, Hoeteck Wee, Abuzer Yakaryilmaz, as well as my brother Andrey Vihrov and my family for their support during the making of this thesis.

Contents

Introduction

Relevance of the Thesis

Probabilistic and quantum computation can be much more powerful than the classical deterministic computation. There are many famous examples of problems that can be solved more efficiently in these models of computation. For example, Freivalds's randomized $n \times n$ matrix multiplication verification algorithm runs in time $O(n^2)$ [Fre77], while the best known deterministic algorithm requires $O(n^{2.37})$ time [LG14]. Quicksort utilizes a randomized strategy to improve the constant factor in the running time of sorting n numbers [Hoa61]. Shor's quantum integer factorization algorithm takes polynomial time [Sho97], while the best known classical algorithms require sub-exponential time [LL93]. Grover's quantum algorithm finds an item in the database of size n in time $O(\sqrt{n})$ [Gro96], while any classical algorithm requires at least $\Omega(n)$ time.

While in some cases there are significant algorithmic speedups, it is not always true that the randomized or quantum algorithm is more efficient then the deterministic one. One such example is binary search, where one has to find a number in a sorted array of size n. In all three of the aforementioned models of computation, any correct algorithm must have running time at least $\Omega(\log n)$ [Amb99, HNS02, LM04]. The same is true for the task of sorting an array of n numbers, where in each model the running time is $\Omega(n \log n)$.

Can a quantum algorithm algorithm solve an NP-complete problem in polynomial time? What is the maximum possible speedup of a randomized algorithm? How do probabilistic, quantum and deterministic algorithms compare to each other? These and other questions are crucial in understanding the limits of physically possible computational devices. Thus there is a need to develop theoretical tools that allow us to give mathematically robust answers to these questions.

Objectives of the Research

There are two main objectives of this thesis. The first is studying the limitations of quantum walk algorithms. The second is studying the lower bounds of randomized algorithms in the query computational model.

Quantum walks. Quantum walks are a generalization of the classical random walk. The random walk is a probabilistic process in which an object traverses some given structure (for example, a graph) in a randomized manner defined by a transition operator (for example, a Markov chain). There is a vast variety of applications of such processes which often appear in seemingly unrelated areas of study. These include motion of particles in fluids, gambling

processes, stock prices, brain activity, social network analysis and much more. In computer science, random walks serve as the basis of many efficient algorithms [MR95, AF02].

In a quantum walk, the process is defined by the rules of quantum mechanics. The state of the object is a quantum state which is a superposition over the given structure. It is defined by a wave function, and the position of the walker is not definite until the state is measured. The state of the walk evolves according to some unitary transformation. If the measurement is applied, the wave function will collapse and the position of the walker will become a determined classical state. Each position has some probability of being measured according to the wave function of the quantum state.

There are several ways to define quantum walks. In a continuous-time quantum walk, the system evolves continuously according to the Schrödinger equation [FG98]. In a discrete-time quantum walk, a single discrete step of the walk is defined by a unitary operator [Mey96]. There is a close mathematical connection between the two, as the continuous quantum walk can be obtained from the discrete in an appropriate limit [Chi10].

Compared to random walks, quantum walks often exhibit drastically different behaviour. There are examples where the hitting time of the walk (the expected time to reach a particular vertex in a graph) is exponentially faster quantumly than classically [CFG02]. For the n-dimensional hypercube graph, the mixing time (the expected time to spread evenly among the vertices of the graph) of the quantum walk is $O(n)$, while the random walk requires $\Theta(n \log n)$ time [MPAD08]. In contrast to the random walks, quantum walks also do not converge to the stationary distribution.

Since the introduction of quantum walks [ADZ93], quantum walks have become a staple in design of quantum algorithms. They found a prominent application in quantum search on graphs [CG04, SKW03, Amb03b, AKR05], and since then have been developed into a clear mathematical framework [Sze04, San08]. In a seminal work, Ambainis presented an optimal quantum algorithm for the element distinctness problem based on a quantum walk [Amb04]. Since then, a large variety of quantum walk-based algorithms have been discovered, for such problems as triangle finding [MSS05], formula evaluation [ACR+07], group commutativity testing [MN07], speeding up backtracking algorithms [Mon15, AK17] and many others. Grover's search algorithm [Gro96] can also be viewed as a quantum walk on a graph, which is one of the main subjects of this research.

Query complexity. The second object of this work is studying the complexity of the algorithms in the query computational model. Computational complexity studies the amount of time needed to find the answer to a computational problem. There are two sides to this question: an upper bound on the running time is given by presenting an algorithm that solves the problem; a lower bound on the running time needed to solve the problem is obtained by giving a mathematical proof that any correct algorithm requires that amount of time.

One of the most important open questions in theoretical computer science is the P versus NP problem, which asks whether any computational task, where the answer can be verified efficiently also has an efficient algorithm that solves the task. The question would be settled in negative if there were superpolynomial lower bounds for any of the NP problems. However, it is very difficult to prove strong lower bounds in the standard Turing computational model or the circuit model [All96]. Therefore, one approach is to make the task simpler and prove lower bounds in a simpler model.

This is the origin of the query complexity, which offers a clean and simple model for proving algorithmic limitations. In the query model, the algorithm has to compute a function

on a given finite input string. The input can be accessed via a black-box oracle: the algorithm provides the oracle with an index of the string, and the oracle returns the symbol of the input in that position. The algorithm must perform a number of queries and return the answer to the problem based on the collected information about the input. The running time of the algorithm is the number of queries asked to the oracle; all other computational expenses during the run of the algorithm are ignored.

Many standard problems can be cast to the query model, e.g. search in a sorted array or comparison-based sorting. Query complexity offers optimal lower bounds for these problems, which shows that the classical algorithms such as binary search or mergesort are asymptotically optimal [HNS02]. It also has applications in circuit complexity [Weg87], parallel computing [Sim83], construction of oracles in complexity theory [Tar89] and communication complexity [BdW01, GPW17].

Perhaps the most fruitful application of query complexity has been found in quantum computing. There is a large selection of query problems where quantum algorithms are more efficient than the deterministic and randomized algorithms. For example, the main component of Shor's algorithm for period finding and Grover's algorithm can be seen as query procedures [Sho97, Gro96]. Since then, the family of quantum query algorithms has grown substantially [Amb04, MSS05, BR12], and a lot of them have been proven to perform as well in the standard quantum ram model [GLM08], for example, Grover's search and Ambainis's element distinctness algorithm.

Concurrently, query complexity brought up a rich theory of quantum lower bounds. The first general quantum lower bounds used such measures as block sensitivity [Nis89] and approximate polynomial degree [BBC+01]. In an influential paper, Ambainis introduced the quantum adversary method [Amb00]. Since then, the adversary bound has been one of the most effective lower bound methods in query complexity [AS04, LM04, BR12]. A large variety of adversary methods have been found since then [HNS02, ŠS06], culminating in the general adversary bound which exactly characterizes the power of quantum query algorithms [HLŠ07, Rei09]. Adversary bounds have also been lifted back to randomized query complexity, for example, giving optimal lower bounds for the local search problem, which is a crucial problem in combinatorial optimization [Aar06, LM04].

Query complexity has also been used to explore the relationship between deterministic, randomized and quantum models of computation. In this model, quantum algorithms are always more powerful than the randomized ones, and the randomized setting is more powerful than the deterministic setting. It has been shown that the amount of time needed to compute a total function (defined on all inputs) is polynomially related for all three models [BdW02]. There are also a number of examples that separate the power of these models [ABDK16, ABB+17]. The precise relation between these settings remains an important open question. In the general case, there are examples of exponential and unbounded separations between quantum vs. randomized and randomized vs. deterministic running times, such as Simon's problem [Sim97], Forrelation problem [AA18] and promise majority.

Overview of the Results

The thesis consists of two parts, dedicated to the quantum walks and query complexity, respectively.

Quantum Walks. In Part I, we investigate properties of Grover's discrete-time quantum walk on graphs. It is a generalization of Grover's search algorithm that finds a marked item in the database [Gro96]. This kind of quantum walk is widely used in quantum search and design of efficient quantum algorithms. As an example, it finds a marked vertex on a complete graph of n vertices in $O(\sqrt{n})$ time [CG04]. On the 2-dimensional $n \times n$ lattice, it runs in time $O(\sqrt{n \log n})$ [ABN+13]. We define Grover's quantum walk in Chapter 1.

In Chapter 2, we study localization properties of Grover's quantum walk. Localization is a quantum phenomenon where the state of the walk remains the same at all times during the evolution of the walk [VFQF17]. This property of quantum walks has potential applications in the design of quantum algorithms and quantum optics [KRBD10, KKJ15]. The results are as follows:

- We introduce the *oscillatory localization* where the state of the system alternates between two states during the course of the quantum walk. Equivalently, we study eigenstates with eigenvalue 1 of the two steps of the quantum walk operator. We give a complete characterization of such states, and apply the theory of electrical networks to theoretically estimate the amount of localization, given the starting state of the walk.

- As a corollary, we show that low electrical resistance in the corresponding electrical network, as well as high connectivity in the given graph imply oscillatory localization. We then present a wide range of examples that exhibit this kind of localization, including the complete graph, periodic lattices, Boolean hypercube and general edge-transitive graphs.

In Chapter 3, we study scenarios when Grover's search does not give any computational advantage over the classical algorithms. For example, while the quantum search can find a single marked item on the complete graph $O(\sqrt{n})$, if there are two adjacent marked vertices, quantum search does not give any speedup [NR16]. We study such configurations of multiple marked vertices in graphs where Grover's quantum walk does not diverge much from the uniform starting state, thus not providing any advantage over classical search. We show the following results:

- We study quantum *stationary states* which approximate the starting state of the quantum walk. Stationary states are the eigenstates of the quantum walk operator with eigenvalue 1. We give a complete characterization of such states, revealing a connection to the oscillatory localization.

- We also study the conditions on the existence of such stationary states. We prove that if the marked vertices form a single non-bipartite connected component, then stationary states always exist regardless of the graph or the configuration of the marked vertices. We then show the sufficient and necessary conditions for the stationary states to exist if the marked states form a single connected bipartite component.

Query Complexity. In Part II, we study lower bound methods for the randomized query complexity. Randomized query algorithms have been shown to be more powerful than the deterministic ones. For example, they give speedups for such problems as local search [Aar06], monotonicity testing [GGLR98], recursive majority [MNSX11] etc. In Chapter 4, we give an introduction to the deterministic, randomized and quantum settings in the query model.

In Chapter 5, we study adversary lower bound methods for randomized query complexity. This family of bounds has originally been introduced in quantum computation and only then brought back to the classical setting [Aar06, LM04]. We show the following results:

- We prove that all known classical adversary bounds are equivalent for total functions. As a surprising fact, they are all equal to a well-known lower bound *fractional block sensitivity*. Along the way, we introduce a new classical adversary bound we call *rank-1 relational adversary bound*. The rank-1 adversary is also equivalent to the other adversary bounds and is easier to apply. This equivalence result simplifies the large palette of lower bounds available in randomized query complexity. It also shows that fractional block sensitivity is a fundamental measure for total functions with many different formulations.

- For partial functions, we show examples of functions where the adversary bounds are asymptotically different. Thus the equivalence described above does not hold in general case, in contrast with the quantum setting, where all adversary methods are equivalent even for partial functions [ŠS06] (except the general quantum adversary bound).

In Chapter 6, we study the relationship between fractional block sensitivity $\mathrm{fbs}(f)$ and block sensitivity $\mathrm{bs}(f)$. The latter is a lower bound even on the deterministic query complexity. Block sensitivity is also a crucial ingredient in proving polynomial relationship between deterministic, randomized and quantum query algorithms [BdW02]. Fractional block sensitivity has been used to prove optimal relationship between bounded-error and zero-error randomized query complexities [KT16].

The fractional block sensitivity is a generalization of the block sensivitiy and there are close connections between the two, for example, they tend to converge to each other under function composition [Tal13]. We prove the following results:

- We show that for any partial function on inputs of length n, $\mathrm{fbs}(f) \leq \sqrt{n \cdot \mathrm{bs}(f)}$. Thus there is a certain barrier to the lower bounds the fractional block sensitivity can obtain.

- We exhibit an explicit function for which $\mathrm{fbs}(f) = \Omega(n \cdot \mathrm{bs}(f))$ for any value of $\mathrm{bs}(f)$. Thus the upper bound from the previous result is tight and cannot be improved.

- We also show a slight improvement to the separation between $\mathrm{fbs}(f)$ and $\mathrm{bs}(f)$ for total functions.

Approval of the Results

The results of this thesis are published in the following papers. All of them have been indexed in Elsevier Scopus and Web of Science.

[APVW16] Andris Ambainis, Krišjānis Prūsis, Jevgēnijs Vihrovs, and Thomas G. Wong. Oscillatory localization of quantum walks analyzed by classical electric circuits. *Phys. Rev. A*, 94:062324, 2016.

Approximate contribution of the author: 55%.

[PVW16b] Krišjānis Prūsis, Jevgēnijs Vihrovs, and Thomas G. Wong. Stationary states in quantum walk search. *Phys. Rev. A*, 94:032334, 2016.

Approximate contribution of the author: 75%.

[AKPV18] Andris Ambainis, Martins Kokainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. All Classical Adversary Methods are Equivalent for Total Functions. In *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*, volume 96 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Approximate contribution of the author: 70%.

[APV18] Andris Ambainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. On block sensitivity and fractional block sensitivity. *Lobachevskii Journal of Mathematics*, 39(7):967––969, 2018.

Approximate contribution of the author: 70%.

These results have been presented by the author in the following international conferences and workshops:

- LU 75. zinātniskā konference, Riga, Latvia, 2017.

  Presentation: *Lokalizācija kvantu klejošanā.*

- Workshop on Quantum Computing at Kyoto University, Kyoto, Japan, 2017.

  Presentation: *Oscillatory Localization of Quantum Walks Analyzed by Classical Electric Circuits.*

- CCQ 2017 (Computation and Cryptography with Qu-bits), Kazan, Russia, 2017.

  Presentation: *On Block Sensitivity and Fractional Block Sensitivity.*

- Joint Estonian-Latvian Theory Days, Tartu, Estonia, 2017.

  Presentation: *All Classical Adversary Bounds are Equivalent for Total Functions.*

Apart from the papers relevant to this thesis, the author has contributed to the following papers during the course of the doctoral program:

[APV16] Andris Ambainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. Sensitivity versus certificate complexity of Boolean functions. In *Computer Science – Theory and Applications*, pages 16–28, Cham, 2016. Springer International Publishing.

[JKK+18] Rahul Jain, Hartmut Klauck, Srijita Kundu, Troy Lee, Miklos Santha, Swagato Sanyal, and Jevgēnijs Vihrovs. Quadratically tight relations for randomized query complexity. In *Computer Science – Theory and Applications*, pages 207–219, Cham, 2018. Springer International Publishing.

[BVW18] Balthazar Bauer, Jevgēnijs Vihrovs, and Hoeteck Wee. On the inner product predicate and a generalization of matching vector families. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2018)*, volume 122 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 41:1–41:13, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[ABI+19] Andris Ambainis, Kaspars Balodis, Jānis Iraids, Martins Kokainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. Quantum speedups for exponential-time dynamic programming algorithms. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1783–1793, 2019.

During the course of the doctoral program, the author has also given talks not directly relevant to this thesis at the following international conferences:

- CSR 2016 (the 11th International Computer Science Symposium in Russia), Saint Petersburg, Russia, 2016.

  Presentation: *Sensitivity Versus Certificate Complexity of Boolean Functions.*

- CSR 2018 (the 13th International Computer Science Symposium in Russia), Moscow, Russia, 2018.

  Presentation: *Quadratically Tight Relations for Randomized Query Complexity.*

- FSTTCS 2018 (38th Annual Conference on Foundations of Software Technology and Theoretical Computer Science), Ahmedabad, India, 2018.

  Presentation: *On the Inner Product Predicate and a Generalization of Matching Vector Families.*

Part I

Quantum Walks

Overview of Part I

In this part we study the properties of Grover's quantum walk on graphs, which we describe in Chapter 1. This kind of quantum walk has given provable speedups for many graph algorithms, e.g., traversing a graph or searching for a marked vertex. Hovewer, there are certain things it cannot speed up, which is the focus of the following chapters.

In Chapter 2, we examine the localization properties of Grover's quantum walk. It is a quantum phenomenon where the walk remains close to its starting state at all times during its evolution. As a consequence, the walk does not disperse throughout the graph as opposed to the classical random walk. We introduce the *oscillatory localization*, where the "walker" always alternates between two quantum states, remaining localized. This can be also formulated as examining eigenstates of the quantum walk operator squared with eigenvalue 1. First, we develop a complete characterization of such states. Then we connect them to the electric network theory, which gives us a tool to estimate the amount of localization for an arbitrary given starting state of the quantum walk. We also show examples of oscillatory localization on a variety of graphs.

In Chapter 3, we consider the limitations of Grover's quantum search. We study the configurations of the marked items on the graph where the quantum walk does not give any speedup over the classical search. Specifically, we examine quantum states which remain invariant under the quantum walk operator and are close to the starting state. For such states, Grover's quantum does not give any advantage over the trivial guessing, where the algorithm randomly picks a vertex and checks whether it is marked. Equivalently, we are looking at the eigenstates of the quantum search operator with eigenvalue 1, which we call *stationary states*. We completely characterize these states and then give a description of the stationary states closest to the starting state of the quantum walk. If one knows such a stationary state, then it gives the possibility to estimate the probability of finding the marked vertex under any number of steps. If the starting state is nearly the same as the closest stationary state, then the quantum walk does not give any advantage. Thus we give a description of a large family of configurations where Grover's quantum search is not efficient.

Chapter 1

Grover's Quantum Walk

In this chapter we give a brief introduction into the model of quantum computation and describe Grover's quantum walk on graphs. For a detailed introduction in quantum computation and information, see [Wat18].

1.1 Introduction

The model of quantum computation is a generalization of the classical probabilistic model of computation. It utilizes phenomena of quantum mechanics to obtain algorithmic speedups for many computational problems. For example, the groundbreaking algorithm of Shor can factorize an integer in polynomial time, while the best known classical algorithm runs in superpolynomial time [Sho97]. Another famous example is Grover's algorithm, which finds a marked item in the database of N items in $O(\sqrt{N})$ time, while classically any such algorithm needs $\Omega(N)$ time [Gro96]. Quantum algorithms can be employed in optimization, cryptography, simulation of physical systems, solving of linear systems and many other areas [Mon16].

1.2 Linear Algebra

First we briefly define relevant structures and notation from linear algebra.

Matrices. An $n \times m$ *matrix* A is an array of elements with n columns and m rows. For *square* matrices, $n = m$. The element in the i-th row and the j-th column is denoted by A_{ij}. We will work mainly with square complex matrices.

The *transpose* of a matrix A is a matrix A^{T} such that $A_{ij} = A^{\mathrm{T}}_{i,j}$ for all i, j. The *conjugate transpose* of a matrix A is a matrix A^* that is A transposed and all its elements replaced by their complex conjugates.

Example 1.

$$A = \begin{pmatrix} 0 & 1 \\ i & 0 \end{pmatrix} \qquad A^{\mathrm{T}} = \begin{pmatrix} 0 & i \\ 1 & 0 \end{pmatrix} \qquad A^* = \begin{pmatrix} 0 & -i \\ 1 & 0 \end{pmatrix}$$

We denote the $n \times n$ identity matrix by I_n. A matrix L is called *linear* if the following conditions hold:

1. $L(|\psi\rangle + |\varphi\rangle) = L|\psi\rangle + L|\varphi\rangle$.
2. $L(\alpha|\psi\rangle) = \alpha(L|\psi\rangle)$, for any scalar α.

A linear matrix U is called *unitary* if $UU^* = I_n$. A matrix H is called *Hermitian* if $H = H^*$.

Tensor product. The *tensor* or *Kronecker product* of an $n \times m$ matrix A with an $k \times l$ matrix B is defined as an $nk \times ml$ matrix

$$A \otimes B = \begin{pmatrix} A_{11}B & A_{12}B & \dots & A_{1m}B \\ A_{21}B & A_{22}B & \dots & A_{2m}B \\ \dots & \dots & \dots & \dots \\ A_{n1}B & A_{n2}B & \dots & A_{nm}B \end{pmatrix}$$

It has the following properties:

1. Distributivity: $A \otimes (B + C) = A \otimes B + A \otimes C$ and $(A + B) \otimes C = A \otimes C + B \otimes C$.
2. Associativity: $(A \otimes B) \otimes C = A \otimes (B \otimes C)$.
3. For any scalar α: $(\alpha A) \otimes B = A \otimes (\alpha B) = \alpha(A \otimes B)$.
4. The mixed-product property: $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$.

Vectors. A *vector* of length n is a $n \times 1$ matrix (also called a column vector). We will use the *braket* notation and write a vector ψ as $|\psi\rangle$. The conjugate transpose of $|\psi\rangle$ is denoted by $\langle\psi|$ (this is a row vector).

The *inner product* of two vectors $|\psi\rangle = (\psi_1, \dots, \psi_n)^{\mathrm{T}}$ and $|\varphi\rangle = (\varphi_1, \dots, \varphi_n)^{\mathrm{T}}$ is defined as a number

$$\langle\psi|\varphi\rangle = \sum_{i=1}^{n} \psi_i^* \varphi_i.$$

The *outer product* of $|\psi\rangle$ and $|\varphi\rangle$ is defined as $|\phi\rangle\langle\varphi|$ and is an $n \times n$ matrix.

The *Euclidean norm*, *ℓ_2-norm* or *length* of a vector $|\psi\rangle$ is defined as $\|\psi\| = \sqrt{\langle\psi|\psi\rangle}$ (it can be checked that $\langle\psi|\psi\rangle$ is always non-negative). A vector $|\psi\rangle$ is called a *unit* vector if its length is 1. Two vectors $|\psi\rangle$ and $|\varphi\rangle$ are called *orthogonal* if $\langle\psi|\varphi\rangle = 0$.

The n-dimensional *complex Hilbert space* is the set of all complex vectors of length n, complete with the inner product operation defined above. An *orthonormal basis* of this space is a set of n vectors $|\psi_1\rangle, |\psi_2\rangle, \dots, |\psi_n\rangle$ such that:

- All $|\psi_i\rangle$ are unit vectors.
- For any $i \neq j$, vectors $|\psi_i\rangle$ and $|\psi_j\rangle$ are orthogonal.

Eigenvectors. An *eigenvector* of an $n \times n$ matrix A with *eigenvalue* λ is a vector $|v\rangle$ such that

$$A|v\rangle = \lambda|v\rangle.$$

The pair (v, λ) is called an *eigenpair* of A.

A unitary $n \times n$ matrix U has n orthonormal eigenvectors

$$|\psi_1\rangle, |\psi_2\rangle, \dots, |\psi_n\rangle$$

with eigenvalues

$$e^{i\theta_1}, e^{i\theta_2}, \ldots, e^{i\theta_n},$$

for some $\theta_1, \theta_2, \ldots, \theta_n \in [0, 2\pi)$. Another way to see these eigenvectors is as a complete orthonormal basis of the n-dimensional Hilbert space.

1.3 Quantum Information

Qubits. Classical computers store information in binary states called *bits*: a bit is a state that is always either 0 or 1. On the other hand, quantum computers store information in *qubits*. A qubit is in a *superposition* of two classical states $|0\rangle$ and $|1\rangle$. It means that the state can be written as

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

for some complex $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$. α and β are called the *amplitudes* of the corresponding states. The states $|0\rangle$ and $|1\rangle$ are the basis of a two-dimensional complex space $\mathbb{C}^2$,

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

In general, a quantum computer can operate on a multiqubit system. An n-qubit quantum state is a unit vector in a 2^n-dimensional complex Hilbert space $\mathbb{C}^{2n}$. The vectors

$$|b_1 b_2 \ldots b_n\rangle = |b_1\rangle \otimes |b_2\rangle \otimes \ldots \otimes |b_n\rangle$$

form a complete orthonormal basis of this space. For example, the basis of a 2 qubit system is $\{|00\rangle , |01\rangle , |10\rangle , |11\rangle\}$.

Often a quantum algorithm operates only with n different classical states $|1\rangle , |2\rangle , \ldots , |n\rangle$. The quantum state then is a superposition $\sum_{i=1}^{n} \alpha_i |i\rangle$, for some complex α_i such that

$$|\alpha_1|^2 + |\alpha_2|^2 + \ldots + |\alpha_n|^2 = 1.$$

This can be implemented as a quantum system with $\lceil \log_2 n \rceil$ qubits (when i is written in binary, $|i\rangle$ gives the corresponding basis state). The set of states $\{|1\rangle , |2\rangle , \ldots , |n\rangle\}$ is also called the *standard basis* of the system.

Transformations. Quantum states can be modified using unitary transformations. If the quantum state $|\psi\rangle$ is n-dimensional, then an $n \times n$ unitary matrix U can be applied to transform it into the state $U |\psi\rangle$.

Unitary matrices satisfy the following properties:

1. They preserve the length of vectors: $\|U |\phi\rangle\| = \|\phi\|$ (preserve the ℓ_2 norm). Hence, a unitary transformation transforms a quantum state into another quantum state.

2. They preserve the inner product between vectors: $\langle \psi | \varphi \rangle = \langle U\psi | U\varphi \rangle$.

Measurement. If a measurement is performed on a qubit $\alpha|0\rangle + \beta|1\rangle$, then the state becomes $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$. It then is said that the state *collapses* into one of the two states.

If the quantum state is a superposition $\sum_{i=1}^{n} \alpha_i |i\rangle$, then after the measurement it collapses to $|i\rangle$ with probability $|\alpha_i|^2$. This is the simplest form of measurement, which is performed in the standard basis. There are more general types of measurement, but the measurement in the standard basis is enough for the purpose of this work.

The goal of a quantum algorithm is to carefully shape the quantum state using unitary transformations into a state where the probability of measuring a "good" state (which would give the answer to the problem) is much higher than measuring a "bad" state.

1.4 Grover's Quantum Walk

Quantum walk is a generalization of the classical random walk, in which a walker traverses a graph in a probabilistic way. Random walks can simulate various physical and mathematical processes and are widely used in the construction of efficient algorithms [Lov93].

In this section we describe a quantum walk algorithm based on Grover's algorithm for searching a marked item in a database. This approach was introduced in [AKR05].

1.4.1 Quantum Walk

Let $G = (V, E)$ be a graph. Let $N = |V|$ be the number of vertices in G. The quantum walk operates in a superposition over the orthonormal standard basis $\{|u\rangle \otimes |v\rangle \mid u, v \in V\}$. For simplicity, we will also write $|u\rangle \otimes |v\rangle$ also as $|uv\rangle$. Thus the system evolves in the N^2-dimensional complex Hilbert space.

During the course of Grover's quantum walk, the quantum state will have non-zero support only on the *edge states*, namely all $|uv\rangle$ such that $\{u, v\} \in E$. Note that for a single pair $\{u, v\}$, there are two distinct states, $|uv\rangle$ and $|vu\rangle$. Let $m = 2|E|$, then in fact the system evolves in the m-dimensional complex Hilbert subspace.

There are two components of Grover's quantum walk, the coin and the shift operations C and S. Both of them are $m \times m$ unitary transformations. One step of the quantum walk is given by first applying C, and S afterwards.

In the classical random walk, a walker is always located in some vertex of the graph. If the walker is at a vertex u, the probability of moving to some neighbour of u in the next step is equal to $1/\deg(u)$. Thus the process can be understood as the walker throwing a fair coin with $\deg(u)$ sides to decide the next location, and then moving (shifting) to this neighbour.

In the quantum setting, both transformations perform the same functions, but operate in a different way. Here, C is the Grover diffusion transformation [Gro96]. For a vertex u, define $|\psi_u\rangle$ to be the equal superposition of the edges to its neighbours,

$$|\psi_u\rangle = \frac{1}{\sqrt{\deg(u)}} \sum_{v \sim u} |uv\rangle ,$$

and let

$$I_u = \sum_{v \sim u} |uv\rangle \langle uv| .$$

Then we define C as

$$C = \sum_{u \in V} \left(2 \left|\psi_u\right\rangle \left\langle\psi_u\right| - I_u\right).$$

The shift operation simply swaps the endpoints of an edge,

$$S \left|uv\right\rangle = \left|vu\right\rangle.$$

Thus one step of the quantum walk is defined by the unitary

$$U = SC. \tag{1.1}$$

Example 2. Let us consider a single step of Grover's quantum walk on the *2-dimensional grid graph*. Suppose that the starting state of the walk is just $|ab\rangle$, for some adjacent vertices a and b. Let the neighbours of a be b, c, d, e. Then the coin transformation acts on the edge $|ab\rangle$ as

$$C \left|ab\right\rangle = -\frac{1}{2} \left|ab\right\rangle + \frac{1}{2} \left|ac\right\rangle + \frac{1}{2} \left|ad\right\rangle + \frac{1}{2} \left|ae\right\rangle.$$

The shift transformation then flips the directions of the edges,

$$U \left|\psi_0\right\rangle = SC \left|ab\right\rangle = -\frac{1}{2} \left|ba\right\rangle + \frac{1}{2} \left|ca\right\rangle + \frac{1}{2} \left|da\right\rangle + \frac{1}{2} \left|ea\right\rangle.$$

Consider the following task. We start in some vertex a of the graph and need to reach some other vertex b in the smallest time possible. In the random walk process, the time is equal to the number of steps made by the walker. The expected time is also called the *hitting time*. Now we describe the general quantum walk algorithm for this problem [Amb03b]. The quantum walk starts in a vertex a (in an equal superposition over the neighbours of u) and evolves by applying U. The time then is equal to the number of applications of U.

Algorithm 1 Quantum walk algorithm.

1. Prepare the starting state, the uniform superposition

$$\left|\psi_0\right\rangle = \frac{1}{\sqrt{\deg(u)}} \sum_{v \sim u} \left|uv\right\rangle.$$

2. Apply T steps of U for some choice of T,

$$\left|\psi_T\right\rangle = U^T \left|\psi_0\right\rangle.$$

3. Measure the state $|\psi_t\rangle$ and obtain some edge state $|uv\rangle$.

4. Check whether u is equal to b.

This algorithm leads to strong speedups of the hitting time. For example, this quantum walk gives an exponential speedup for a specific graph on $\Theta(2^d)$ vertices, where the quantum hitting time is $O(d^2)$ and the classical hitting time is $\Theta(2^d)$ [CFG02].

On the other hand, quantum walks sometimes exhibit the localization phenomenon, where the quantum state remains localized in the same position at all times during the process. This kind of behaviour is not present in random walks. In Chapter 2, we study the oscillatory localization of Grover's quantum walk, in which the system alternates between two quantum states.

1.4.2 Quantum Search

Consider the setting where some vertices of the graph are marked, and the goal is to find any of these vertices. Now we will use the oracle transformation Q that flips the sign of the edges state if the source vertex is marked,

$$\begin{aligned} Q\left|uv\right\rangle &= -\left|vu\right\rangle, && \text{if } u \text{ is marked,} \\ Q\left|vu\right\rangle &= \left|vu\right\rangle, && \text{if } u \text{ is not marked.} \end{aligned}$$

A single step of Grover's quantum search is given by

$$U = SCQ, \tag{1.2}$$

thus first we apply the oracle transformation and then one step of the quantum walk. Now we describe the general search algorithm:

Algorithm 2 Quantum search algorithm.

1. Prepare the starting state, the uniform superposition

$$\left|\psi_0\right\rangle = \frac{1}{\sqrt{2|E|}} \sum_{u\in V} \sum_{v\sim u} \left|uv\right\rangle. \tag{1.3}$$

2. Apply T steps of U for some choice of T,

$$\left|\psi_T\right\rangle = U^T \left|\psi_0\right\rangle.$$

3. Measure the state $\left|\psi_t\right\rangle$ and obtain some edge state $\left|uv\right\rangle$.

4. Check whether u is marked.

This algorithm gives a speedup for a search of a single marked item for a large variety of graphs. For example, this algorithm finds the marked item in $O(\sqrt{N})$ steps for the complete graph on N vertices (see, for example, [Won15] for analysis), $O(\sqrt{N \log N})$ steps for the $\sqrt{N} \times \sqrt{N}$ 2D grid [ABN+13]. Note that this quantum search does not always necessarily give a quantum speedup. If there are multiple marked vertices, there are even examples of configurations where this quantum search does not give any speedup over classical random guessing [NR16]. These kinds of configurations are the focus of Chapter 3.

Chapter 2

Oscillatory Localization

In this chapter we describe the results concerning the oscillatory localization of Grover's quantum walk. They are based on the following paper:

[APVW16] Andris Ambainis, Krišjānis Prūsis, Jevgēnijs Vihrovs, and Thomas G. Wong. Oscillatory localization of quantum walks analyzed by classical electric circuits. *Phys. Rev. A*, 94:062324, 2016.

2.1 Introduction

Localization was first observed in discrete-time quantum walks by Mackay et al. [MBSS02], whose numerical simulations demonstrated that an initially localized quantum walker on the two-dimensional (2D) lattice had a high probability of remaining at its initial location. This behavior was further investigated numerically by Tregenna et al. [TFMK03], and it was analytically proved to persist for all time by Inui, Konishi, and Konno [IKK04]. Since this seminal work, localization in quantum walks has been an area of thriving research (see Section 2.2.9 of [VA12] for an overview and the references therein). Such localization is a purely quantum phenomenon, starkly different from the diffusive behavior of classical random walks. Furthermore, the ability to localize a quantum walker has potential applications in quantum optics, quantum search algorithms, and investigating topological phases [KRBD10, KKJ15].

2.2 Overview of the Results

We introduce a new type of localization where the quantum walker jumps back and forth between two locations, which we term it *oscillatory localization*. The first hint of this behavior appears in Inui, Konishi, and Konno's aforementioned analysis of the 2D walk [IKK04], where the probability of finding the walker at its initial location is high at even times and small at odd times. Our analysis shows that this is due to the walker jumping back and forth between its initial location and an adjacent site, and in this work, we prove that it occurs on a wide variety of graphs, including complete graphs, complete bipartite graphs, hypercubes, square lattices of high dimension, expander graphs, and high degree graphs.

We study this type of localization with Grover's quantum walk on the graph, see Section 1.4. For simplicity, we assume that the graph is regular with degree d. In the Section 2.3, we give a simple example of oscillatory localization of the quantum walk on the complete

graph. The analysis is straightforward enough that the exact evolution can be determined using basic linear algebra. This forms intuition for more advanced analytical techniques, beginning in Section 2.4. There, we determine the eigenvectors of U^2 with eigenvalue 1. We show that there are only two different types, which we call uniform states and flip states, and they form a complete orthogonal basis for exact oscillatory states. So the projection of an arbitrary state onto these gives a lower bound on the extent of the oscillation, as shown in Section 2.5. While the projection of an arbitrary state onto uniform states is trivial, it is much more challenging to find its projection onto flip states.

So for the rest of the chapter, we develop a method for lower-bounding the projection onto flip states using classical electric networks in Section 2.6. To do this, we define a bijection between flip states and circulation flows in a related graph. Since electric current is a circulation flow, we prove that oscillations on a graph occurs if the power dissipation on a related electric network is low. Then we apply this framework to certain localized starting states, showing that effective resistance can be used instead of power dissipation. That is, low electric resistance implies oscillatory localization of these states of the quantum walk. Since effective resistance is inversely related to edge-connectivity, it follows that high edge-connectivity also implies localization for the particular starting states.

Finally, in Section 2.7, we apply this network formulation to several examples, proving that oscillatory localization occurs on a wide variety of regular graphs, including complete graphs, complete bipartite graphs, hypercubes, square lattices of high dimension, expander graphs, and high degree graphs.

Many connections between effective resistance and classical random walks are known, such as with hitting time, commute time, and cover time [DS84, Tet91, CRR+96]. For quantum walks, however, such connections are relatively new. Belovs et al. [BCJ+13] bounds the running time of a quantum walk algorithm for 3-Distinctness in terms of the resistance of a graph.

2.3 Localization on the Complete Graph

We begin with a simple example of oscillatory localization on the complete graph of N vertices, an example of which is shown in Fig. 2.1. As depicted in the figure, the walker is

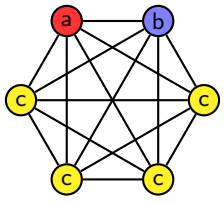


Figure 2.1: The complete graph of $N = 6$ vertices. The quantum walker begins at vertex a, pointing at vertex b. By symmetry, the remaining vertices, labeled c, evolve identically.

initially located at a single vertex, labeled a, and points towards another vertex, labeled b. That is, the initial state of the system is $|a\rangle \otimes |b\rangle$. By the symmetry of the quantum walk, all the other vertices will evolve identically; let us call them c vertices. Grouping them together,

we obtain a 7D subspace for the evolution of the system:

$$\begin{aligned}
|ab\rangle &= |a\rangle \otimes |b\rangle \\
|ac\rangle &= |a\rangle \otimes \frac{1}{\sqrt{N-2}} \sum_c |c\rangle \\
|ba\rangle &= |b\rangle \otimes |a\rangle \\
|bc\rangle &= |b\rangle \otimes \frac{1}{\sqrt{N-2}} \sum_c |c\rangle \\
|ca\rangle &= \frac{1}{\sqrt{N-2}} \sum_c |c\rangle \otimes |a\rangle \\
|cb\rangle &= \frac{1}{\sqrt{N-2}} \sum_c |c\rangle \otimes |b\rangle \\
|cc\rangle &= \frac{1}{\sqrt{N-2}} \sum_c |c\rangle \otimes \frac{1}{\sqrt{N-3}} \sum_{c' \sim c} |c'\rangle \,.
\end{aligned}$$

So the system begins in $|ab\rangle$. The system evolves by repeated applications of the quantum walk operator (1.1), which in the $\{|ab\rangle, |ac\rangle, |ba\rangle, |bc\rangle, |ca\rangle, |cb\rangle, |cc\rangle\}$ basis is

$$U = \begin{pmatrix}
0 & 0 & -\frac{N-3}{N-1} & \frac{2\sqrt{N-2}}{N-1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{N-3}{N-1} & \frac{2}{N-1} & \frac{2\sqrt{N-3}}{N-1} \\
-\frac{N-3}{N-1} & \frac{2\sqrt{N-2}}{N-1} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{2}{N-1} & -\frac{N-3}{N-1} & \frac{2\sqrt{N-3}}{N-1} \\
\frac{2\sqrt{N-2}}{N-1} & \frac{N-3}{N-1} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{2\sqrt{N-2}}{N-1} & \frac{N-3}{N-1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{2\sqrt{N-3}}{N-1} & \frac{2\sqrt{N-3}}{N-1} & \frac{N-5}{N-1}
\end{pmatrix}.$$

This matrix can be obtained by explicit calculation, or by using Eq. (9) of [PVW16a].

In Fig. 2.2, we plot the probability in $|ab\rangle$ (black circles) and $|ba\rangle$ (red squares) as the quantum walk evolves, and we see that the system roughly oscillates between the two states. This is an example of oscillatory localization. As the number of vertices N increases, the probability in each state at even and odd times goes to 1, so the oscillation becomes more and more certain. The precise numerical values for these probabilities are shown in Table 2.1, along with their corresponding amplitudes.

To prove that this oscillatory localization between $|ab\rangle$ and $|ba\rangle$ persists for all time, we express the initial state $|ab\rangle$ in terms of the eigenvectors and eigenvalues of U. The (unnormalized) eigenvectors and eigenvalues are

$$\psi_1 = \left(\frac{1}{N},\ 0,\ -\frac{N-3}{N(N-1)},\ \frac{2\sqrt{N-2}}{N(N-1)},\ \frac{2\sqrt{N-2}}{N(N-1)},\ 0,\ \frac{\sqrt{(N-2)(N-3)}}{N(N-1)}\right)^{\mathrm{T}}, \qquad 1$$

$$\psi_1' = \frac{(N-2)(N-3)}{2N(N-1)}\left(1,\ \frac{-1}{\sqrt{N-2}},\ -1,\ \frac{1}{\sqrt{N-2}},\ \frac{1}{\sqrt{N-2}},\ \frac{-1}{\sqrt{N-2}},\ 0\right)^{\mathrm{T}}, \qquad 1$$

$$\psi_{-1} = \frac{N-3}{2(N-1)}\left(1,\ \frac{-1}{\sqrt{N-2}},\ 1,\ \frac{-1}{\sqrt{N-2}},\ \frac{-1}{\sqrt{N-2}},\ \frac{-1}{\sqrt{N-2}},\ \frac{2}{\sqrt{(N-2)(N-3)}}\right)^{\mathrm{T}}, \qquad -1$$

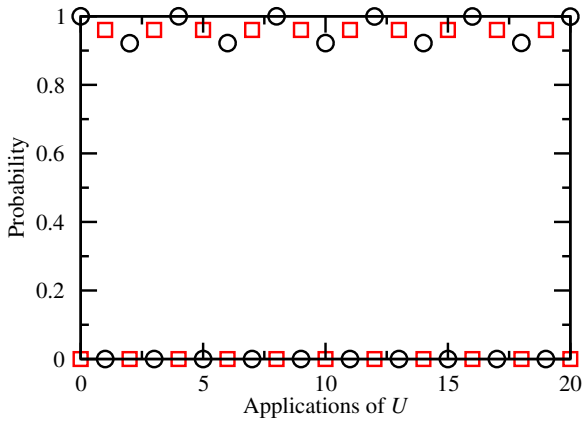


Figure 2.2: Quantum walk on the complete graph of $N = 16$ vertices, starting in $|ab\rangle$. The probability in $|ab\rangle$ (black circles) and $|ba\rangle$ (red squares) as the quantum walk U is applied.

$$\psi_+ = \frac{1}{N[(N-2) - i\sqrt{N(N-2)}]}\Big(1 - i\sqrt{N(N-2)},\ -i(N-3)\sqrt{N},\ N-3,$$
$$-2\sqrt{N-2},\ (N-2)^{3/2} + i\sqrt{N},\ 0,\ -\sqrt{N-3}(\sqrt{N-2} + i\sqrt{N})\Big)^{\mathrm{T}}, \qquad e^{i\theta}$$

$$\psi'_+ = \frac{N-3}{N[(N-2) - i\sqrt{N(N-2)}]}\Big(1,\ \frac{1}{2}\left(\sqrt{N-2} + i\sqrt{N}\right),\ -1,$$
$$\frac{-1}{2}\left(\sqrt{N-2} + i\sqrt{N}\right),\ \frac{-1}{2}\left(\sqrt{N-2} - i\sqrt{N}\right),\ \frac{1}{2}\left(\sqrt{N-2} - i\sqrt{N}\right),\ 0\Big)^{\mathrm{T}}, \qquad e^{i\theta}$$

$$\psi_- = \frac{1}{N[(N-2) + i\sqrt{N(N-2)}]}\Big(1 + i\sqrt{N(N-2)},\ i(N-3)\sqrt{N},\ N-3,$$
$$-2\sqrt{N-2},\ (N-2)^{3/2} - i\sqrt{N},\ 0,\ -\sqrt{N-3}(\sqrt{N-2} - i\sqrt{N})\Big)^{\mathrm{T}}, \qquad e^{-i\theta}$$

$$\psi'_- = \frac{N-3}{N[(N-2) + i\sqrt{N(N-2)}]}\Big(1,\ \frac{1}{2}\left(\sqrt{N-2} - i\sqrt{N}\right),\ -1,$$
$$\frac{-1}{2}\left(\sqrt{N-2} - i\sqrt{N}\right),\ \frac{-1}{2}\left(\sqrt{N-2} + i\sqrt{N}\right),\ \frac{1}{2}\left(\sqrt{N-2} + i\sqrt{N}\right),\ 0\Big)^{\mathrm{T}}, \qquad e^{-i\theta}$$

where

$$\cos\theta = \frac{-1}{N-1}, \quad \sin\theta = \frac{-\sqrt{N(N-2)}}{N-1}.$$

Note that $\theta \in (\pi, 3\pi/2)$. Alternatively, if it were defined in $(0, \pi/2)$, then the last four eigenvalues would have explicit minus signs.

It is straightforward to verify that the initial state $|ab\rangle = (1,0,0,0,0,0,0)^{\mathrm{T}}$ is the sum of

t	$\lvert\langle ab\rvert U^t\lvert ab\rangle\rvert^2$	$\lvert\langle ba\rvert U^t\lvert ab\rangle\rvert^2$	$\langle ab\rvert U^t\lvert ab\rangle$	$\langle ba\rvert U^t\lvert ab\rangle$
0	1	0	1	0
1	0	0.960004	0	−0.979798
2	0.921608	0	0.960004	0
3	6.52861×10^{-7}	0.960004	0.000807998	−0.979798
4	0.999967	0	0.999984	0
5	6.52329×10^{-7}	0.960004	−0.000807669	−0.979798
6	0.921671	0	0.960037	0
7	2.60825×10^{-6}	0.960004	0.00161501	−0.979798
8	0.999869	0	0.999935	0
9	2.60399×10^{-6}	0.960004	−0.00161369	−0.979798
10	0.921796	0	0.960102	0
11	5.855×10^{-6}	0.960004	0.00241971	−0.979798
12	0.999707	0	0.999853	0
13	5.84066×10^{-6}	0.960004	−0.00241675	−0.979798
14	0.921983	0	0.9602	0
15	0.0000103735	0.960004	0.00322079	−0.979798
16	0.999479	0	0.999739	0
17	0.0000103396	0.960004	−0.00321553	−0.979798
18	0.922233	0	0.96033	0
19	0.0000161359	0.960004	0.00401695	−0.979798
20	0.999187	0	0.999593	0

Table 2.1: For the quantum walk on the complete graph of $N = 16$ vertices, the probability and amplitude in $|ab\rangle$ and $|ba\rangle$.

these unnormalized eigenvectors, i.e.,

$$|ab\rangle = \psi_1 + \psi_1' + \psi_{-1} + \psi_+ + \psi_+' + \psi_- + \psi_-'.$$

Then the state of the system after t applications of U is

$$U^t\,|ab\rangle = \psi_1 + \psi_1' + (-1)^t\psi_{-1} + e^{i\theta t}\left(\psi_+ + \psi_+'\right) + e^{-i\theta t}\left(\psi_- + \psi_-'\right).$$

We can work out the amplitude of this in $|ab\rangle$ and $|ba\rangle$. Beginning with $|ab\rangle$:

$$\begin{aligned}
\langle ab|U^t|ab\rangle &= \frac{1}{N} + \frac{(N-2)(N-3)}{2N(N-1)} + (-1)^t\frac{N-3}{2(N-1)} + e^{i\theta t}\frac{1}{N} + e^{-i\theta t}\frac{1}{N} \\
&= \frac{N^2-3N+4}{2N(N-1)} + (-1)^t\frac{N-3}{2(N-1)} + \frac{2}{N}\cos(\theta t) \\
&= \begin{cases} \dfrac{N-2}{N} + \dfrac{2}{N}\cos(\theta t), & t \text{ even} \\ \dfrac{2}{N(N-1)} + \dfrac{2}{N}\cos(\theta t), & t \text{ odd} \end{cases}.
\end{aligned} \tag{2.1}$$

Now for $|ba\rangle$:

$$\langle ba|U^t|ab\rangle = -\frac{N-3}{N(N-1)} - \frac{(N-2)(N-3)}{2N(N-1)} + (-1)^t \frac{N-3}{2(N-1)} + 0 + 0$$
$$= \begin{cases} 0, & t \text{ even} \\ -\dfrac{N-3}{N-1}, & t \text{ odd} \end{cases}. \tag{2.2}$$

Using these formulas with $N = 16$, we get exactly the amplitudes in $|ab\rangle$ and $|ba\rangle$ in Table 2.1. Furthermore, for large N, we get that the amplitude in $|ab\rangle$ roughly alternates between 1 and 0, while the amplitude in $-|ba\rangle$ roughly alternates between 0 and 1. So for large N, the system alternates between being in $|ab\rangle$ and $|ba\rangle$ with probability nearly 1. This is an example of oscillatory localization, and it persists for all time.

2.4 Exact Oscillatory States

While the above example of oscillatory localization on the complete graph was simple enough to be exactly analyzed, for general graphs we expect such analysis to be intractable. Here we begin to develop a more general theory for determining when oscillatory localization occurs.

To start, we observe that oscillatory localization implies that the state of the system returns to itself after two applications of the quantum walk U (1.1). In other words, states that *exactly* oscillate are eigenvectors of U^2 with eigenvalue 1. In this section, we find these eigenstates, assuming that the graph G is d-regular, connected, and undirected. We show that there are only two distinct types of 1-eigenvectors of U^2, which we call uniform states and flip states, and that any 1-eigenvector of U^2 can be expressed in terms of them.

In general, although the 1-eigenvectors of U^2 oscillate between two states, they may not be localized. We disregard this for the time being, finding oscillatory states regardless of their spatial distributions. Later in the work, we project localized initial states, such as $|ab\rangle$ from the complete graph in the last section, onto these general oscillatory states and prove oscillatory localization on a large variety of graphs.

2.4.1 Uniform States

The first type of eigenvector of U^2 with eigenvalue 1 is a *uniform state*. To define it, we start by introducing some notation.

Definition 1. Let V be the vertex set of G. For a vertex subset $T \subseteq V$, define the outer product

$$|\sigma_T\rangle = |s_T\rangle \otimes |s_c\rangle ,$$

where

$$|s_T\rangle = \frac{1}{\sqrt{|T|}} \sum_{t \in T} |t\rangle \quad \text{and} \quad |s_c\rangle = \frac{1}{\sqrt{d}} \sum_{i=1}^{d} |i\rangle$$

are equal superpositions over the vertices and directions, respectively. Evaluating the tensor product,

$$|\sigma_T\rangle = \frac{1}{\sqrt{d|T|}} \sum_{\substack{t \in T \\ v \sim t}} |tv\rangle ,$$

where $|tv\rangle$ denotes the quantum walker at vertex t, pointing towards vertex v.

Using this notation $|\sigma_T\rangle$, we define the uniform states of G, depending on whether G is non-bipartite or bipartite:

Definition 2. The *uniform states of* G are

- $|\sigma_V\rangle$ if G is non-bipartite (i.e., the uniform superposition over all the vertices and directions).
- $|\sigma_X\rangle$, $|\sigma_Y\rangle$, and their linear combinations, if G is bipartite, where X and Y are the partite sets of G (i.e., the uniform superpositions over each partite set and all directions, and linear combinations of them).

Now we give a simple proof showing that uniform states, defined above, are indeed 1-eigenvectors of U^2.

Lemma 1. *The uniform states are eigenvectors of* U^2 *with eigenvalue 1.*

Proof. For an arbitrary $T \subseteq V$, we have

$$U|\sigma_T\rangle = S(I_N \otimes C)(|s_T\rangle \otimes |s_c\rangle) = S(|s_T\rangle \otimes C|s_c\rangle) = S(|s_T\rangle \otimes |s_c\rangle).$$

Let us consider separately the cases when G is non-bipartite or bipartite.

When G is non-bipartite, consider $|\sigma_V\rangle$. We have

$$U|\sigma_V\rangle = S(|s_V\rangle \otimes |s_c\rangle) = S\left(\frac{1}{2\sqrt{dN}} \sum_{\substack{v\in V \\ u\sim v}} (|uv\rangle + |vu\rangle)\right) = \frac{1}{2\sqrt{dN}} \sum_{\substack{v\in V \\ u\sim v}} (|vu\rangle + |uv\rangle) = |\sigma_V\rangle .$$

Therefore, $|s_V\rangle$ is an eigenvector of U with eigenvalue 1, and hence it is also an eigenvector of U^2 with eigenvalue 1.

Now suppose G is bipartite. For $|\sigma_X\rangle$, we have

$$U|\sigma_X\rangle = S(|s_X\rangle \otimes |s_c\rangle) = S\left(\frac{1}{\sqrt{dN/2}} \sum_{\substack{x\in X \\ y\sim x}} |xy\rangle\right) = \frac{1}{\sqrt{dN/2}} \sum_{\substack{y\in Y \\ x\sim y}} |yx\rangle = |\sigma_Y\rangle .$$

Thus in two steps, $|\sigma_X\rangle$ evolves to $|\sigma_Y\rangle$ and then back to $|\sigma_X\rangle$. Therefore, $|\sigma_X\rangle$ is an eigenvector of U^2 with eigenvalue 1. The same holds for $|\sigma_Y\rangle$. □

2.4.2 Flip States

The second type of eigenvector of U^2 with eigenvalue 1 is called a *flip state*. To define it, we first introduce the average amplitude of the edges pointing out from a vertex and into a vertex. Note that each undirected edge $\{u, v\}$ actually consists of two amplitudes: one for the walker at u pointing towards v, and one for the walker at v pointing towards u. So we can formulate each undirected edge as two directed edges (u, v) and (v, u) that are described by the states $|uv\rangle$ and $|vu\rangle$ [HBF03]. Then we have the following definition:

Definition 3. For a state $|\psi\rangle$, define the *average outgoing and incoming amplitudes* at a vertex u as

$$\overline{u_\psi^{\text{out}}} = \frac{1}{d}\sum_{v\sim u}\langle uv|\psi\rangle \quad \text{and} \quad \overline{u_\psi^{\text{in}}} = \frac{1}{d}\sum_{v\sim u}\langle vu|\psi\rangle\,.$$

These averages are important because the Grover coin C performs the "inversion about the average" of Grover's algorithm [Gro96], as explained in the following lemma:

Lemma 2. *Consider a general coin state $|\psi_c\rangle = \sum_{i=1}^d \alpha_i |i\rangle$. The Grover coin $C = 2|s_c\rangle\langle s_c| - I_d$ inverts each amplitude α_i about the average amplitude $\overline{\psi_c} = \frac{1}{d}\sum_{j=1}^d \alpha_j$, i.e.,*

$$C|\psi_c\rangle = \sum_{i=1}^d (2\overline{\psi_c} - \alpha_i)|i\rangle\,.$$

Additionally note that if the average amplitude is zero (i.e., $\overline{\psi_c} = 0$), then $C|\psi_c\rangle = -|\psi_c\rangle$.

Proof.

$$C|\psi_c\rangle = (2|s_c\rangle\langle s_c| - I_d)|\psi_c\rangle = 2|s_c\rangle\langle s_c|\psi_c\rangle - |\psi_c\rangle = 2\cdot\frac{1}{\sqrt{d}}\sum_{i=1}^d |i\rangle\frac{1}{\sqrt{d}}\sum_{j=1}^d \alpha_j - |\psi_c\rangle$$

$$= 2\left(\frac{1}{d}\sum_{j=1}^d \alpha_j\right)\sum_{i=1}^d |i\rangle - |\psi_c\rangle = 2\overline{\psi_c}\sum_{i=1}^d |i\rangle - |\psi_c\rangle = \sum_{i=1}^d (2\overline{\psi_c} - \alpha_i)|i\rangle\,.$$

□

Now a flip state can be defined in terms of these average outgoing and incoming amplitudes:

Definition 4. We say that a state $|\phi\rangle$ is a *flip state* if for each vertex u we have

$$\overline{u_\psi^{\text{out}}} = 0 \quad \text{and} \quad \overline{u_\psi^{\text{in}}} = 0. \tag{2.3}$$

In other words, the sum of the amplitudes of the edges starting at u is 0, and the sum of the amplitudes of the edges ending at u is 0, for every vertex in the graph.

To explain why this is called a flip state, we introduce the concept of negating and flipping the amplitudes between each pair of vertices:

Definition 5. For a state $|\psi\rangle$, define the *flipped state* $|\widetilde{\psi}\rangle$ to be such that

$$\langle uv|\widetilde{\psi}\rangle = -\langle vu|\psi\rangle$$

for any edge (u, v).

A short lemma and proof shows that flip states are equal to their flipped versions under action by U, hence their name:

Lemma 3. *Let $|\phi\rangle$ be a flip state. Then $U|\phi\rangle = |\widetilde{\phi}\rangle$.*

Proof. We have

$$U|\phi\rangle = S(I_N \otimes C)|\phi\rangle = S(I_N \otimes C)\sum_{(u,v)} \langle uv|\phi\rangle |uv\rangle = S\left[\sum_{(u,v)} \left(2\overline{u_\phi^{\text{out}}} - \langle uv|\phi\rangle\right)|uv\rangle\right]$$

$$= S\left[\sum_{(u,v)} -\langle uv|\phi\rangle |uv\rangle\right] = \sum_{(u,v)} -\langle uv|\phi\rangle |vu\rangle = |\widetilde{\phi}\rangle .$$

Note that the sum is over all directed edges (u, v), and the third line is obtained from Lemma 2 since the Grover coin inverts about the average. The fourth line is due to $|\phi\rangle$ being a flip state, so $\overline{u_\phi^{\text{out}}} = 0$. □

Note that the flipped version of a flip state is also a flip state:

Lemma 4. *Let $|\phi\rangle$ be a flip state. Then $|\widetilde{\phi}\rangle$ is also a flip state.*

Proof. For any vertex u, $\overline{u_{\widetilde{\phi}}^{\text{out}}} = -\overline{u_\phi^{\text{in}}} = 0$ and $\overline{u_{\widetilde{\phi}}^{\text{in}}} = -\overline{u_\phi^{\text{out}}} = 0$. □

Now it is straightforward to prove that flip states are 1-eigenvectors of U^2, so they oscillate between two states.

Lemma 5. *Any flip state $|\phi\rangle$ is an eigenvector of U^2 with eigenvalue 1.*

Proof. Since the flipped state $|\widetilde{\phi}\rangle$ is also a flip state by Lemma 4,

$$U^2|\phi\rangle = U|\widetilde{\phi}\rangle = |\phi\rangle$$

by Lemma 3. Therefore $|\phi\rangle$ is an eigenvector of U^2 with eigenvalue 1. □

2.4.3 Expansion of Eigenvectors

In the last two sections, we defined uniform states and flip states, both of which are eigenvectors of U^2 with eigenvalue 1. In this section, we work towards a theorem that proves that all 1-eigenvectors of U^2 can be written as linear combinations of uniform states and/or flip states. That is, uniform states and flip states form a complete basis for states exhibiting exact oscillations between two states. Towards this goal, we first prove some general properties of the 1-eigenvectors of U^2:

Lemma 6. *Let $|\psi\rangle$ be an eigenvector of U^2 with eigenvalue 1, and denote $|\psi'\rangle = U|\psi\rangle$. Suppose u and v are connected by an edge in G. Then*

$$\overline{u_\psi^{\text{out}}} = \overline{v_{\psi'}^{\text{out}}}.$$

Proof. Denote $\langle uv|\psi\rangle = \delta$. Let us examine the effect of two steps of the quantum walk (1.1) on this amplitude, as shown in Fig. 2.3.

After the first step, we have

$$\langle vu|\psi'\rangle = \langle vu|U|\psi\rangle = 2\overline{u_\psi^{\text{out}}} - \delta,$$

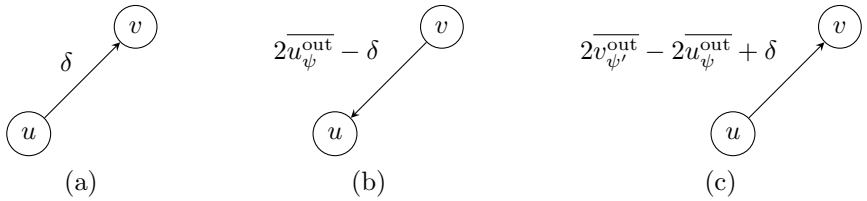


Figure 2.3: Evolution of a single amplitude shown in (a), after one application of U in (b), and two applications of U in (c).

since the Grover coin causes the amplitude to be inverted about the average (Lemma 2), and then the flip-flop shift causes the edge to switch directions. After the second step, we get

$$\langle uv|U^2|\psi\rangle = \langle uv|U|\psi'\rangle = 2\overline{v_{\psi'}^{\text{out}}} - \langle vu|\psi'\rangle = 2\overline{v_{\psi'}^{\text{out}}} - 2\overline{u_\psi^{\text{out}}} + \delta.$$

Since $|\psi\rangle$ is an eigenvector of U^2 with eigenvalue 1, we have $\langle uv|U^2|\psi\rangle = \langle uv|\psi\rangle$. Hence $2\overline{v_{\psi'}^{\text{out}}} - 2\overline{u_\psi^{\text{out}}} = 0$ and $\overline{v_{\psi'}^{\text{out}}} = \overline{u_\psi^{\text{out}}}$. □

Lemma 7. *Let $|\psi\rangle$ be an eigenvector of U^2 with eigenvalue 1. Suppose u and v are two vertices (not necessarily distinct) of G.*

(a) If there exists a walk of even length between u and v in G, then

$$\overline{u_\psi^{\text{out}}} = \overline{v_\psi^{\text{out}}}.$$

(b) If there exists a walk of odd length between u and v in G, then

$$\overline{u_\psi^{\text{out}}} = \overline{v_\psi^{\text{in}}}.$$

Proof. Denote $|\psi'\rangle = U|\psi\rangle$.

(a) Suppose $\{u,t\}$ and $\{t,w\}$ are edges of G. By Lemma 6,

$$\overline{u_\psi^{\text{out}}} = \overline{t_{\psi'}^{\text{out}}} = \overline{w_\psi^{\text{out}}}.$$

Then the statement holds by transitivity.

(b) Suppose there is an edge $\{u,t\}$. By Lemma 6,

$$\begin{aligned}
\overline{u_\psi^{\text{out}}} = \overline{t_{\psi'}^{\text{out}}} &= \frac{1}{d}\sum_{w\sim t}\langle tw|\psi'\rangle = \frac{1}{d}\sum_{w\sim t}\langle tw|S(I_N\otimes C)|\psi\rangle = \frac{1}{d}\sum_{w\sim t}\langle wt|(I_N\otimes C)|\psi\rangle \\
&= \frac{1}{d}\sum_{w\sim t}\left(2\overline{w_\psi^{\text{out}}} - \langle wt|\psi\rangle\right) = \frac{1}{d}\sum_{w\sim t}\left(2\overline{u_\psi^{\text{out}}} - \langle wt|\psi\rangle\right) \quad \text{by (a)} \\
&= 2\overline{u_\psi^{\text{out}}} - \frac{1}{d}\sum_{w\sim t}\langle wt|\psi\rangle = 2\overline{u_\psi^{\text{out}}} - \overline{t_\psi^{\text{in}}}.
\end{aligned}$$

Hence $\overline{u_\psi^{\text{out}}} = \overline{t_\psi^{\text{in}}}$. The statement holds by transitivity once again. □

With these lemmas in place, we are now able to prove the main result of this section, that exact oscillatory states are composed entirely of uniform states and/or flip states.

Theorem 8. *Let $|\psi\rangle$ be an eigenvector of U^2 with eigenvalue 1. Then for some flip state $|\phi\rangle$,*

(a) if G is non-bipartite,

$$|\psi\rangle = \alpha\,|\phi\rangle + \beta_V\,|\sigma_V\rangle\,,$$

(b) if G is bipartite,

$$|\psi\rangle = \alpha\,|\phi\rangle + \beta_X\,|\sigma_X\rangle + \beta_Y\,|\sigma_Y\rangle\,.$$

Proof. We prove each case separately:

(a) Let u and v be two arbitrary vertices of G. Since G is not bipartite, it contains a cycle of odd length. Therefore there exists both a walk of even length and a walk of odd length between u and v, as G is a connected graph. Thus $\overline{u_\psi^{\text{out}}} = \overline{v_\psi^{\text{out}}}$ and $\overline{u_\psi^{\text{out}}} = \overline{v_\psi^{\text{in}}}$ by Lemma 7. Therefore $\overline{v_\psi^{\text{out}}} = \overline{v_\psi^{\text{in}}} = \overline{\psi}$ for any vertex v.

Consider the (likely unnormalized) state $|\varphi\rangle = |\psi\rangle - |\sigma_V\rangle\langle\sigma_V|\psi\rangle$. We prove that, up to normalization, it is a flip state by checking the two conditions of (2.3). First, note that

$$\begin{aligned}\langle\sigma_V|\psi\rangle &= \sum_{u\in V}\sum_{v\sim u}\frac{1}{\sqrt{dN}}\langle uv|\psi\rangle = \sqrt{\frac{d}{N}}\sum_{u\in V}\sum_{v\sim u}\frac{1}{d}\langle uv|\psi\rangle \\ &= \sqrt{\frac{d}{N}}\sum_{u\in V}\overline{u_\psi^{\text{out}}} = \sqrt{\frac{d}{N}}N\overline{\psi} = \sqrt{dN}\,\overline{\psi}.\end{aligned} \tag{2.4}$$

On the other hand, $\langle uv|\sigma_V\rangle = \frac{1}{\sqrt{dN}}$ for any pair of adjacent vertices u and v. Hence,

$$\begin{aligned}\overline{u_\varphi^{\text{out}}} &= \frac{1}{d}\sum_{v\sim u}\langle uv|\varphi\rangle = \frac{1}{d}\sum_{v\sim u}\left(\langle uv|\psi\rangle - \langle uv|\sigma_V\rangle\langle\sigma_V|\psi\rangle\right) \\ &= \frac{1}{d}\sum_{v\sim u}\left(\langle uv|\psi\rangle - \frac{1}{\sqrt{dN}}\sqrt{dN}\,\overline{\psi}\right) = \frac{1}{d}\sum_{v\sim u}\left(\langle uv|\psi\rangle - \overline{\psi}\right) = \overline{u_\psi^{\text{out}}} - \overline{\psi} = 0.\end{aligned} \tag{2.5}$$

This is the first condition of (2.3). The second condition is proved similarly to equations (2.5):

$$\overline{u_\varphi^{\text{in}}} = \frac{1}{d}\sum_{v\sim u}(\langle vu|\psi\rangle - \langle vu|\sigma_V\rangle\langle\sigma_V|\psi\rangle) = \overline{u_\psi^{\text{in}}} - \overline{\psi} = 0.$$

So $|\varphi\rangle$ is a flip state.

(b) Let the partite sets be X and Y. Let $\overline{\psi_T} = \frac{1}{|T|}\sum_{t\in T}\overline{t_\psi^{\text{out}}}$. Suppose $x \in X$ and $y \in Y$. Then we have the following properties by Lemma 7:

$$\overline{x_\psi^{\text{out}}} = \overline{y_\psi^{\text{in}}} = \overline{\psi_X} \quad \text{and} \quad \overline{y_\psi^{\text{out}}} = \overline{x_\psi^{\text{in}}} = \overline{\psi_Y}.$$

Then similarly to (a), we can prove that the state $|\varphi\rangle = |\psi\rangle - |\sigma_X\rangle\langle\sigma_X|\psi\rangle - |\sigma_Y\rangle\langle\sigma_Y|\psi\rangle$ is a flip state by checking the two conditions of (2.3). First, note that similarly to equations (2.4),

$$\langle\sigma_X|\psi\rangle = \sum_{x\in X}\sum_{y\sim X}\frac{1}{\sqrt{dN/2}}\langle xy|\psi\rangle = \sqrt{dN/2}\,\overline{\psi_X}.$$

On the other hand, $\langle xy|\sigma_X\rangle = \frac{1}{\sqrt{dN/2}}$ for any edge (x, y). Moreover, $\langle xy|\sigma_Y\rangle = 0$. Hence,

$$\begin{aligned}\overline{x_\varphi^{\text{out}}} &= \frac{1}{d}\sum_{y\sim x}\left(\langle xy|\psi\rangle - \langle xy|\sigma_X\rangle\langle\sigma_X|\psi\rangle - \langle xy|\sigma_Y\rangle\langle\sigma_Y|\psi\rangle\right) \qquad (2.6)\\ &= \frac{1}{d}\sum_{y\sim x}\left(\langle xy|\psi\rangle - \frac{1}{\sqrt{dN/2}}\sqrt{dN/2}\,\overline{\psi_X} - 0\cdot\langle\sigma_Y|\psi\rangle\right)\\ &= \frac{1}{d}\sum_{y\sim x}\left(\langle xy|\psi\rangle - \overline{\psi_X}\right) = \overline{x_\psi^{\text{out}}} - \overline{\psi_X} = 0. \qquad (2.7)\end{aligned}$$

This is the first condition of (2.3). The second condition comes similarly to equations (2.6)–(2.7),

$$\overline{y_\varphi^{\text{in}}} = \frac{1}{d}\sum_{x\sim y}\left(\langle xy|\psi\rangle - \langle xy|\sigma_X\rangle\langle\sigma_X|\psi\rangle - \langle xy|\sigma_Y\rangle\langle\sigma_Y|\psi\rangle\right) = \overline{y_\psi^{\text{in}}} - \overline{\psi_X} = 0.$$

Similarly, we prove that $\overline{y_\varphi^{\text{out}}} = 0$ and $\overline{x_\varphi^{\text{in}}} = 0$. So $|\varphi\rangle$ is a flip state. □

We end this section by showing that uniform states and flip states are orthogonal to each other, so they serve as an orthonormal basis for 1-eigenvectors of U^2.

Lemma 9. *Any flip state $|\phi\rangle$ is orthogonal to any state $|\sigma_T\rangle$:*

Proof.

$$\langle\phi|\sigma_T\rangle = \sum_{t\in T}\sum_{v\sim t}\langle\phi|tv\rangle\langle tv|\sigma_T\rangle = \sum_{t\in T}\frac{1}{\sqrt{d|T|}}\sum_{v\sim t}\langle\phi|tv\rangle = \sum_{t\in T}\sqrt{\frac{d}{|T|}}\cdot\overline{t_\phi^{\text{out}}} = 0.$$

□

Also, the states $|\sigma_X\rangle$ and $|\sigma_Y\rangle$ are orthogonal because for any edge $|uv\rangle$, only one of them has a non-zero amplitude at this edge. Thus flip states and uniform states form a complete orthogonal basis for the eigenvectors of U^2 with eigenvalue 1.

2.5 Approximate Oscillatory States

In the previous section, we found the states that *exactly* exhibit oscillation between two quantum states, returning to themselves after two applications of U. We showed that these states are spanned by uniform states and flip states. In our example of oscillatory localization on the complete graph, however, we showed that the system *approximately* alternated between $|ab\rangle$ and $-|ba\rangle$. So, while these states are not exact 1-eigenvectors of U^2, they are "close enough" that oscillatory localization still occurs.

In this section, we give conditions for when a starting state $|\psi_0\rangle$ is "close enough" to being a 1-eigenvector of U^2 that it exhibits oscillations. We do this by expanding $|\psi_0\rangle$ as a linear combination of flip states, uniform states, and whatever state remains. If the overlap of $|\psi_0\rangle$ with the flip states and/or uniform states is sufficiently large, then oscillations occur. That is, the state approximately alternates between $|\psi_0\rangle$ at even steps and $U|\psi_0\rangle$ at odd steps.

2.5.1 Estimate on the Oscillations

The following theorem gives a bound on the extent of oscillations for an arbitrary starting state.

Theorem 10. *Let $|\psi_0\rangle$ be the starting state of the quantum walk. It can be expressed as*

$$|\psi_0\rangle = \alpha\,|\phi\rangle + \beta\,|\sigma\rangle + \gamma\,|\rho\rangle\,,$$

where

- *$|\phi\rangle$ is a normalized flip state;*
- *$|\sigma\rangle$ is a normalized uniform state, equal to $|\sigma_V\rangle$ if G is non-bipartite, or a normalized linear combination of $|\sigma_X\rangle$ and $|\sigma_Y\rangle$ if G is bipartite;*
- *$|\rho\rangle$ is some normalized "remainder" state orthogonal to $|\phi\rangle$ and $|\sigma\rangle$.*

Then

(a) after an even number of steps $2t$,

$$\left|\langle\psi_0|U^{2t}|\psi_0\rangle\right| \geq 2\left(|\alpha|^2 + |\beta|^2\right) - 1,$$

(b) after an odd number of steps $2t+1$,

$$\left|\langle\widetilde{\psi_0}|U^{2t+1}|\psi_0\rangle\right| \geq 2\max\left(|\alpha|^2, |\beta|^2\right) - 1.$$

Proof. We prove each part separately.

(a) After $2t$ steps, the state of the quantum walk is

$$U^{2t}\,|\psi_0\rangle = \alpha\,|\phi\rangle + \beta\,|\sigma\rangle + \gamma U^{2t}\,|\rho\rangle\,.$$

Since unitary operators preserve the inner product between vectors, we have $\langle\phi|U^{2t}|\rho\rangle = \langle\sigma|U^{2t}|\rho\rangle = 0$. Therefore,

$$\langle\psi_0|U^{2t}|\psi_0\rangle = |\alpha|^2 + |\beta|^2 + |\gamma|^2\,\langle\rho|U^{2t}|\rho\rangle\,.$$

Then we have

$$\left|\langle\psi_0|U^{2t}|\psi_0\rangle\right| \geq |\alpha|^2 + |\beta|^2 - |\gamma|^2 = |\alpha|^2 + |\beta|^2 - (1 - |\alpha|^2 - |\beta|^2) = 2\left(|\alpha|^2 + |\beta|^2\right) - 1.$$

(b) After $2t+1$ steps the state of the quantum walk is

$$U^{2t+1}\,|\psi_0\rangle = \alpha U^{2t+1}\,|\phi\rangle + \beta U^{2t+1}\,|\sigma\rangle + \gamma U^{2t+1}\,|\rho\rangle = \alpha\,|\widetilde{\phi}\rangle + \beta U\,|\sigma\rangle + \gamma U^{2t+1}\,|\rho\rangle\,.$$

For a non-bipartite graph,

$$U\,|\sigma\rangle = U\,|\sigma_V\rangle = |\sigma_V\rangle = -\,|\widetilde{\sigma_V}\rangle = -\,|\widetilde{\sigma}\rangle\,.$$

For a bipartite graph,

$$U\,|\sigma\rangle = U(p\,|\sigma_X\rangle + q\,|\sigma_Y\rangle) = p\,|\sigma_Y\rangle + q\,|\sigma_X\rangle = -p\,|\widetilde{\sigma_X}\rangle - q\,|\widetilde{\sigma_Y}\rangle = -\,|\widetilde{\sigma}\rangle\,.$$

Hence, in either case,

$$U^{2t+1}\left|\psi_0\right\rangle = \alpha\left|\widetilde{\phi}\right\rangle - \beta\left|\widetilde{\sigma}\right\rangle + \gamma U^{2t+1}\left|\rho\right\rangle.$$

The flipped starting state is

$$\left|\widetilde{\psi_0}\right\rangle = \alpha\left|\widetilde{\phi}\right\rangle + \beta\left|\widetilde{\sigma}\right\rangle + \left|\widetilde{\rho}\right\rangle.$$

To obtain the value of $\langle\widetilde{\psi_0}|U^{2t+1}|\psi_0\rangle$, we look at the inner products of the vectors that contribute to $|\widetilde{\psi_0}\rangle$ and $|\psi_0\rangle$. Since unitary operations preserve the inner product between vectors, we have that $\langle\widetilde{\phi}|U^{2t+1}|\rho\rangle = \langle\phi|\rho\rangle = 0$ and $\langle\widetilde{\sigma}|U^{2t+1}|\rho\rangle = \langle\sigma|\rho\rangle = 0$. Note that the "flip" transformation that takes $|\psi\rangle$ to $|\widetilde{\psi}\rangle$ is unitary. Hence $\langle\widetilde{\rho}|\widetilde{\phi}\rangle = \langle\rho|\phi\rangle = 0$ and $\langle\widetilde{\rho}|\widetilde{\sigma}\rangle = \langle\rho|\sigma\rangle = 0$. Therefore,

$$\langle\widetilde{\psi_0}|U^{2t+1}|\psi_0\rangle = |\alpha|^2 - |\beta|^2 + |\gamma|^2\,\langle\widetilde{\rho}|U^{2t}|\rho\rangle.$$

Thus we have

$$\begin{aligned}\left|\langle\widetilde{\psi_0}|U^{2t+1}|\psi_0\rangle\right| &\geq \left||\alpha|^2 - |\beta|^2\right| - |\gamma|^2 \\ &= \left||\alpha|^2 - |\beta|^2\right| - (1 - |\alpha|^2 - |\beta|^2) \\ &= 2\max\left(|\alpha|^2, |\beta|^2\right) - 1.\end{aligned}$$

□

2.5.2 Conditions for Oscillations

With this theorem, we can determine the conditions on $|\alpha|^2$ and $|\beta|^2$ for oscillations between two states to occur. If $|\alpha|^2 + |\beta|^2 > \frac{1}{2}$, then after any number of even steps the quantum walk is in $|\psi_0\rangle$ with probability $\Theta(1)$. This occurs because the starting state is closer to some eigenvector of U^2 with eigenvalue 1 than any other eigenvector with a different eigenvalue. At odd steps, the quantum walk is in $|\widetilde{\psi_0}\rangle$ with probability $\Theta(1)$ if either $|\alpha|^2 > \frac{1}{2}$ or $|\beta|^2 > \frac{1}{2}$. Thus in this case, the starting state should be very close to either a flip or a uniform state.

The value of β is easy to explicitly calculate. If the graph is non-bipartite, then the only uniform state is $|\sigma_V\rangle$, so $\beta = \langle\sigma_V|\psi_0\rangle$. If the graph is bipartite, then there are two uniform basis states $|\sigma_X\rangle$ and $|\sigma_Y\rangle$, so $|\beta|^2 = |\beta_X|^2 + |\beta_Y|^2$, where $\beta_X = \langle\sigma_X|\psi_0\rangle$ and $\beta_Y = \langle\sigma_Y|\psi_0\rangle$.

In general, the value of $|\alpha|^2$ is more difficult to find since the size of the basis for the flip states can be large. In the next section, however, we prove that $|\alpha|^2$ can be lower bounded using power dissipation, effective resistance, and connectivity of a related classical electric circuit. Since the quantum walk alternates between $|\psi_0\rangle$ and $|\widetilde{\psi_0}\rangle$ with probability $\Theta(1)$ if $|\alpha|^2 > \frac{1}{2}$, these quantities of classical circuits can be used to inform whether the quantum walk oscillates between two states.

2.6 Oscillations Using Electric Circuits

In this section, we describe the connection between the flip states of the quantum walk on a graph and electric networks. That gives us a tool to estimate the amount of the localization, given a particular starting state.

2.6.1 Network Flows and Flip States

We show that there is a close connection between the flip states of a quantum walk on a graph and the flows in a related network, of which electric current is a special case. Recall that a quantum walk has two amplitudes on each edge $\{u, v\}$, one from vertex u going to vertex v and another from v going to u. So to associate this to a network flow or current, we need to split up these two amplitudes. This can be done using the *bipartite double graph* from graph theory [BCN89]. Given a graph G, its bipartite double graph G_b is constructed as follows. For each vertex v in G, there are two vertices v_{out} and v_{in} in G_b. For each edge $\{u, v\}$ in G, there are two edges $\{u_{\text{out}}, v_{\text{in}}\}$ and $\{v_{\text{out}}, u_{\text{in}}\}$ in G_b, connected as shown in Fig. 2.4. As an example, consider the complete graph of three vertices in Fig. 2.5a. Applying this doubling procedure, we get its bipartite double graph in Fig. 2.5b. Note that this bipartite double graph is a cycle, which we make evident by rearranging the graph in Fig. 2.5c.

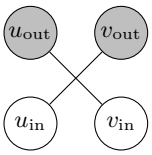
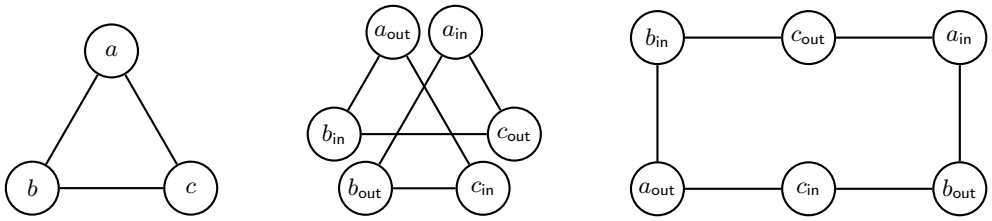


Figure 2.4: The counterpart of an edge $\{u, v\}$ of G in the bipartite double graph G_b.



(a) (b) (c)

Figure 2.5: (a) K_3, the complete graph of three vertices. (b) The bipartite double graph of K_3. (c) Rearrangement of the bipartite double graph of K_3.

Now let us define a network flow in a graph, which is another concept from graph theory [Bol79]. Let G be a graph and $E(G)$ its edge set. Denote by $\vec{E}(G) = \{(u, v) \mid \{u, v\} \in E(G)\}$ the set of directed edges of G. A *network flow* is a function $f : \vec{E}(G) \to \mathbb{C}$ that assigns a certain amount of flow passing through each edge [Bol79]. A network flow is called a *circulation* if it satisfies two properties: (a) Skew symmetry: the flow on an edge (u, v) is equal to the negative flow on the reversed edge (v, u), *i.e.*, $f(u, v) = -f(v, u)$, and (b) Flow conservation: the amount of the incoming flow at a vertex v is equal to the amount of the flow outgoing from v, *i.e.*, $\sum_{u:u\sim v} f(u, v) = 0$.

Each circulation f of G_b maps to a (possibly unnormalized) flip state $|\phi'\rangle$ of G (whose normalized form we denote by $|\phi\rangle$), and vice versa, by the following bijection:

$$f(u_{\text{out}}, v_{\text{in}}) = \langle uv|\phi'\rangle\,, \quad f(v_{\text{in}}, u_{\text{out}}) = -\langle uv|\phi'\rangle\,. \tag{2.8}$$

The amplitudes of the outgoing edges of u sum up to 0, thus the flow is conserved at vertex u_{out}. Similarly flow conservation holds also at vertex u_{in}.

Now consider an arbitrary starting state $|\psi_0\rangle$. To find the value of $|\alpha|^2$, we need the flip state $|\phi\rangle$ that is the closest to $|\psi_0\rangle$ among all the flip states. Alternatively, we can search for an optimal circulation in G_b. Next we show that a circulation that is sufficiently close to optimal can be obtained using electric networks.

2.6.2 Electric Networks and Oscillations

We examine *electric networks* [DS84], which are graphs where each edge is replaced by a unit resistor. Each vertex of the graph may also be either a source or a sink of some amount of current. We construct an electric network $\mathcal{N}_b$ from G_b and $|\psi_0\rangle$ in the following fashion. The vertex set of $\mathcal{N}_b$ is equal to that of G_b. Examine the amplitude $\langle uv|\psi_0\rangle = \delta$ at an edge $|uv\rangle$.

(a) If $\delta = 0$, add an edge $\{u_{\text{out}}, v_{\text{in}}\}$ to $\mathcal{N}_b$ with a unit resistance assigned.

(b) If $\delta \neq 0$, inject δ units of current at v_{in} and extract the same amount at u_{out}.

Note that $\mathcal{N}_b$ could have multiple sources and sinks of current by the construction.

For example, let G be a complete graph of three vertices a, b, c, which we considered in Fig. 2.5a. Its double bipartite graph G_b is a cycle of length 6, as shown in Fig. 2.5c. Say the starting state of the walk is $|\psi_0\rangle = |ab\rangle$. Then the electric network $\mathcal{N}_b$ is a path of length 5, as the edge $\{a_{\text{out}}, b_{\text{in}}\}$ is excluded from the electric network, as shown in Fig. 2.6. A unit current is then injected at b_{in} and extracted at a_{out}.

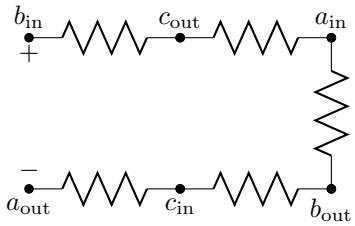


Figure 2.6: The electric network $\mathcal{N}_b$ for the complete graph of three vertices a, b, c with starting state $|\psi_0\rangle = |ab\rangle$.

Let $I_b(x, y)$ be the current flowing on the edge from x to y in $\mathcal{N}_b$. From this current, we can construct a circulation f of G_b. Consider an edge $\{u_{\text{out}}, v_{\text{in}}\}$ in G_b.

(a) If $\{u_{\text{out}}, v_{\text{in}}\} \in E(\mathcal{N}_b)$, set

$$f(u_{\text{out}}, v_{\text{in}}) = -f(v_{\text{in}}, u_{\text{out}}) = I_b(u_{\text{out}}, v_{\text{in}}).$$

(b) Otherwise, set

$$f(u_{\text{out}}, v_{\text{in}}) = -f(v_{\text{in}}, u_{\text{out}}) = \langle uv|\psi_0\rangle\,.$$

By construction, f satisfies skew symmetry. Since the net current flowing into a vertex equals the net current leaving it, f also satisfies flow conservation, so it is a circulation.

By the bijection (2.8), this circulation f corresponds to a flip state $|\phi'\rangle$, which in general is unnormalized. Since each amplitude in $|\phi'\rangle$ is either an amplitude in $|\psi_0\rangle$ or a current in $\mathcal{N}_b$, we have

$$\sum_{(u,v)} |\langle uv|\phi'\rangle|^2 = \sum_{(u,v)} |\langle uv|\psi_0\rangle|^2 + \sum_{\{x,y\}\in E(\mathcal{N}_b)} |I_b(x, y)|^2.$$

Since the starting state $|\psi_0\rangle$ is normalized, the first sum on the right-hand side is 1. For the rightmost sum, note the power dissipation through a resistor is I^2R [Gia08], where I is the current through the resistor and R its resistance. In our network, $R = 1$ since we have unit resistors, so the power dissipation is I^2. Then the rightmost sum is equal to the power dissipation $P(\mathcal{N}_b)$ in the network, i.e.,

$$P(\mathcal{N}_b) = \sum_{\{x,y\}\in E(\mathcal{N}_b)} |I_b(x,y)|^2.$$

Therefore,

$$\sum_{(u,v)} |\langle uv|\phi'\rangle|^2 = 1 + P(\mathcal{N}_b).$$

It is also possible that the current cannot flow in $\mathcal{N}_b$, depending on the starting state $|\psi_0\rangle$. For example, there is no flow if G is the complete graph of three vertices with $|\psi_0\rangle = \frac{1}{\sqrt{2}}(|ab\rangle + |ba\rangle)$. In this case, we regard the power dissipation as being infinitely large.

Normalizing $|\phi'\rangle$ and calling it $|\phi\rangle$,

$$|\phi\rangle = \frac{1}{\sqrt{1 + P(\mathcal{N}_b)}} |\phi'\rangle .$$

We are interested in how close this flip state $|\phi\rangle$ is to the starting state $|\psi_0\rangle$. This is given by the inner product

$$\begin{aligned}\langle\psi_0|\phi\rangle &= \frac{1}{\sqrt{1 + P(\mathcal{N}_b)}} \langle\psi_0|\phi'\rangle = \frac{1}{\sqrt{1 + P(\mathcal{N}_b)}} \sum_{(u,v)} \langle\psi_0|uv\rangle \langle uv|\phi'\rangle \\ &= \frac{1}{\sqrt{1 + P(\mathcal{N}_b)}} \sum_{(u,v)} |\langle\psi_0|uv\rangle|^2 = \frac{1}{\sqrt{1 + P(\mathcal{N}_b)}}.\end{aligned}$$

To go from the second to third line, recall when $\langle uv|\psi_0\rangle = \delta \neq 0$, then δ units of current are injected at v_{in} and extracted from u_{out}. Then $f(u_{\text{out}}, v_{\text{in}}) = \delta$, and from the bijection (2.8), $\langle uv|\phi'\rangle = \delta = \langle uv|\psi_0\rangle$. For the last line, $|\psi_0\rangle$ is a normalized state. Hence the contribution of the flip states to the starting state is lower bounded by

$$|\alpha|^2 \geq |\langle\psi_0|\phi\rangle|^2 = \frac{1}{1 + P(\mathcal{N}_b)}. \tag{2.9}$$

This lower bound is maximized when the power dissipation is minimized. Thomson's Principle states that the values of the current determined by Kirchhoff's Circuit Laws minimize the power dissipation. Hence the current through the network $\mathcal{N}_b$ naturally yields the best bound in (2.9).

Together with Theorem 10, this gives the following estimate on the oscillations: after an even number of steps $2t$ or an odd number of steps $2t + 1$,

$$\left|\langle\psi_0|U^{2t}|\psi_0\rangle\right|, \left|\langle\widetilde{\psi_0}|U^{2t+1}|\psi_0\rangle\right| \geq 2|\alpha|^2 - 1 \geq 2 \cdot \frac{1}{1 + P(\mathcal{N}_b)} - 1 = \frac{1 - P(\mathcal{N}_b)}{1 + P(\mathcal{N}_b)}. \tag{2.10}$$

This leads to our first and most general result linking electric circuits and oscillations:

Theorem 11. *For an arbitrary starting state $|\psi_0\rangle$, low power dissipation $P(\mathcal{N}_b) < 1$ implies oscillations between $|\psi_0\rangle$ and $|\widetilde{\psi_0}\rangle$.*

In general, the power dissipation of a circuit may be hard to find. But for some initial states, we can relate it to effective resistance, which is often easier to find and is well-studied for a number of graphs.

2.6.3 Oscillatory Localization of Single-Edge States

Consider the starting state $|\psi_0\rangle = |ab\rangle$, where the walker is initially localized at vertex a and points towards vertex b. Then the current in $\mathcal{N}_b$ is a unit flow (flow of one unit from b_{in} to a_{out}), and thus $P(\mathcal{N}_b) = R(\mathcal{N}_b)$ where $R(\mathcal{N})$ denotes the effective resistance of $\mathcal{N}$.

A related notion is the *resistance distance* $\Omega_{a,b}$ between two vertices a and b in a graph [KR93]. It is the effective resistance between a and b in a network obtained from the graph by replacing each edge by a unit resistor.

The only edge in which G_b and $\mathcal{N}_b$ differ is $\{a_{\text{out}}, b_{\text{in}}\}$, so we may interpret G_b as an electric network where $\{a_{\text{out}}, b_{\text{in}}\}$ with a unit resistance and $\mathcal{N}_b$ are connected in parallel. Hence low resistance distance $\Omega_{a_{\text{out}},b_{\text{in}}}$ in G_b implies low $R(\mathcal{N}_b)$:

$$\Omega_{a_{\text{out}},b_{\text{in}}} = \frac{1}{\frac{1}{1} + \frac{1}{R(\mathcal{N}_b)}} = 1 - \frac{1}{1 + R(\mathcal{N}_b)}, \tag{2.11}$$

where we used the parallel resistance formula [Gia08]. Substituting in (2.10), we obtain

$$\left|\langle\psi_0|U^{2t}|\psi_0\rangle\right|, \left|\langle\widetilde{\psi_0}|U^{2t+1}|\psi_0\rangle\right| \geq 1 - 2\Omega_{a_{\text{out}},b_{\text{in}}}. \tag{2.12}$$

Since $|\widetilde{\psi_0}\rangle = -|ba\rangle$, we have proven that

Theorem 12. *Low resistance distance $\Omega_{a_{\text{out}},b_{\text{in}}} < 1/2$ in G_b implies oscillatory localization between $|\psi_0\rangle = |ab\rangle$ and $|\widetilde{\psi_0}\rangle = -|ba\rangle$.*

Notably, if G is bipartite, then G_b consists of two identical copies of G. It follows that $\Omega_{a,b} = \Omega_{a_{\text{out}},b_{\text{in}}}$ because the current may flow from a_{out} to b_{in} in only one of the copies. Then low effective resistance in the original graph G is sufficient to imply localization of $|ab\rangle$.

2.6.4 Localization of Self-Flip States

For a particular set of starting states, we can essentially repeat the same analysis with the original graph G, not its bipartite double graph G_b. We say that $|\psi\rangle$ is a *self-flip state* if $\langle uv|\psi\rangle = -\langle vu|\psi\rangle$ for each edge $|uv\rangle$. That is, $|\psi\rangle = |\widetilde{\psi}\rangle$. Then a self-flip state undergoing oscillations is stationary, since it alternates between $|\psi\rangle$ and $|\widetilde{\psi}\rangle = |\psi\rangle$.

Suppose the initial state $|\psi_0\rangle$ is a self-flip state. We construct an electric network $\mathcal{N}$ with the same vertex set as G. Examine each *undirected edge* $\{u, v\}$ once; let $\langle uv|\psi_0\rangle = \delta$.

(a) If $\delta = 0$, add an edge $\{u, v\}$ to $\mathcal{N}$ with a unit resistance assigned.

(b) If $\delta \neq 0$, inject δ units of current at v and extract the same amount at u.

Let $I(u, v)$ again be the current flowing on the edge from u to v in $\mathcal{N}$. Since $\langle uv|\psi_0\rangle = -\langle vu|\psi_0\rangle$ and $I(u, v) = -I(v, u)$, we can construct a circulation f of G similarly to the previous construction of f from I_b. For an edge $\{u, v\}$ in G:

(a) If $\{u, v\} \in E(\mathcal{N})$, set
$$f(u, v) = -f(v, u) = I(u, v).$$

(b) Otherwise, set
$$f(u, v) = -f(v, u) = \langle uv|\psi_0\rangle .$$

Let $|\phi'\rangle$ be the unnormalized flip state corresponding to f according to (2.8). Again, since the amplitude on each edge $|uv\rangle$ in $|\phi'\rangle$ is either $I(u,v)$ or $\langle uv|\psi_0\rangle$, we have

$$\sum_{(u,v)} |\langle uv|\phi'\rangle|^2 = \sum_{(u,v)} |\langle uv|\psi_0\rangle|^2 + 2\sum_{\{x,y\}\in E(\mathcal{N})} |I(x,y)|^2.$$

The last sum has a factor of 2 as each edge $\{u,v\}$ from $\mathcal{N}$ sets the amplitudes of both $|uv\rangle$ and $|vu\rangle$ by the construction of f. Thus the expression is equal to $1+2P(\mathcal{N})$. By the same procedure as before, we use this to find the normalized state $|\phi\rangle$, the overlap $\langle\psi_0|\phi\rangle$, and a lower bound on $|\alpha|^2$, yielding a lower bound on localization:

$$\left|\langle\psi_0|U^{2t}|\psi_0\rangle\right|, \left|\langle\widetilde{\psi_0}|U^{2t+1}|\psi_0\rangle\right| \geq \frac{1-2P(\mathcal{N})}{1+2P(\mathcal{N})}. \tag{2.13}$$

This is similar to (2.10), and it yields the next result:

Theorem 13. *For a self-flip starting state $|\psi_0\rangle$, low power dissipation $P(\mathcal{N}) < 1/2$ implies that $|\psi_0\rangle$ is stationary.*

We can relate this to effective resistance for the particular starting state $|\psi_0\rangle = (|ab\rangle - |ba\rangle)/\sqrt{2}$, where the particle is localized at vertices a and b. By constructing $\mathcal{N}$ as described before, we obtain a current of $1/\sqrt{2}$ units from a to b. Then the power dissipation is equal to this current squared times the effective resistance, so $2P(\mathcal{N}) = R(\mathcal{N})$. Then similarly to (2.11), we deduce that

$$\Omega_{a,b} = \frac{1}{\frac{1}{1}+\frac{1}{R(\mathcal{N})}} = 1 - \frac{1}{1+R(\mathcal{N})}.$$

Therefore, substituting this in (2.13), we get

$$\left|\langle\psi_0|U^{2t}|\psi_0\rangle\right|, \left|\langle\widetilde{\psi_0}|U^{2t+1}|\psi_0\rangle\right| \geq 1 - 2\Omega_{a,b}.$$

This proves another new result:

Theorem 14. *Low resistance distance $\Omega_{a,b} < 1/2$ in G implies localization of the starting state $|\psi_0\rangle = (|ab\rangle - |ba\rangle)/\sqrt{2}$.*

2.6.5 High Connectivity

Resistance distance is known to be closely related to edge-connectivity. Two vertices s and t are said to be k-edge-connected if there exists k edge-disjoint paths from s to t.

A tight relation was proved in [AGM13]: if s and t are k-edge-connected, then

$$\Omega_{s,t} = O\left(\frac{N^{2/3}}{k}\right). \tag{2.14}$$

If one also looks at the lengths of the paths, then a stronger statement is true. Let s and t be connected by k edge-disjoint paths with lengths $\ell_1, \ell_2, \ldots, \ell_k$. The resistance distance between s and t in the subgraph induced by these paths may only be larger than in the original graph. In this subgraph, the paths correspond to k resistors connected in parallel with resistances equal to $\ell_1, \ell_2, \ldots, \ell_k$. Then we have the following upper bound:

$$\Omega_{s,t} \leq \frac{1}{\sum_{i=1}^{k} \frac{1}{\ell_i}}. \tag{2.15}$$

In the context of this work, we arrive at the following conclusion by Theorems 12 and 14, that high edge-connectivity implies low resistance distance:

Theorem 15. *High edge-connectivity between a_{out} and b_{in} in G_b implies oscillatory localization between $|ab\rangle$ and $-|ba\rangle$. On the other hand, high edge-connectivity between a and b in G implies that $|\psi_0\rangle = (|ab\rangle - |ba\rangle)/\sqrt{2}$ is stationary.*

In particular, $k = \omega(N^{2/3})$ means that $\Omega_{s,t} = o(1)$ by (2.14) and therefore implies localization. For instance, edge-connectivity between any two vertices in the complete graph is high: $\Theta(N)$ and then $\Omega_{s,t} = O\left(1/N^{1/3}\right)$. On the other hand, edge-connectivity between two vertices s and t connected by an edge in a d-dimensional hypercube is only $\Theta(d) = \Theta(\log N)$, so (2.14) is not enough. There are $\Theta(d)$ edge-disjoint paths of length 3, however, between s and t; thus by (2.15), the resistance is small: $\Omega_{s,t} = O\left(3/\log N\right)$. Therefore, edge-connectivity is another useful measure of graphs that implies (oscillatory) localization of quantum walks.

2.7 Examples

A large variety of regular graphs have low resistance distance. For instance, the resistance distance between any two vertices u and v connected by an edge in a d-regular edge-transitive graph is given by [Fos49, KR93]:

$$\Omega_{u,v} = \frac{N-1}{dN/2} \approx \frac{2}{d}. \tag{2.16}$$

Similarly for the bipartite double graph,

$$\Omega_{u_{\text{out}},v_{\text{in}}} = \frac{2N-1}{dN} \approx \frac{2}{d}. \tag{2.17}$$

Thus in any case, the resistance distance is small provided the degree d is high. Then from Theorems 12 and 14, $|ab\rangle$ oscillates with $-|ba\rangle$, while $(|ab\rangle - |ba\rangle)/\sqrt{2}$ is localized, for edge-transitive graphs including complete graphs, complete bipartite graphs, hypercubes, and arbitrary-dimensional square lattices with degree greater than four. In addition, both of these states correspond to a single edge in the electric network. Then the current, which minimizes the energy dissipation, exactly corresponds to the flip state closest to $|\psi_0\rangle$, which implies equality in (2.9). Then $|\alpha|^2$ can be found exactly by substituting (2.16) or (2.17) into (2.11) and then into (2.9), yielding $|\alpha|^2 = (dN - 2N + 2)/dN$ and $(dN - 2N + 1)/dN$, respectively.

Using this result, let us revisit our initial example in Fig. 2.1 of a quantum walk on the complete graph with starting state $|\psi_0\rangle = |ab\rangle$. Then the degree is $d = N - 1$, and the probability overlap of the starting state with a flip state is

$$|\alpha|^2 = \frac{N(N-1) - 2N + 1}{N(N-1)} = 1 - \frac{1}{N-1} - \frac{1}{N}.$$

The value of $|\beta|^2$ can also be calculated explicitly as

$$|\beta|^2 = |\langle ab|\sigma_V\rangle|^2 = \left|\frac{1}{\sqrt{dN}}\right|^2 = \frac{1}{N(N-1)}.$$

Using these precise values of $|\alpha|^2$ and $|\beta|^2$ in Theorem 10, the amplitude of the state being in its initial state $|ab\rangle$ at even timesteps is

$$\left|\langle ab|U^{2t}|ab\rangle\right| \geq 2\left(|\alpha|^2+|\beta|^2\right)-1 = 2\left(1-\frac{1}{N-1}-\frac{1}{N}+\frac{1}{N(N-1)}\right)-1 = 1-\frac{4}{N}.$$

On the other hand, the amplitude of being in its flipped version $-|ba\rangle$ at odd timesteps is

$$\begin{aligned}\left|\langle ba|U^{2t+1}|ab\rangle\right| &\geq 2\max\left(1-\frac{1}{N-1}-\frac{1}{N}, \frac{1}{N(N-1)}\right)-1 \\ &= 2\left(1-\frac{1}{N-1}-\frac{1}{N}\right)-1 = 1-\frac{2}{N-1}-\frac{2}{N}.\end{aligned}$$

Let us compare these bounds to the exact result in Section 2.3. There, we found in (2.1) that at even steps, the system was in $|ab\rangle$ with amplitude

$$\frac{N-2}{N}+\frac{2}{N}\cos(\theta t) = 1-\frac{2}{N}+\frac{2}{N}\cos(\theta t) \geq 1-\frac{4}{N},$$

so the bound we just derived from Theorem 10 is tight. From Section 2.3, we found in (2.2) that at odd steps that the system was in $-|ba\rangle$ with amplitude

$$\frac{N-3}{N-1} = 1-\frac{2}{N-1}.$$

So the bound we just derived is close to being tight.

Expander graphs [HLW06] are generally *not* edge-transitive, and they have significant applications in communication networks, representations of finite graphs, and error correcting codes. Expander graphs of degree d have resistance distance $\Theta(1/d)$ [CRR+96], hence expander graphs of degree $d = \omega(1)$ exhibit localization of the initial state $(|ab\rangle - |ba\rangle)/\sqrt{2}$.

A general graph, which may be irregular, also has low resistance distance $\Omega_{a,b} \leq \frac{4}{d} \leq \frac{8}{N}$ when its minimum degree $d \geq \lfloor \frac{N}{2} \rfloor$ is high [CRR+96]. So $(|ab\rangle - |ba\rangle)/\sqrt{2}$ is also localized for these graphs.

Chapter 3

Stationary States

In this chapter we describe the results concerning the stationary states in Grover's quantum search. They are based on the following paper:

[PVW16b] Krišjānis Prūsis, Jevgēnijs Vihrovs, and Thomas G. Wong. Stationary states in quantum walk search. *Phys. Rev. A*, 94:032334, 2016.

3.1 Introduction

Quantum computers are well-known for their ability to outperform classical computers in many algorithmic applications [Mon16]. One famous example is unstructured search, where one out of N items in a database is marked by an oracle that responds yes or no as to whether an item is marked. A classical computer finds the marked item in $O(N)$ queries, while a quantum computer takes $O(\sqrt{N})$ queries using Grover's algorithm [Gro96]. If there are k marked items, then the classical and quantum computers respectively search in $O(N/k)$ and $O(\sqrt{N/k})$ time. So additional marked items make the search problem easier for both types of computers, as expected.

If there is structure to the database, however, then there are scenarios where additional marked items make search easier for a classical computer but harder for a quantum computer. More precisely, say the database is formulated as a graph of N vertices where one or more vertices are marked. The edges of the graph define the structure by which one moves from vertex to vertex. To find a marked vertex, one approach is to classically and randomly walk on the graph, querying the oracle with each step until a marked vertex is found. Then the more marked vertices, the easier the search problem becomes since there will be more marked vertices for the random walk to stumble upon.

The opposite can be true in the quantum regime, where additional marked vertices make the problem harder, not easier. This occurs using a discrete-time Grover's quantum walk, where the vertices of the graph define an N-dimensional Hilbert space (see Section 1.4 for the definition). As shown by Ambainis, Kempe, and Rivosh [AKR05], applying this quantum algorithm to search for a unique marked vertex on the two-dimensional (2D) periodic square lattice yields a success probability of $O(1/\log N)$ after $O(\sqrt{N \log N})$ steps. With amplitude amplification [BHMT00], this results in an overall runtime of $O(\sqrt{N} \log N)$. Now say there are two marked vertices that are adjacent to each other. Classically, this makes the search problem easier. But Nahimovs and Rivosh [NR16] recently showed that the quantum walk search algorithm now takes time $O(N)$, completely losing its quantum speedup. This is

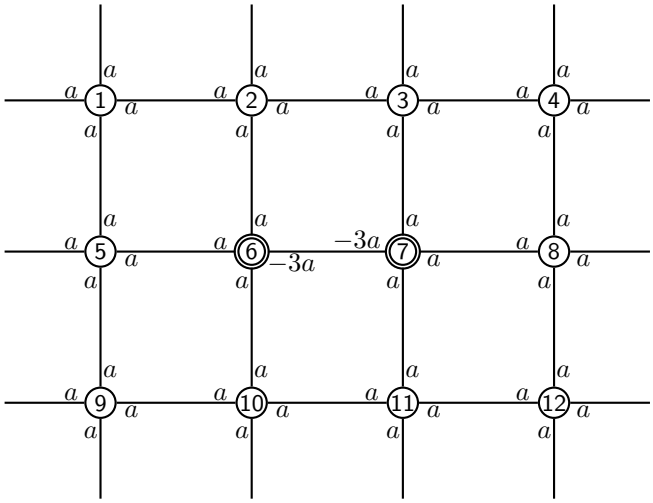


Figure 3.1: An optimal stationary state for the 4×3 periodic square grid with an adjacent pair of marked vertices (vertices 6 and 7, indicated by double circles).

because the initial uniform state (1.3) is approximately equal to a 1-eigenvector of the search operator U (1.1), and so the system fails to evolve, and the quantum algorithm is equivalent to classically guessing and checking. So in this example, having an additional marked vertex makes the search problem harder, not easier, for the quantum algorithm.

More precisely, the stationary state that approximates the initial uniform state (1.3) is depicted in Fig. 3.1. This figure reveals three properties, identified by Nahimovs and Rivosh [NR16] and further explored in follow-up work by Nahimovs and Santos [NS17], of stationary states. First, the directional amplitudes of unmarked vertices (depicted by single circles) are equal. For example, vertex 1 has amplitude a (chosen to normalize the overall state) in each of its four directions. Second, the directional amplitudes of marked vertices (depicted by double circles) sum to 0. For example, the sum of vertex 6's amplitudes is $a - 3a + a + a = 0$. Third, the directional amplitudes of adjacent vertices pointing to each other are equal. For example, vertices 6 and 7 point to each other with amplitude $-3a$. Nahimovs, Rivosh, and Santos [NR16, NS17] showed that a state with these three properties is a stationary state (i.e., 1-eigenvector) of the quantum walk search operator U (1.1). Going through each operator in U, the query Q flips the sign of the marked vertices, the coin C again flips the sign of the marked vertices since their average amplitudes are zero, and the shift S swaps pairs of amplitudes pointing to each other, which are equal. Thus U leaves such states invariant.

Another example, which to the best of our knowledge is historically the first example of the quantum walk search algorithm beginning in an approximate stationary state, is the simplex of complete graphs with a fully marked clique [WA15]. An example is depicted in Fig. 3.2, and the labels express the stationary state satisfying the three conditions of Nahimovs, Rivosh, and Santos [NR16, NS17]. Since the initial uniform state (1.3) is approximately equal to this stationary state, the quantum algorithm is no better than classically guessing and checking.

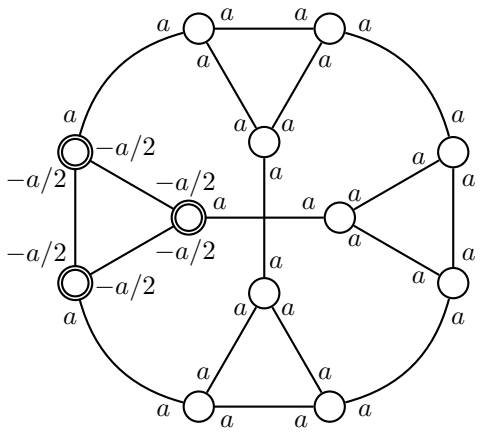


Figure 3.2: An optimal stationary state for the simplex of 4 complete graphs, each with 3 vertices, with a completely marked clique (indicated by double circles).

3.2 Overview of the Results

In this work, we show that the stationary states described by Nahimovs, Rivosh, and Santos [NR16, NS17] are not the only stationary states. We give the exact necessary and sufficient conditions on the amplitudes between adjacent vertices for a state to be stationary. In doing so, we greatly expand the number of known stationary configurations to an infinite class. We do this by incorporating concepts from [APVW16] regarding uniform and flip states, which form an orthogonal basis for directional states. Then we show that the type of stationary state described by Nahimovs, Rivosh, and Santos [NR16, NS17] is optimal, in the sense that it is the stationary states closest to the initial uniform state (1.3). Then we prove two theorems about the existence of stationary states on general graphs and with a connected component of marked vertices. In particular, if the marked vertices form a bipartite component, then a stationary state exists if and only if the sum of the amplitudes to be assigned to each partite set are equal. If they form a non-bipartite component, on the other hand, then a stationary state always exists.

3.3 Stationary States

In this section, we begin by adapting the concepts of uniform and flip states from Section 2.4, which form an orthogonal basis for directional states. Then we give conditions for a state to be stationary under the quantum walk search operator U (1.1). This exactly characterizes the stationary states of the walk, and it greatly expands stationary states beyond those of Nahimovs, Rivosh, and Santos [NR16, NS17]. Nonetheless, we prove that their stationary states are optimal, meaning they are the stationary states closest to the initial uniform state (1.3).

3.3.1 Uniform and Flip States

Let $|v_c\rangle$ be the (likely unnormalized) directional state at vertex v. For example, in Fig. 3.1, the directional state at vertex 6 is $|6_c\rangle = a\,|\uparrow\rangle - 3a\,|\rightarrow\rangle + a\,|\downarrow\rangle + a\,|\leftarrow\rangle$. In general, we can

express a directional state as $|v_c\rangle = \sum_{i=1}^{d_v} \alpha_i |i\rangle$. From this, we define uniform and flip states:

Definition 6. We call $|v_c\rangle$ a *uniform state* if $\alpha_1 = \alpha_2 = \ldots = \alpha_{d_v}$.

Definition 7. We call $|v_c\rangle$ a *flip state* if $\sum_{i=1}^{d_v} \alpha_i = 0$.

Note that whereas previously we defined uniform and flip states with regard to all vertices, now we only define them here with regards to individual vertices.

Uniform and flip states are useful because they are a complete orthogonal basis for directional states. To prove this, we first show that any uniform state $|\sigma\rangle$ is orthogonal to any flip state $|\phi\rangle$:

$$\langle\phi|\sigma\rangle = \sum_{i=1}^{d_v} \langle\phi|i\rangle \langle i|\sigma\rangle = \overline{\sigma} \sum_{i=1}^{d_v} \langle\phi|i\rangle = 0,$$

where $\overline{\sigma} = \frac{1}{d_v} \sum_{i=1}^{d_v} \langle i|\sigma\rangle$ denotes the average of the amplitudes of $|\sigma\rangle$.

Now let us show that uniform and flip states are a complete basis. Consider an arbitrary directional state $|v_c\rangle$, and let $\overline{v_c} = \frac{1}{d_v} \sum_{i=1}^{d_v} \langle i|v_c\rangle$ be the average of its amplitudes. Define the uniform state $|v_\sigma\rangle$ such that $\langle i|v_\sigma\rangle = \overline{v_c}$ for all i. Now consider the state $|v_\phi\rangle = |v_c\rangle - |v_\sigma\rangle$. It is a flip state, since

$$\sum_{i=1}^{d_v} \langle i|v_\phi\rangle = \sum_{i=1}^{d_v} \langle i|v\rangle - \sum_{i=1}^{d_v} \langle i|v_\sigma\rangle = \sum_{i=1}^{d_v} \langle i|v\rangle - d_v \cdot \overline{v_c} = 0.$$

Thus $|v_c\rangle$ can be expressed as a linear combination of uniform and flip states. For reference, let us write this as a Lemma:

Lemma 16. *Any directional state $|v_c\rangle$ can be expressed as the sum of a uniform state $|v_\sigma\rangle$ and a flip state $|v_\phi\rangle$.*

3.3.2 General Stationary States

Using uniform and flip states, we now derive if and only if conditions for a state to be stationary under the quantum walk search operator U (1.1).

Theorem 17. *Let $|\psi\rangle$ be the state of the quantum walk. For each pair of adjacent vertices a and b, let the amplitude on $|ab\rangle$ in $|\psi\rangle$ be $\sigma_1 + \phi_1$, where σ_1 comes from the uniform part of a and ϕ_1 comes from the flip part of a. Similarly, let the amplitude on $|ba\rangle$ in $|\psi\rangle$ be $\sigma_2 + \phi_2$. Then $|\psi\rangle$ is stationary under the quantum walk search operator $U = SCQ$ if and only if:*

1. *if a is unmarked and b is marked, then $\sigma_1 = \phi_2$ and $\phi_1 = -\sigma_2$;*
2. *if a and b are both unmarked, then $\sigma_1 = \sigma_2$ and $\phi_1 = -\phi_2$;*
3. *if a and b are both marked, then $\sigma_1 = -\sigma_2$ and $\phi_1 = \phi_2$.*

Proof. For each pair of adjacent vertices a and b, there are three possibilities for whether or not they are marked, as depicted in Fig. 3.3. Let us consider each of these possibilities.

P1. Suppose one vertex is unmarked and the other is marked. Without loss of generality, say a is unmarked and b marked. Now consider the action of $U = SCQ$. The oracle query flips the amplitude at marked vertices (in this case, the b vertex), the coin inverts about the

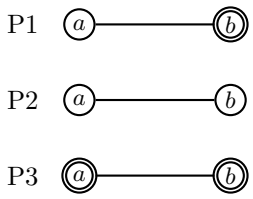


Figure 3.3: For a pair of adjacent vertices a and b, the three possibilities of whether or not they are marked. The double circle indicates a marked vertex.

average (so it leaves stationary states alone and flips the sign of flip states), and the shift swaps the amplitudes at $|ab\rangle$ and $|ba\rangle$. Explicitly, here is what each operator does to the amplitudes:

$$|ab\rangle : \sigma_1 + \phi_1 \xrightarrow{Q} \sigma_1 + \phi_1 \xrightarrow{C} \sigma_1 - \phi_1 \xrightarrow{S} -\sigma_2 + \phi_2$$
$$|ba\rangle : \sigma_2 + \phi_2 \xrightarrow{Q} -\sigma_2 - \phi_2 \xrightarrow{C} -\sigma_2 + \phi_2 \xrightarrow{S} \sigma_1 - \phi_1.$$

For $|\psi\rangle$ to be stationary, the amplitude before and after the application of U must be the same:

$$\sigma_1 + \phi_1 = -\sigma_2 + \phi_2$$
$$\sigma_2 + \phi_2 = \sigma_1 - \phi_1.$$

The solution is given by $\sigma_1 = \phi_2$ and $\phi_1 = -\sigma_2$.

P2. Now suppose that both a and b are unmarked. Then similarly, for $|\psi\rangle$ to be stationary, we have

$$\sigma_1 + \phi_1 = \sigma_2 - \phi_2$$
$$\sigma_2 + \phi_2 = \sigma_1 - \phi_1.$$

The solution to this system is $\sigma_1 = \sigma_2$ and $\phi_1 = -\phi_2$.

P3. Lastly, suppose that both a and b are marked. Then for $|\psi\rangle$ to be stationary, we have

$$\sigma_1 + \phi_1 = -\sigma_2 + \phi_2$$
$$\sigma_2 + \phi_2 = -\sigma_1 + \phi_1.$$

Here we have $\sigma_1 = -\sigma_2$ and $\phi_1 = \phi_2$.

A stationary state satisfies all these properties, and a state that satisfies these properties for all edges is stationary. □

Example 3. Figure 3.4 gives an example of a general stationary state for a pair of adjacent marked vertices on the 2D grid. For each edge, the amplitude is labeled by the uniform contribution plus the flip contribution. For example, consider the edge between vertices 5 and 6. For $|5, 6\rangle$, we have amplitude $a + b$, which means $\sigma_1 = a$ and $\phi_1 = b$. For $|6, 5\rangle$, we have amplitude $-b + a$, which means that $\sigma_2 = -b$ and $\phi_2 = a$. This satisfies P1, where $\sigma_1 = \phi_2 = a$ and $\phi_1 = -\sigma_2 = b$. Similarly, the rest of the edges satisfy Theorem 17, so this is a stationary state.

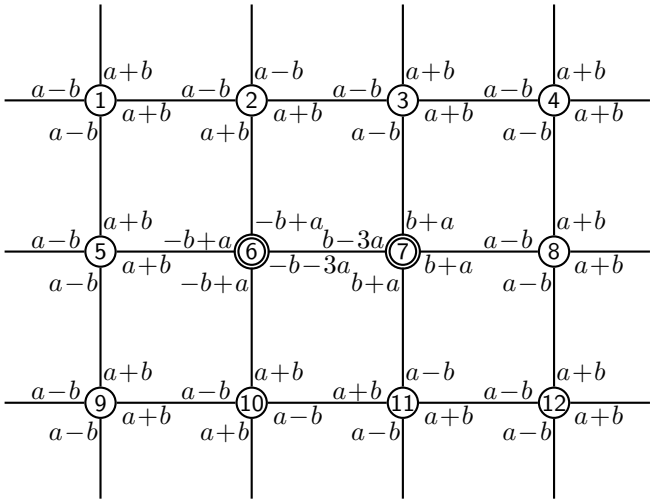


Figure 3.4: A general stationary state for the 4×3 periodic square grid with an adjacent pair of marked vertices (vertices 6 and 7, indicated by double circles). For each amplitude, the uniform component is listed first followed by its flip component.

Note that Theorem 17 exactly characterizes all stationary states of the search, and it greatly expands the number of known stationary states. For example, in Fig. 3.4, a and b are continuous parameters (which also normalize the overall state), so there is an infinite number of stationary states. This contrasts with Nahimovs, Rivosh, and Santos [NR16, NS17], who only considered stationary states where unmarked vertices were uniform states and marked vertices were flip states, as in Fig. 3.1.

3.3.3 Optimal Stationary States

From the previous Theorem, a general stationary state can have both uniform and flip components at each vertex, and this can lead to an infinite number of stationary states. Algorithmically, however, we are interested in the *optimal* stationary state, meaning the stationary state closest to the initial uniform state (1.3). That is, we are interested in the stationary state $|\psi\rangle$ such that $|\langle\psi_0|\psi\rangle|$ is maximized.

As we prove next, it turns out that this optimal stationary state is precisely the one described by Nahimovs, Rivosh, and Santos [NR16, NS17].

Theorem 18. *The stationary state $|\psi\rangle$ maximizing $|\langle\psi_0|\psi\rangle|$ satisfies the following properties:*

1. *the directional state of every unmarked vertex is a uniform state;*
2. *the directional state of every marked vertex is a flip state;*
3. *the amplitudes of adjacent vertices pointing to each other are equal. That is, $\langle uv|\psi\rangle = \langle vu|\psi\rangle$ for all $u \sim v$.*

Proof. We will prove that if $|\psi\rangle$ is an arbitrary stationary state, then the flip part of each unmarked vertex and the uniform part of each marked vertex together contribute zero to the inner product $\langle\psi_0|\psi\rangle$. Hence we can remove them, resulting in each unmarked vertex being

a uniform state and each marked vertex being a flip state. Then upon normalization, this has maximal overlap with the initial uniform state.

The contribution from any flip state to the inner product $\langle\psi_0|\psi\rangle$ is 0, since we showed in Lemma 16 that any flip state is orthogonal to a uniform state. Thus the flip parts of the unmarked vertices contribute nothing to $\langle\psi_0|\psi\rangle$, so we remove them. This proves part (a) of the Theorem.

Next we show that the total contribution from the uniform parts of the marked vertices to $\langle\psi_0|\psi\rangle$ is equal to 0. Let the total sum of the amplitudes of $|\psi\rangle$ and $U|\psi\rangle$ be s and s', respectively. Beginning with s,

$$s = \sum_{v=1}^{N} \sum_{i=1}^{d_v} \langle i|v_c\rangle .$$

From Lemma 16, we can express a directional state $|v_c\rangle = |v_\sigma\rangle + |v_\phi\rangle$, where $|v_\sigma\rangle$ is a uniform state and $|v_\phi\rangle$ is a flip state. The flip states in the sum can be ignored, since $\sum_{i=1}^{d_v} \langle i|v_\phi\rangle = 0$. Then

$$s = \sum_{v=1}^{N} \sum_{i=1}^{d_v} \langle i|v_\sigma\rangle .$$

Now for s'. The oracle flips the sign of the amplitude at marked vertices, so

$$s' = \sum_{v\notin M} \sum_{i=1}^{d_v} \langle i|v_\sigma\rangle - \sum_{v\in M} \sum_{i=1}^{d_v} \langle i|v_\sigma\rangle ,$$

where M is the set of the marked vertices. Since $|\psi\rangle$ is stationary, $s = s'$. Therefore

$$\sum_{v\in M} \sum_{i=1}^{d_v} \langle i|v_\sigma\rangle = 0.$$

Thus if we look at the contribution from the uniform states at marked vertices to $\langle\psi_0|\psi\rangle$, it is indeed equal to 0:

$$\sum_{v\in M} \sum_{i=1}^{d_v} \langle v_c(0)|v_\sigma\rangle = \sum_{v\in M} \sum_{i=1}^{d_v} \langle v_c(0)|i\rangle \langle i|v_\sigma\rangle = \sum_{v\in M} \sum_{i=1}^{d_v} \frac{1}{\sqrt{2|E|}} \langle i|v_\sigma\rangle$$
$$= \frac{1}{\sqrt{2|E|}} \sum_{v\in M} \sum_{i=1}^{d_v} \langle i|v_\sigma\rangle = 0.$$

Hence we remove the uniform parts from the marked vertices. This proves part (b) of the Theorem.

Now that we have removed the flip components from unmarked vertices and the uniform components from marked vertices, consider what this did to the properties of Theorem 17:

- For P1, we now have $\phi_1 = \sigma_2 = 0$ and $\sigma_1 = \phi_2$.
- For P2, we now have $\phi_1 = \phi_2 = 0$ and $\sigma_1 = \sigma_2$.
- For P3, we now have $\sigma_1 = \sigma_2 = 0$ and $\phi_1 = \phi_2$.

So for the resulting state to be stationary (i.e., satisfy these three properties), we require that the amplitudes of adjacent vertices pointing to each other are equal, yielding part (c) of the Theorem. □

As an example of this reduction from a general stationary state to the optimal one, consider again the marked pair of adjacent vertices in Fig. 3.4. We can remove the flip components from the unmarked vertices (the $\pm b$ parts) and the uniform components from the marked vertices (also the $\pm b$ parts), resulting in Fig. 3.1, which is the optimal stationary state (with normalization).

3.4 Existence of Stationary States

In Theorems 17 and 18, general and optimal stationary states are given in terms of the relations between the uniform and flip components of each amplitude, depending on whether vertices are marked or not (cases P1, P2, and P3 in Fig. 3.3). In practice, however, finding a solution to all these conditions, if one exists, can be difficult. So in this section, we give two theorems for the existence of stationary states.

To investigate the existence of stationary states, it suffices to study optimal ones. This is because any general stationary state can be optimized to have maximal overlap with the initial uniform state (1.3) according to Theorem 18, namely by removing the flip components from unmarked vertices and the uniform components from unmarked vertices (and normalizing). So if no optimal stationary states exist for a certain graph and configuration of marked vertices, then no general stationary states exist, either. Furthermore, optimal stationary states determine how closely the initial uniform state (1.3) of the quantum walk search algorithm is to being stationary.

Before deriving our two theorems, let us provide some additional background. Nahimovs, Rivosh, and Santos [NR16, NS17] showed that for general graphs, a marked pair of adjacent vertices has an optimal stationary state if both marked vertices have the same degree. This is depicted in Fig. 3.5. Note that we only need to assign one amplitude to each edge of the graph since optimal stationary states have the same amplitude in both directions, i.e., part (c) of Theorem 18. This contrasts with a marked triangle in Fig. 3.6, where Nahimovs, Rivosh, and Santos gave the optimal stationary state, even if the marked vertices have different degrees. This raises the question of why equal degrees were used for the pair while unequal degrees were allowed for the triangle. Our next two theorems precisely explain why: It is because the pair is bipartite, which has constraints for the existence of stationary states, while the triangle is non-bipartite, which always has a stationary state.

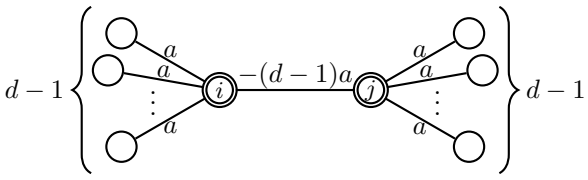


Figure 3.5: The optimal stationary state for a marked pair of adjacent vertices of equal degree.

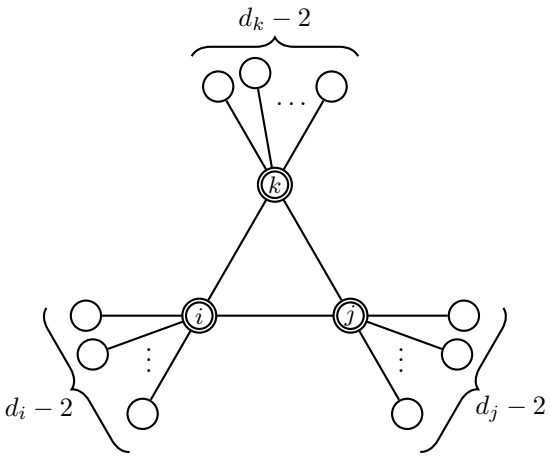


Figure 3.6: A marked triangle, where vertices can have unequal degree.

Assume that the marked vertices form a connected component M. The unmarked vertices may form one or more connected components $U_1, \ldots, U_{k_u}$. To construct an optimal stationary state, we first assign the amplitudes in each U_i so that each vertex is the same uniform state. This ensures that the unmarked vertices are uniform and the amplitudes of adjacent unmarked vertices pointing to each other are equal. Now we must determine how to assign the amplitudes of the marked vertices.

A marked vertex $v \in M$ is connected to some of the unmarked vertices. Since these unmarked vertices have already been assigned amplitudes, the amplitudes on these edges are also known. Let us say the sum of the amplitudes on these edges is equal to Σ. As v is marked, it should be a flip state, so the sum of the edges going from v to the other marked vertices is equal to $-\Sigma$. We call this value the *shortage* at vertex v and denote it by $s_v = -\Sigma$. Thus each vertex of M has some shortage value, and we want to know when it is possible to assign the amplitudes on the edges between the marked vertices so as to neutralize all of the shortages.

In the following two theorems, we consider when M is bipartite and non-bipartite. If it is bipartite, then the shortages can be neutralized if and only if the sum of the shortages on both partite sets are equal. On the other hand, if the marked vertices are non-bipartite, then the shortages can always be neutralized. This explains why Nahimovs, Rivosh, and Santos used equal degrees for the pair of marked vertices in Fig. 3.5, whereas different degrees were allowed for the marked triangle in Fig. 3.6.

3.4.1 Bipartite Marked Component

Theorem 19. *If M is bipartite, then we can assign amplitudes to neutralize the shortages at each marked vertex if and only if the sum of the shortages on both partite sets are equal.*

Proof. Let the partite sets be X and Y.

($\Rightarrow$) If the state is optimally stationary, then the amplitudes of adjacent vertices pointing to each other are equal. Then the sum of the shortages at X must be equal to that of Y.

($\Leftarrow$) Now we assume that the sum of the shortages of both partite sets are equal. We prove that an optimal stationary state exists by giving a procedure for assigning the amplitudes.

Pick any vertex $v \in M$ with $s_v \neq 0$. Without loss of generality suppose that $v \in X$. Suppose that removing v and all its incident edges breaks up M into connected components $C_1, \ldots, C_t$. For example, consider the marked connected component in Fig. 3.7a, which is bipartite. Consider vertex $v = 1$. If we remove it and its edges, then we have two connected components $C_1 = \{2, 4, 5, 6\}$ and $C_2 = \{3, 7\}$.

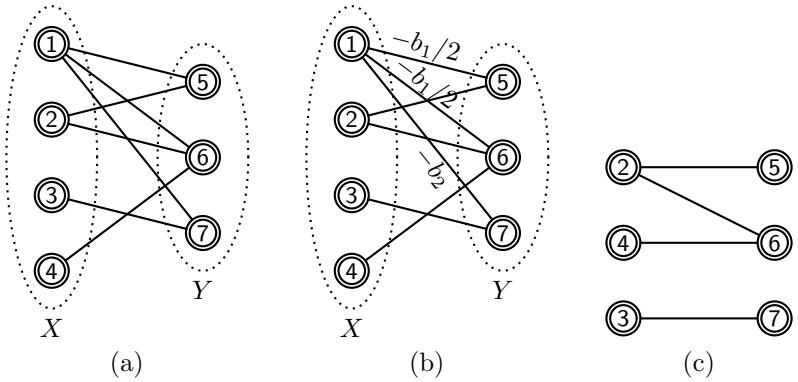


Figure 3.7: (a) A bipartite marked connected component. (b) Assignments of vertex 1's edges within the marked connected component to neutralize its shortage, making it a flip state. (c) The marked vertices after removing vertex 1.

For each connected component C_i, define b_i, where we take the shortages in the X partite set and subtract the shortages in the Y partite set:

$$b_i = \sum_{x \in C_i \cap X} s_x - \sum_{y \in C_i \cap Y} s_y.$$

For our example, we have

$$b_1 = s_2 + s_4 - s_5 - s_6$$
$$b_2 = s_3 - s_7.$$

Now consider the shortage at v, plus these b_i's:

$$s_v + \sum_{i=1}^{t} b_i = \sum_{x \in X} s_x - \sum_{y \in Y} s_y = 0.$$

This equals zero because the sum of the shortages of both partite sets are equal. Solving for the shortage at v, we find that it is equal to the negative of the sum of the b_i's:

$$s_v = -\sum_{i=1}^{t} b_i.$$

So for our example in Fig. 3.7a, we have $s_1 = -(b_1 + b_2)$.

Let the number of edges from v to C_i be n_i. Then on each of these edges we assign an amplitude of $-b_i/n_i$. By construction, this neutralizes the shortage of v, since

$$\sum_{i=1}^{t} n_i \cdot \left(-\frac{b_i}{n_i}\right) = -\sum_{i=1}^{t} b_i = s_v.$$

For our example, since v connects to C_1 through two edges, we assign $-b_1/2$ to each of those edges. And since v connects to C_2 through one edge, we assign $-b_2$ to that edge. This is shown in Fig. 3.7b. By construction, we have now neutralized the shortage $s_1 = -(b_1 + b_2)$ at $v = 1$, making it a flip state.

Besides neutralizing the shortage of v, this assignment also causes the sums of the shortages of C_i on both of its partite sets to now be equal. For our example, in Fig. 3.7b, the shortages of vertices 5, 6, and 7 have changed due to the assignments for $v = 1$. They are now

$$\begin{aligned} s'_5 &= s_5 + \frac{b_1}{2} \\ s'_6 &= s_6 + \frac{b_1}{2} \\ s'_7 &= s_7 + b_2. \end{aligned}$$

Now let us sum the shortages for C_1. The sum in X is $s_2 + s_4$, and the sum of the shortages in Y is $s'_5 + s'_6 = s_5 + s_6 + b_1 = s_2 + s_4$. So they are equal. This similarly holds for C_2. In general, we subtract the assigned amplitudes from the shortages of the neighbors of v and denote the new shortages by s'. Each neighbor of v in C_i is in Y, and each such neighbor has $s'_y = s_y + b_i/n_i$. As there are n_i such neighbors, we have that

$$\begin{aligned} \sum_{x \in C_i \cap X} s'_x - \sum_{y \in C_i \cap Y} s'_y &= \sum_{x \in C_i \cap X} s_x - \sum_{y \in C_i \cap Y} s_y - b_i \\ &= b_i - b_i = 0. \end{aligned}$$

Each C_i is now a separate bipartite connected component, and the sum of the shortages on both partite sets in each C_i are equal. Visualizing this for our example, removing vertex $v = 1$ leaves the two disconnected components, as shown in Fig. 3.7c. Each of these C_i's has the property that the sum of the shortages on both partite sets are equal, so we can recursively repeat the assignment procedure until the shortage at each vertex is 0. The recursion stops when we have pairs, which can be made to have zero shortage. For example, consider the connected pair of vertices 3 and 7 in Fig. 3.7c. Since the sum of shortages in X and Y are equal, we have that $s_3 = s_7$. Using the procedure, let $v = 3$. Then $C_1 = \{7\}$ and $b_1 = -s_7$. So we assign the edge an amplitude of $-b_1 = s_7$. Since $s_3 = s_7$, we have neutralized the shortages of both vertices. □

Applying this to the marked pair of adjacent vertices in Fig. 3.5, the unmarked vertices are assumed to form a single connected component, so all their amplitudes are a. Then the marked vertices have shortages $s_i = (d_i - 1)a$ and $s_j = (d_j - 1)a$. For a stationary state to exist, these shortages must be equal, which means the marked vertices must have equal degree $d_i = d_j$. This proves why Nahimovs, Rivosh, and Santos [NR16, NS17] could only find a stationary state with this requirement.

Now say the unmarked vertices form multiple connected components $U_1, \ldots, U_{k_u}$. For example, consider the marked pair of adjacent vertices in Fig. 3.8, where each marked vertex is connected to a different unmarked connected component U_1 and U_2. For each vertex in U_1, we assign all edges the same amplitude a, while for U_2, we assign the value b. Note that any value of a and b makes the unmarked vertices uniform, as desired. But for an optimal stationary state to exist, we specifically require that $2a = 3b$ so that the sum of the shortages on marked vertices i and j are equal.

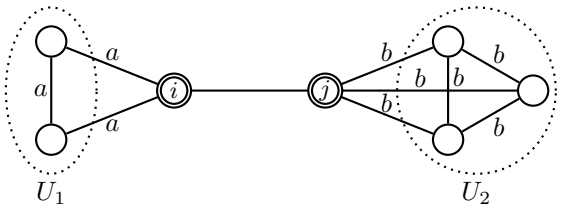


Figure 3.8: A graph with a marked pair of adjacent vertices and two unmarked connected components.

In general, it may not be possible to assign uniform states to the U_i's so that the sum of the shortages on the partite sets of M are equal, so stationary states do not always exist. A simple example is a graph of two vertices connected by a single edge, where one of the two vertices is marked. Then there is no assignment to the edge that defines a stationary state. Furthermore, one can construct infinitely many examples where there is one unmarked component U and one marked bipartite component M such that the sums of the shortages cannot be equal no matter how the amplitudes are assigned on the unmarked component. Given this, we leave the details of how to assign the unmarked components for further research.

3.4.2 Non-Bipartite Marked Component

Now we consider the case when the marked vertices form a non-bipartite connected component M. In contrast to the previous theorem for bipartite graphs, here no constraints arise, so stationary states always exist for non-bipartite M.

Theorem 20. *If M is non-bipartite, then we can always assign the amplitudes to neutralize the shortages at each marked vertex.*

Proof. We prove the theorem by giving an explicit procedure for assigning the amplitudes. Since M is non-bipartite, there exists a cycle of odd length $\{v_1, \ldots, v_k\}$.

For now, suppose that this cycle contains all vertices of M. Then there always exists a solution for assigning the amplitudes on the edges of this cycle so to neutralize the shortages. It is given by

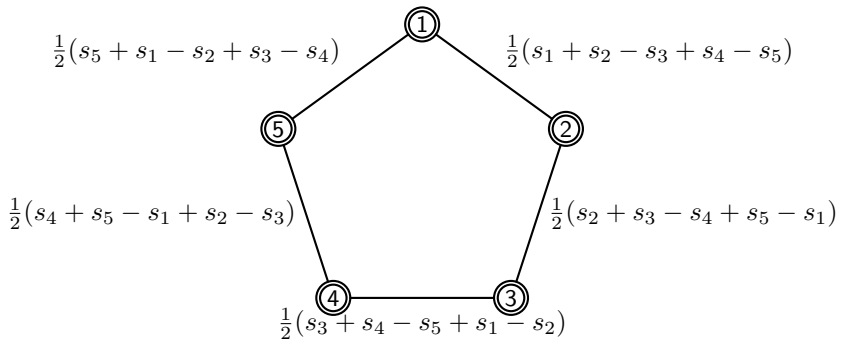
$$\langle v_i v_{i+1}|\psi\rangle = \langle v_{i+1} v_i|\psi\rangle = \frac{1}{2}\sum_{j=1}^{k}(-1)^{(i-j) \pmod k} s_{v_j}.$$

For example, for a 5-cycle of marked vertices, we assign the edges as shown in Fig. 3.9a. So the assignment of the edges neutralizes the shortages. In general, we have

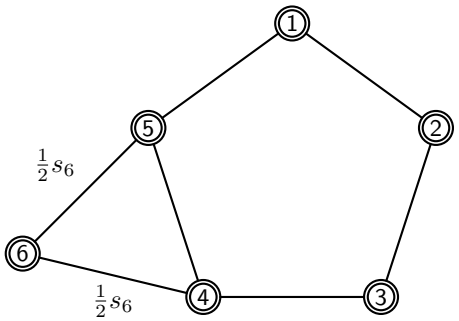
$$\langle v_{i-1} v_i|\psi\rangle + \langle v_i v_{i+1}|\psi\rangle = \frac{1}{2}\sum_{j=1}^{k}\left[(-1)^{(i-1-j) \pmod k} + (-1)^{(i-j) \pmod k}\right] s_{v_j}.$$

Here $x \pmod k$ is an integer from 0 to $k-1$, so in particular, $(-1)^{-1 \pmod k} = (-1)^{k-1} = 1$ as k is odd. The only time when $(i-1-j) \pmod k$ and $(i-j) \pmod k$ are equal (mod 2) is when $i = j$. Otherwise they sum up to 0. Hence the value of this sum is equal to s_{v_i}.

For all the other edges not in this cycle we assign amplitude 0. For example, in Fig. 3.9a, if there were an edge connecting vertices 1 and 3, we would assign zero amplitude to it.



(a)



(b)

Figure 3.9: (a) A non-bipartite marked connected component, entirely constituting an odd cycle. Amplitudes have been assigned to the edges to neutralize the shortage at each vertex, yielding an optimal stationary state. (b) With an additional connected marked vertex. Amplitudes have been assigned to the edges to neutralize the shortage at the additional vertex.

Now suppose there are some vertices of M not in this cycle. Pick a vertex u from among them that is the farthest from the cycle. More formally, $\min_{i=1}^{k} d(u, v_i)$ is maximized for u, where $d(a, b)$ is the shortest distance between a and b in M. Let the degree of u in this component be $\deg(u)$. Assign an amplitude of $s_u/\deg(u)$ for each edge from u in this component. For example, Fig. 3.9b now includes an extra vertex labeled 6, and we assign its edges within M the value $s_6/2$, neutralizing its shortage.

This way, the shortage of u is neutralized and now we are left with a connected component $M \setminus u$, since u was the farthest from the cycle. As u did not belong to the cycle, we can repeat the procedure recursively until the cycle contains all vertices of the component. □

Again, this explains why Nahimovs, Rivosh, and Santos [NR16, NS17] were able to find a stationary state for the marked triangle in Fig. 3.6, where each marked vertex could have different degrees.

3.4.3 Multiple Marked Connected Components

We end this section with some brief remarks about multiple marked connected components. In the previous two theorems, the marked vertices were all connected to each other, forming a single marked connected component M. Now say there are multiple such connected components $M_1, M_2, \ldots, M_{k_m}$. For the bipartite M_i's, if the sum of the shortages of both partite

sets are equal, then we can assign amplitudes to neutralize the shortages using Theorem 19. As discussed in Section 3.4.1, however, changing the uniform states of unmarked components $U_1, \ldots, U_{k_u}$, can change these sums. On the other hand, the non-bipartite M_i's can always have their shortages neutralized using Theorem 20.

Part II

Query Complexity

Overview of Part II

In this part we look at the complexity of algorithms in the query computational model. Specifically, we study methods that provide lower bounds on the time complexity of the randomized query algorithms. In Chapter 4, we give an introduction to the query model.

First, we investigate the relationship between various classical adversary bounds that have originated from quantum query complexity. Our main result concerns the behaviour of these methods on total and partial functions. It states that all the known classical adversary bounds are asymptotically equivalent for all total functions. For partial functions, we show examples of functions where some of these lower bounds give asymptotically different estimates. These results are described in Chapter 5.

Second, we study the relationship between two well-known closely related lower bounds, block sensitivity and fractional block sensitivity. We show a separation between these two measures and prove that it is optimal for partial functions. These results are described in Chapter 6.

Chapter 4

Query Complexity

In this chapter we give a brief introduction to the query computational model. For a detailed description of the model, see survey [BdW02].

4.1 Introduction

The goal of computational complexity theory is to understand the power of algorithms in different computational models. However, it is notoriously hard to understand the exact complexity of many algorithmic problems in the Turing model of computation. Query complexity is a very simple model of computation that offers a clean framework to prove the limitations of what an algorithm can and cannot do. A large variety of algorithms can be seen as query algorithms.

Query model has two important applications. The first is proving lower bounds on the amount of time required to solve a computational problem. In the query setting, the running time of the algorithm is the number of queries an algorithm makes during its run. A query lower bound implies a time lower bound in the Turing computational model. For example, binary search and sorting can be proven optimal by lower bounds in the query model [LM04].

The second application is the comparison of deterministic, randomized and quantum query algorithms. Let f be a function that the algorithm needs to compute. The respective running times of the fastest algorithm computing f are denoted by $\mathrm{D}(f), \mathrm{R}(f)$ and $\mathrm{Q}(f)$. Each model is more powerful than the last one, hence $\mathrm{Q}(f) \leq \mathrm{R}(f) \leq \mathrm{D}(f)$.

The main question is determining the relation between these models in the general case. If f is a total function (defined on the whole domain of the inputs), then the following relations are known:

$$\mathrm{D}(f) = O(\mathrm{R}(f)^3) \qquad \mathrm{D}(f) = O(\mathrm{Q}(f)^6) \qquad \mathrm{R}(f) = O(\mathrm{Q}(f)^6).$$

The best known separations between these measures are:

$$\mathrm{D}(f) = \Omega(\mathrm{R}(f)^2) \qquad \mathrm{D}(f) = \Omega(\mathrm{Q}(f)^4) \qquad \mathrm{R}(f) = \Omega(\mathrm{Q}(f)^{2.5}).$$

A major problem in query complexity is closing the gaps in these relations. For these results and an overview of the state-of-the-art relationships between these and many other measures, see [ABDK16] and [ABB+17].

4.2 Query Algorithms

In the query model, an algorithm has to compute a function $f : S \to H$, given a string x from $S \subseteq G^n$, where G and H are finite alphabets. The function f corresponds to the problem being solved, and the string $x = (x_1, x_2, \ldots, x_n)$ is the input to the problem. We say that f is *total* if $S = G^n$, and *partial* otherwise. A function is called *Boolean* if $G = H = \{0, 1\}$.

In the beginning of the computation, no information about x is known to the algorithm. The algorithm can use a black-box oracle *query* operation that, given an index $i \in [n]$, returns the value of x_i. After a number of queries (possibly, adaptive), the algorithm must compute $f(x)$. The cost of the computation is the number of queries to the oracle.

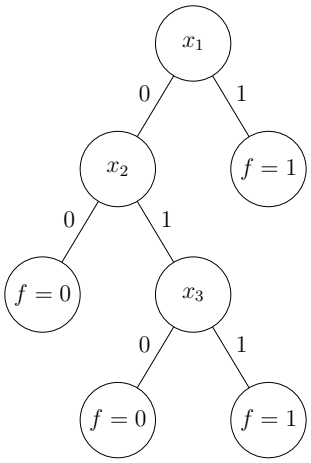


Figure 4.1: A decision tree computing the Boolean function $f(x) = x_1 \vee (x_2 \wedge x_3)$.

Deterministic Query Complexity. In the deterministic setting, an algorithm *computes* the function f if for all inputs x, the algorithm outputs the correct answer $f(x)$. The algorithm is adaptive, meaning that the index the algorithm asks in the k-th query depends on the results of the previous $k - 1$ queries. The *running time* of the algorithm is the maximum number of queries it makes on some input. The *deterministic query complexity* of f is the smallest running time among all algorithms that compute f, and is denoted by $\mathrm{D}(f)$.

A deterministic query algorithm can also be viewed as a binary decision tree. Each vertex is labeled by some x_i, and each leaf is assigned a Boolean value, either 0 or 1. The algorithm evaluates the tree as follows. It starts at the root of the tree and stops if it reaches a leaf. If the algorithm reaches a vertex, it queries the input bit it is labeled by. If the result of the query is 0, the algorithm proceeds to the left child, and to the right child otherwise. When it reaches a leaf, it returns the assigned value. Because of this reason query complexity is also called *decision tree complexity*. Figure 4.1 illustrates an example of a decision tree.

Randomized Query Complexity. In the randomized setting, the algorithm is allowed to use randomness when choosing the next query. Similarly to the deterministic case, a randomized algorithm can also be viewed as a decision tree. In this case the tree can also have special internal nodes labeled by real numbers between 0 and 1. When the algorithm reaches such a node, labeled by p, it chooses to evaluate its left child with probability p and its right child with probability $1 - p$.

In this work we focus solely on the so-called *bounded-error* randomized algorithms, which means that the algorithm sometimes can produce an incorrect answer. More specifically, we say that a randomized algorithm *computes* f with probability $2/3$ if for all inputs x in the domain it outputs the correct answer $f(x)$ with probability at least $2/3$.

The *running time* of the algorithm on an input is the maximum possible number of queries it can ask on that input (over all possible random choices). As before, the *running time* of the algorithm is its maximum running time over all inputs, and the *randomized query complexity* of f is the smallest running time among all algorithms that compute f, and is denoted by $\mathrm{R}(f)$. Clearly, randomized query algorithms are at least as powerful as deterministic ones, therefore $\mathrm{R}(f) \leq \mathrm{D}(f)$.

Quantum Query Complexity. In this setting, the algorithm operates in a quantum state. A k-qubit quantum state $|\psi\rangle$ is a superposition over all possible Boolean strings of length k,

$$|\psi\rangle = \sum_{i \in \{0,1\}^k} \alpha_i |i\rangle ,$$

where α_i are complex numbers called *amplitudes* and $\sum_{i \in \{0,1\}^k} |\alpha_i|^2 = 1$. In other words, $|\psi\rangle$ is a unit vector in the k-dimensional complex space $\mathbb{C}^k$.

There are two things that can be done with a quantum state:

- *Transforming the state using a unitary.* A matrix $U \in \mathbb{C}^{k \times k}$ is called *unitary* if $U^* U = I$, where U^* is the conjugate transpose of U and I is the identity matrix. Applying U to $|\psi\rangle$ results in the state $|\psi'\rangle = U |\psi\rangle$. Unitary matrices preserve the length of vectors, hence $|\psi'\rangle$ is also a unit vector.

- *Measuring the state.* According to quantum mechanics, if we measure $|\phi\rangle$ in the standard computational basis $\{|i\rangle \mid i \in \{0,1\}^k\}$, then the outcome of the measurement is $|i\rangle$ with probability $|\alpha_i|^2$. This can be interpreted as the state collapsing into a single classical state after the measurement.

Now we are ready to describe how a quantum query algorithm works. The state of the algorithm is a quantum state in the form

$$|\psi\rangle = \sum_{i,b,z} \alpha_{i,b,z} |i, b, z\rangle .$$

Here, the first register i is a $\lceil \log_2 n \rceil$ long string that denotes one of the states from 1 to n. The second register b is a Boolean value that is used to store the result of the query. Finally, z is a workspace of the algorithm which stores some relevant information (it is a Boolean string of some fixed length m).

The notion of the query operation is modified to the quantum setting. It is a unitary operation O that acts as follows:

$$O |i, b, z\rangle = |i, b \oplus x_i, z\rangle .$$

Hence applying O to the whole state $|\psi\rangle$ can be thought of as querying all the bits in superposition, and storing the result in the second register.

The quantum algorithm starts in the all-zeros state $|0\rangle$, and applies a number of unitaries in sequence. It alternates between oracle calls and unitary transformations that do not depend on the input x. Namely, a quantum algorithm that utilizes T queries transforms the starting state to

$$|\psi\rangle = U_T O U_{T-1} O \dots O U_1 O U_0 |0\rangle .$$

Here, U_i is a fixed unitary transformation applied after the i-th query. At the end of the computation, the algorithm measures $|\psi\rangle$ and depending on the result outputs some answer.

The *running time* of the algorithm is the number of applications of O. Similarly to the randomized case, we examine only bounded-error algorithms. The *quantum query complexity* is the minimum possible number of queries made by a bounded-error quantum algorithm that computes f, and is denoted by $\mathrm{Q}(f)$. Quantum query algorithms can simulate randomized algorithms, hence $\mathrm{Q}(f) \leq \mathrm{R}(f)$. For consistency, they are also often called *quantum decision trees*.

4.3 Lower Bounds

Lower bounds are methods that allow us to produce provable limitations on the time necessary for an algorithm to solve a problem. In query complexity, a measure $A(f)$ is a lower bound for another measure $B(f)$ if $B(f) = \Omega(A(f))$ for all functions f being considered. For example, if $D(f) \geq M(f)$ for some measure $M(f)$, then any deterministic algorithm that computes f must use at least $M(f)$ queries in the worst case.

Another application of lower bounds is relating the power of different computational models. For example, block sensitivity measure $\mathrm{bs}(f)$ (see below) lower bounds randomized query complexity, $\mathrm{R}(f) = \Omega(\mathrm{bs}(f))$. On the other hand, it is known that for total functions, $\mathrm{D}(f) \leq \mathrm{bs}(f)^3$ [Nis89, BBC+01]. By combining these two results, we immediately obtain that $\mathrm{D}(f) \leq \mathrm{R}(f)^3$. Therefore, randomized algorithms give at most polynomial advantage over deterministic ones for total functions.

In this work we mainly focus on two fundamental query lower bound methods called the *block sensitivity* and *certificate complexity*.

Block Sensitivity. For $x \in S$, a subset of indices $B \subseteq [n]$ is a *sensitive block* of x if there exists a y such that $f(x) \neq f(y)$ and $B = \{i \mid x_i \neq y_i\}$. The *block sensitivity* $\mathrm{bs}(f, x)$ of f on x is the maximum number k of disjoint subsets $B_1, \dots, B_k \subseteq [n]$ such that B_i is a sensitive block of x for each $i \in [k]$. The block sensitivity of f is defined as $\mathrm{bs}(f) = \max_{x \in S} \mathrm{bs}(f, x)$.

Let $\mathcal{B} = \{B \mid \exists y : f(x) \neq f(y) \text{ and } B = \{i \mid x_i \neq y_i\}\}$ be the set of sensitive blocks of x. The *fractional block sensitivity* $\mathrm{fbs}(f, x)$ of f on x is defined as the optimal value of the following linear program:

$$\text{maximize} \sum_{B \in \mathcal{B}} w_x(B) \qquad \text{subject to} \quad \forall i \in [n] : \sum_{\substack{B \in \mathcal{B} \\ i \in B}} w_x(B) \leq 1.$$

Here, we have $w_x \in [0; 1]^{|\mathcal{B}|}$. The fractional block sensitivity of f is defined as $\mathrm{fbs}(f) = \max_{x \in S} \mathrm{fbs}(f, x)$.

When the weights are taken as either 0 or 1, the optimal solution to the corresponding integer program is equal to $\mathrm{bs}(f, x)$. Hence $\mathrm{fbs}(f, x)$ is a relaxation of $\mathrm{bs}(f, x)$, and we have $\mathrm{bs}(f, x) \leq \mathrm{fbs}(f, x)$.

Certificate complexity. An *assignment* is a map $A : \{1, \ldots, n\} \to G \cup \{*\}$. Informally, the elements of G are the values fixed by the assignment and * is a wildcard symbol that can be any letter of G. A string $x \in S$ is said to be consistent with A if for all $i \in [n]$ such that $A(i) \neq *$, we have $x_i = A(i)$. The length of A is the number of positions that A fixes to a letter of G.

For an $h \in H$, an h-certificate for f is an assignment A such that for all strings $x \in A$ we have $f(x) = h$. The *certificate complexity* $\mathrm{C}(f, x)$ of f on x is the size of the shortest $f(x)$-certificate that x is consistent with. The certificate complexity of f is defined as $\mathrm{C}(f) = \max_{x \in S} \mathrm{C}(f, x)$. It is a deterministic lower bound, $\mathrm{D}(f) \geq \mathrm{C}(f)$.

The *fractional certificate complexity* $\mathrm{FC}(f, x)$ of f on $x \in S$ is defined as the optimal value of the following linear program:

$$\text{minimize} \sum_{i \in [n]} v_x(i) \qquad \text{subject to} \quad \forall y \in S \text{ s.t. } f(x) \neq f(y) : \sum_{i : x_i \neq y_i} v_x(i) \geq 1.$$

Here $v_x \in [0; 1]^n$ for each $x \in S$. The fractional certificate complexity of f is defined as $\mathrm{FC}(f) = \max_{x \in S} \mathrm{FC}(f, x)$.

When the weights are taken as either 0 or 1, the optimal solution to the corresponding integer program is equal to $\mathrm{C}(f, x)$. Hence $\mathrm{FC}(f, x)$ is a relaxation of $\mathrm{C}(f, x)$, and we have $\mathrm{FC}(f, x) \leq \mathrm{C}(f, x)$.

It has been shown that $\mathrm{fbs}(f, x)$ and $\mathrm{FC}(f, x)$ are dual linear programs, hence their optimal values are equal, $\mathrm{fbs}(f, x) = \mathrm{FC}(f, x)$. As an immediate corollary, $\mathrm{fbs}(f) = \mathrm{FC}(f)$.

Lower Bounds. Block sensitivity serves as a lower bound for both deterministic and randomized query algorithms, $\mathrm{D}(f) \geq \mathrm{bs}(f)$ and $\mathrm{R}(f) \geq \mathrm{bs}(f)/3$. Certificate complexity is always at least block sensitivity and it is a deterministic lower bound, $\mathrm{D}(f) \geq \mathrm{C}(f)$ [BdW02].

Fractional block sensitivity subsumes standard block sensitivity and gives better lower bounds. Another measure that is asymptotically equivalent to $\mathrm{fbs}(f)$ is the *randomized certificate complexity*, denoted by $\mathrm{RC}(f)$. Hence we have $\mathrm{fbs}(f) = \mathrm{FC}(f) = \Theta(\mathrm{RC}(f))$. Aaronson also introduced a quantum lower bound *quantum certificate complexity* $\mathrm{QC}(f)$ and showed that $\mathrm{QC}(f) = \Theta(\sqrt{\mathrm{RC}(f)})$ (see [Aar08] for the definitions).

Therefore, all the described measures give us the following set of general lower bounds:

Theorem 21. *For all functions f,*

1. $\mathrm{D}(f) \geq \mathrm{C}(f) \geq \mathrm{bs}(f)$.
2. $\mathrm{R}(f) = \Omega(\mathrm{fbs}(f))$.
3. $\mathrm{Q}(f) = \Omega(\sqrt{\mathrm{fbs}(f)})$.

The lower bound relationships between the measures are depicted in Figure 4.2. Since $\mathrm{fbs}(f) \leq \mathrm{C}(f) \leq \mathrm{bs}(f)^2$ [BdW02], the measure $\sqrt{\mathrm{fbs}(f)}$ is a lower bound on $\mathrm{bs}(f)$.

One-Sided Measures. For Boolean functions with $H = \{0, 1\}$, for each measure M from $\mathrm{bs}(f), \mathrm{fbs}(f), \mathrm{FC}(f), \mathrm{C}(f)$ and a Boolean value $b \in \{0, 1\}$, define the corresponding one-sided measure as

$$M^b(f) = \max_{x \in f^{-1}(b)} M(f, x).$$

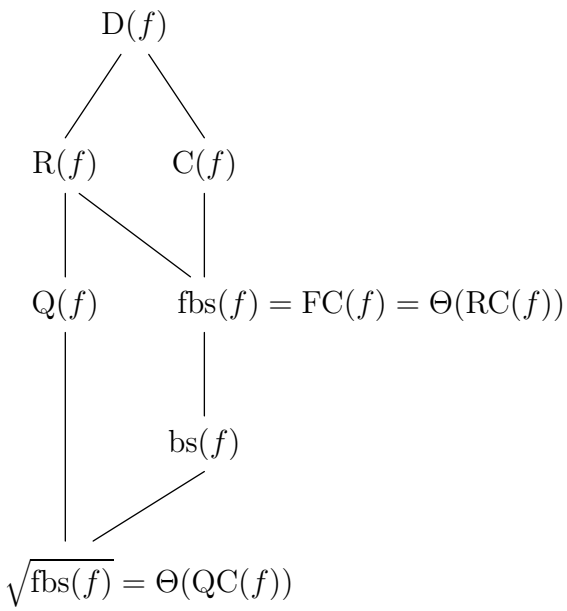


Figure 4.2: The lower bound hierarchy. A smaller measure is placed below the larger measure.

According to the earlier definitions, we then have $M(f) = \max\{M^0(f), M^1(f)\}$. These one-sided measures are useful when, for example, working with compositions of OR with some Boolean function.

Chapter 5

Classical Adversary Bounds

In this chapter we describe the results concerning the classical adversary bounds. They are based on the following paper:

[AKPV18] Andris Ambainis, Martins Kokainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. All Classical Adversary Methods are Equivalent for Total Functions. In *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*, volume 96 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

5.1 Introduction

The adversary method is a versatile general lower bound technique that originates from quantum query complexity. It was introduced by Ambainis in [Amb00] and since then found a wide range of applications. Since then, many generalizations of the quantum adversary method have been introduced (see [ŠS06] for a list of known quantum adversary bounds).

Several of these formulations have been lifted back to the randomized setting. Aaronson proved a classical analogue of Ambainis' relational adversary bound and used it to provide a lower bound for the local search problem [Aar06]. Laplante and Magniez introduced the Kolmogorov complexity adversary bound for both quantum and classical settings and showed that it subsumes many other adversary techniques. [LM04]. They also gave a classical variation of Ambainis' adversary bound in a different way than Aaronson. Some of the other adversary methods like spectral adversary have not been generalized back to the randomized setting.

While some relations between the adversary bounds had been known before, Špalek and Szegedy proved that practically all known quantum adversary methods are in fact equivalent [ŠS06] (this excludes the general quantum adversary bound, which gives an exact estimate on quantum query complexity for all Boolean functions [HLŠ07, Rei09]). This result cannot be immediately generalized to the classical setting, as the equivalence follows through the spectral adversary which has no classical analogue.

One could hope for a similar equivalence result to also hold for the classical adversary bounds. Some relations have been known before this work: Laplante and Magniez have shown that the Kolmogorov complexity lower bound is at least as strong as Aaronson's relational and Ambainis' weighted adversary bounds [LM04]. Jain and Klauck have noted that the minimax over probability distributions adversary bound is at most $\mathrm{C}(f)$ for total functions

[JK10]. In general, the relationships among the classical adversary bounds remained unclear until this point.

5.2 Quantum Adversary Bound

The following is the original formulation of Ambainis' adversary bound $\mathrm{Adv}(f)$. It is a general quantum lower bound, and also is often called *positive adversary bound*. There are multiple equivalent definitions of this method, listed in [ŠS06]. This measure should not be confused with the *general adversary bound* $\mathrm{Adv}^{\pm}(f)$, which precisely characterizes quantum query complexity, $\mathrm{Q}(f) = \Theta(\mathrm{Adv}^{\pm}(f))$ [HLŠ07, Rei09].

Theorem 22 ([Amb00]). *Let $f : S \to H$ be any function, where $S \subseteq G^n$. Let $R : S \times S \to \mathbb{R}_{\geq 0}$ be a real-valued function such that $R(x, y) = R(y, x)$ for all $x, y \in S$ and $R(x, y) = 0$ whenever $f(x) = f(y)$. Then for $x \in S$ and an index i, let*

$$\theta(x, i) = \frac{\sum_{y \in S} R(x, y)}{\sum_{y \in S : x_i \neq y_i} R(x, y)},$$

where $\theta(x, i)$ is undefined if the denominator is 0. Denote

$$\mathrm{Adv}(f) = \max_{R} \min_{\substack{x,y \in S, i \in [n]: \\ R(x,y) > 0, x_i \neq y_i}} \sqrt{\theta(x, i)\theta(y, i)}.$$

Then $\mathrm{Q}(f) = \Omega(\mathrm{Adv}(f))$.

To use this lower bound, it is enough to construct an appropriate weight relation $R(\cdot, \cdot)$. Consider the following simple application that shows the optimality of Grover's search. It is a quantum algorithm that, given n items, any of which can be either marked or unmarked, determines the presence of a marked item in time $O(\sqrt{n})$ [Gro96].

> **Example 4.** Let $\textsc{Or}(x_1, \ldots, x_n)$ be the logical OR function on n bits. Let e_i be a string that has a single 1 in position i. Consider the weight scheme $R(0^n, \mathrm{e}_i) = 1$ for all Hamming weight 1 strings e_i, and $R(x, y) = 0$ for all other pairs of inputs.
>
> It follows that $\theta(0^n, i) = n/1 = n$ for all i and $\theta(\mathrm{e}_i, i) = 1/1 = 1$. Therefore, we have $\mathrm{Adv}(\textsc{Or}) \geq \sqrt{n \cdot 1} = \sqrt{n}$, and $\mathrm{Q}(\textsc{Or}) = \Omega(\sqrt{n})$.

5.3 Classical Adversary Bounds

In this section we list the previously known four lower bound methods called the *classical adversary bounds*. Here, $f : S \to H$ be any function, where $S \subseteq G^n$. The following are all known to be lower bounds on bounded-error randomized query complexity.

Relational adversary bound. The following is the first classical generalization of the quantum adversary bound introduced by Aaronson [Aar06]. Define R and θ in the same way as in Theorem 22. Then the classical relational adversary is given by

$$\mathrm{CRA}(f) = \max_{R} \min_{\substack{x,y\in S, i\in[n]: \\ R(x,y)>0, x_i\neq y_i}} \max\{\theta(x,i), \theta(y,i)\}.$$

One of the main applications of this bound has been to give optimal lower bounds on the local search problem. In this problem, we are given a graph $G = (V, E)$ and a function $F : V \to \mathbb{N}$. The task is to find a vertex v such that $F(v) \leq F(u)$ for all neighbours of v. In other words, we need to find a local minimum in the graph.

The local search problem models many combinatorial optimization problems and has many applications in complexity theory, physical systems and quantum adiabatic algorithms [AL97, Aar06]. The relational lower bound gives tight lower bounds for a number of graphs G for both classical and quantum algorithms. Consider that $F(x)$ is accessed by a black-box query oracle, and that G is the Boolean hypercube $\{0,1\}^n$. Zhang proved tight lower bounds for this problem using the classical and quantum relational adversary bounds [Zha09]:

$$\mathrm{R}(\textsc{Local Search}) = \Omega(2^{n/2}\sqrt{n}), \qquad \mathrm{Q}(\textsc{Local Search}) = \Omega(2^{n/3}n^{1/6}).$$

Aaronson has also used the relational adversary to give a lower bound for the problem of inverting a permutation.

Example 5. In the permutation inversion decision problem, we are given a permutation $\pi(1), \ldots, \pi(n)$, and the task is to determine whether $\pi^{-1}(1) \leq n/2$. Permutation elements $\pi(i)$ can be queried by a black-box oracle.

Let X be the set of permutations x such that $x^{-1}(1) \leq n/2$ and Y be the set of permutations y such that $y^{-1}(1) > n/2$. Let $R(x, y) = 1$ iff x and y dffer in exactly 2 positions (namely, $x^{-1}(1)$ and $y^{-1}(1)$). Consider any $x \in X$. The number of $y \in Y$ such that $R(x, y) = 1$ is $n/2$, since there are $n/2$ choices for $y^{-1}(1)$.

Now consider any i such that $i > n/2$. Then for any $x \in X$ there is a single $y \in Y$ such that $x(i) \neq y(i)$. Hence $\theta(x, i) = n/2$. Similarly we prove that if $i \leq n/2$, then $\theta(y, i) = n/2$. Therefore, $\max\{\theta(x, i), \theta(y, i)\} \geq n/2$ for any $x \in X, y \in Y$ and $i \in [n]$. It follows that permutation inversion requires $\Omega(n)$ queries to solve using a randomized algorithm.

Weighted adversary bound [Amb03a, LM04]. Let w, w' be weight schemes as follows.

- Every pair $(x, y) \in S^2$ is assigned a non-negative weight $w(x, y) = w(y, x)$ such that $w(x, y) = 0$ whenever $f(x) = f(y)$.
- Every triple (x, y, i) is assigned a non-negative weight $w'(x, y, i)$ such that $w'(x, y, i) = 0$ whenever $x_i = y_i$ or $f(x) = f(y)$, and $w'(x, y, i), w'(y, x, i) \geq w(x, y)$ for all x, y, i such that $x_i \neq y_i$.

For all x, i, let $wt(x) = \sum_{y\in S} w(x, y)$ and $v(x, i) = \sum_{y\in S} w'(x, y, i)$. Denote

$$\mathrm{CWA}(f) = \max_{w,w'} \min_{\substack{x,y\in S, i\in[n] \\ w(x,y)\neq 0, x_i\neq y_i}} \max\left\{\frac{wt(x)}{v(x,i)}, \frac{wt(y)}{v(y,i)}\right\}.$$

Kolmogorov complexity [LM04]. To define the next adversary bound, we need to use the notion of Kolmogorov complexity. A set of strings $\mathcal{S} \subset \{0,1\}^*$ is called *prefix-free* if there are no two strings in $\mathcal{S}$ such that one is a proper prefix of the other. Equivalently we can think of the strings as programs for the Turing machine. Let M be a universal Turing machine and fix a prefix-free set $\mathcal{S}$. The prefix-free *Kolmogorov complexity* of x given y, is defined as the length of the shortest program from $\mathcal{S}$ that prints x when given y:

$$K(x|y) = \min\{|P| \mid P \in \mathcal{S}, M(P,y) = x\}.$$

For a detailed introduction on Kolmogorov complexity, we refer the reader to [LV08].

Let $\sigma \in \{0,1\}^*$ be any finite string.[1] Denote

$$\mathrm{CKA}(f) = \min_{\sigma} \max_{\substack{x,y \in S \\ f(x) \neq f(y)}} \frac{1}{\sum_{i:x_i \neq y_i} \min\{2^{-K(i|x,\sigma)}, 2^{-K(i|y,\sigma)}\}}.$$

Minimax over probability distributions [LM04]. Let $\{p_x\}_{x \in S}$ be a set of probability distributions over $[n]$. Denote

$$\mathrm{CMM}(f) = \min_{p} \max_{\substack{x,y \in S \\ f(x) \neq f(y)}} \frac{1}{\sum_{i:x_i \neq y_i} \min\{p_x(i), p_y(i)\}}.$$

5.4 Overview of the Results

- First, we introduce a new lower bound we call *rank-1 relational adversary bound* (Section 5.5). This is a restricted version of the relational adversary bound, which never exceeds $\mathrm{CRA}(f)$. More specifically, we require that the relation matrix between the inputs has rank 1, and denote this by $\mathrm{CRA}_1(f)$. This lower bound is much easier to calculate, since there are only $|S|$ unknowns in a rank 1 relation matrix, whereas there can be $|S| \times |S|$ unknowns in the general case.

- Our main result shows that the classical adversary bounds are all equivalent for total functions (Section 5.6). Surprisingly, in that case they are equivalent to the fractional block sensitivity $\mathrm{fbs}(f)$. More specifically, we show that the following set of relations hold for any function f:

  $$\mathrm{fbs}(f) \leq \mathrm{CRA}_1(f) \leq \mathrm{CRA}(f) = \mathrm{CWA}(f) \leq \mathrm{CKA}(f) = \Theta(\mathrm{CMM}(f)),$$

  and that $\mathrm{CMM}(f) \leq \mathrm{fbs}(f)$ for total functions. Therefore, for total functions $\mathrm{CRA}(f) = \Theta\left(\mathrm{CRA}_1(f)\right)$, where the latter is much easier to calculate for Boolean functions.

  All this shows that $\mathrm{fbs}(f)$ is a fundamental lower bound measure for total functions with many different formulations, including the previously known $\mathrm{FC}(f)$ and $\mathrm{RC}(f)$. Another interesting corollary is that since the quantum certificate complexity $\mathrm{QC}(f) = \Theta(\sqrt{\mathrm{RC}(f)})$ is a lower bound on the quantum query complexity [Aar08], we have that by taking the square root of any of the adversary bounds above, we obtain a quantum lower bound for total functions.

[1] By the argument of [ŠS06], we take the minimum over the strings instead of the algorithms computing f.

- Lastly, we study the relationship between adversary bounds for partial functions. We show examples of functions where different adversary bounds give asymptotically distinct estimates (Section 5.7).
  - There is a function f such that $\mathrm{fbs}(f) = O(1)$ and $\mathrm{CRA}_1(f) = \Omega(n)$.
  - There is a function f such that $\mathrm{CRA}(f) = O(1)$ and $\mathrm{CKA}(f) = \Omega(\log n)$.
  - There is a function f such that $\mathrm{CRA}_1(f) = O(\sqrt{n})$ and $\mathrm{CRA}(f) = \Omega(n)$. This particular result was proved by Aleksejs Zajakins in his Bachelor's course thesis [Zaj18] under supervision of the author of this thesis.

  Thus we relationship between classical adversary bound is understood for both total and partial functions.

5.5 Rank-1 Relational Adversary Bound

We introduce the following new restriction of the relational adversary bound. Let R' be any $|S| \times |S|$ matrix of rank 1, such that:

- There exist $u, v : S \to \mathbb{R}_{\geq 0}$ such that $R'(x,y) = u(x)v(y)$ for all $x, y \in S$.
- $R'(x,y) = 0$ whenever $f(x) = f(y)$.

Then set $R(x,y) = \max\{R'(x,y), R'(y,x)\}$.

Let $X = \{x \mid u(x) > 0\}$ and $Y = \{y \mid v(y) > 0\}$. Note that for every $x \in S$, either $u(x)$ or $v(x)$ must be 0, as $R(x,x)$ must be 0, therefore $X \cap Y = \varnothing$. Then denote

$$\mathrm{CRA}_1(f) = \max_{u,v} \min_{\substack{x\in X, y\in Y, i\in[n]: \\ u(x)v(y)>0, x_i\neq y_i}} \max\{\theta(x,i), \theta(y,i)\}.$$

where $\theta(x,i)$ can be simplified to

$$\theta(x,i) = \frac{\sum_{y\in Y} v(y)}{\sum_{y\in Y: x_i\neq y_i} v(y)} \qquad \text{and} \qquad \theta(y,i) = \frac{\sum_{x\in X} u(x)}{\sum_{x\in X: x_i\neq y_i} u(x)}.$$

Naturally, $\mathrm{CRA}_1(f) \leq \mathrm{CRA}(f)$. The rank-1 adversary bound is much easier to use, since it is sufficient to find just two weight vectors u and v.

Alternative definition. As $R(x,y) = 0$ whenever $f(x) = f(y)$, we have that for every output $h \in H$ either $f^{-1}(h) \cap X = \varnothing$ or $f^{-1}(h) \cap Y = \varnothing$. Therefore, $\mathrm{CRA}_1(f)$ effectively bounds the complexity of differentiating between two non-overlapping sets of outputs. This leads to the following equivalent definition for $\mathrm{CRA}_1(f)$:

Proposition 23. *Let $A \cup B = H$ be a partition of the output alphabet, i.e., $A \cap B = \varnothing$. Let p and q be probability distributions over $X := f^{-1}(A)$ and $Y := f^{-1}(B)$, respectively. Then*

$$\mathrm{CRA}_1(f) = \max_{\substack{A,B \\ p,q}} \min_{\substack{i\in[n], \\ g_1,g_2\in G: g_1\neq g_2 \\ \exists x\in X, y\in Y: p(x)q(y)>0}} \frac{1}{\min\{\Pr_{x\sim p}[x_i \neq g_1], \Pr_{y\sim q}[y_i \neq g_2]\}}.$$

Proof. Let u, v be vectors that maximize $\mathrm{CRA}_1(f)$. Let $h \in H$ be any letter and $S_h = f^{-1}(h)$. Since for every x, y, such that $f(x) = f(y)$, we have $u(x)v(y) = 0$, it follows that either $u(x) = 0$ for all $x \in S_h$ or $v(x) = 0$ for all $x \in S_h$. Therefore, we can find a partition $A \cup B = H$ such that:

- if $u(x) > 0$, then $f(x) \in A$;
- if $v(x) > 0$, then $f(y) \in B$;
- for every $h \in H$, either $h \in A$ or $h \in B$.

This partition therefore also defines a partition of the inputs, $X \cup Y = S$, where $X = f^{-1}(A)$ and $Y = f^{-1}(B)$.

Now, notice that $\theta(x, i)$ does not depend on the particular choice of x if $x_i := g_1 \in G$ is fixed. Similarly, let $y_i := g_2 \in G$ be fixed, then $\theta(y, i)$ does not depend on the particular choice of y. This allows to simplify the expression for $\mathrm{CRA}_1(f)$, since for each i we can fix values $g_1 \neq g_2$ (such that there exist $x \in X$, $y \in Y$ with $u(x)v(y) > 0$ and $x_i = g_1$ and $y = g_2$) and ignore the remaining components of x, y, i.e.,

$$\mathrm{CRA}_1(f) = \max_{A,B:A\cup B=H} \max_{u,v} \min_{\substack{i\in[n],\\ g_1,g_2\in G,g_1\neq g_2:\\ \exists x\in X,y\in Y:\\ x_i=g_1,y_i=g_2,\\ u(x)v(y)>0}} \max\left\{ \frac{\sum_{y\in Y} v(y)}{\sum_{y\in Y:y_i\neq g_1} v(y)}, \frac{\sum_{x\in X} u(x)}{\sum_{x\in X:x_i\neq g_2} u(x)} \right\}.$$

Further assume that both X and Y are non-empty, because otherwise the value of CRA_1 would not be defined. Notice that multiplying either u or v with any scalar does not affect the value of CRA_1. Hence, we can scale u and v to probability distributions p and q over X and Y, respectively. More specifically, we can further simplify CRA_1:

$$\begin{aligned}
\mathrm{CRA}_1(f) &= \max_{\substack{A,B:\\ A\cup B=H}} \max_{p,q} \min_{\substack{i\in[n],\\ g_1,g_2\in G:g_1\neq g_2:\\ \exists x\in X,y\in Y:\\ x_i=g_1,y_i=g_2,\\ p(x)q(y)>0}} \frac{1}{\min\left\{ \sum_{\substack{y\in Y\\ y_i\neq g_1}} q(y), \sum_{\substack{x\in X\\ x_i\neq g_2}} p(x) \right\}} \\
&= \max_{\substack{A,B:\\ A\cup B=H}} \max_{p,q} \min_{\substack{i\in[n],\\ g_1,g_2\in G:g_1\neq g_2:\\ \exists x\in X,y\in Y:\\ x_i=g_1,y_i=g_2,\\ p(x)q(y)>0}} \frac{1}{\min\left\{ \Pr_{y\sim q}[y_i \neq g_1], \Pr_{x\sim p}[x_i \neq g_2] \right\}}.
\end{aligned}$$

□

Boolean functions. We can further simplify this definition if the inputs are Boolean:

Proposition 24. *Let $f : S \to H$, where $S \subseteq \{0,1\}^n$. Let $A \cup B = H$ be a partition of the output alphabet, i.e., $A \cap B = \varnothing$. Let p and q be probability distributions over $X := f^{-1}(A)$ and $Y := f^{-1}(B)$, respectively. Then*

$$\mathrm{CRA}_1(f) = \max_{\substack{A,B,\\ p,q}} \min_{\substack{i\in[n],\\ b\in\{0,1\}}} \frac{1}{\min\left\{ \Pr_{y\sim q}[y_i \neq b], \Pr_{x\sim p}[x_i = b] \right\}}.$$

Proof. For $g_1, g_2 \in \{0,1\}$, $g_1 \neq g_2$ implies $g_2 = g_1 \oplus 1$. It follows that

$$\mathrm{CRA}_1(f) = \max_{\substack{A,B,\\ p,q}} \min_{\substack{i\in[n],\\ b\in\{0,1\}:\\ \exists x\in X, y\in Y:\\ x_i=b, y_i\neq b,\\ p(x)q(y)>0}} \frac{1}{\min\{\Pr_{y\sim q}[y_i \neq b], \Pr_{x\sim p}[x_i = b]\}}.$$

Moreover, we can drop the requirement $\exists x \in X, y \in Y : x_i = b, y_i \neq b, p(x)q(y) > 0$. To see that, fix any p, q, and consider the quantities

$$\alpha = \max_{i\in[n]} \max_{\substack{b\in\{0,1\}:\\ \exists x\in X, y\in Y:\\ x_i=b, y_i\neq b,\\ p(x)q(y)>0}} \min\left\{\Pr_{x\sim p}[x_i = b], \Pr_{y\sim q}[y_i \neq b]\right\}$$

$$\beta = \max_{\substack{i\in[n]\\ b\in\{0,1\}}} \min\left\{\Pr_{x\sim p}[x_i = b], \Pr_{y\sim q}[y_i \neq b]\right\}.$$

Clearly, $\alpha \leq \beta$. To show the converse inequality, consider any $i \in [n]$ and (if such exists) $b \in \{0,1\}$ satisfying $u(x)v(y) = 0$ for any $x \in X, y \in Y$ with $x_i = b$, $y_i \neq b$ (to deal with the possibility no such x, y exist, we consider the empty sum to be zero). Then also

$$0 = \sum_{\substack{x\in X, y\in Y\\ x_i=b,\ y_i\neq b}} p(x)q(y) = \left(\sum_{x\in X: x_i=b} p(x)\right)\left(\sum_{y\in Y: y_i\neq b} q(y)\right) = \Pr_{x\sim p}[x_i = b] \cdot \Pr_{y\sim q}[y_i \neq b].$$

Therefore, $\min\{\Pr_{x\sim p}[x_i = b], \Pr_{y\sim q}[y_i \neq b]\} = 0 \leq \alpha$. Thus $\alpha = \beta$. Thus the claim follows. □

5.6 Equivalence of the Adversary Bounds

In this section we prove the main theorem:

Theorem 25. *Let $f : S \to H$ be a partial Boolean function, where $S \subseteq G^n$. Then*

- $\mathrm{fbs}(f) \leq \mathrm{CRA}_1(f) \leq \mathrm{CRA}(f) = \mathrm{CWA}(f)$,
- $\mathrm{CWA}(f) = O(\mathrm{CKA}(f))$,
- $\mathrm{CKA}(f) = \Theta(\mathrm{CMM}(f))$.

Moreover, for total functions $f : G^n \to H$, we have

$$\mathrm{fbs}(f) = \mathrm{CMM}(f).$$

The part $\mathrm{CWA}(f) = O(\mathrm{CKA}(f))$ has been already proven in [LM04].

5.6.1 Fractional Block Sensitivity and the Weighted Adversary Method

First, we prove that fractional block sensitivity lower bounds the relational adversary bound for any partial function.

Proposition 26. *Let $f : S \to H$ be a partial Boolean function, where $S \subseteq G^n$. Then*

$$\mathrm{fbs}(f) \leq \mathrm{CRA}_1(f).$$

Proof. Let $x \in S$ be such that $\mathrm{fbs}(f, x) = \mathrm{fbs}(f)$ and denote $h = f(x)$. Let $H' = H \setminus \{h\}$ and $S' = f^{-1}(H')$.

Let $\mathcal{B}$ be the set of sensitive blocks of x. Let $w : \mathcal{B} \to [0, 1]$ be an optimal solution to the $\mathrm{fbs}(f, x)$ linear program, that is, $\sum_{B \in \mathcal{B}} w(B) = \mathrm{fbs}(f, x)$. For each $B \in \mathcal{B}$, pick a single $y_B \in S'$ such that $B = \{i \mid x_i \neq y_i\}$. Then define $R(x, y_B) := w(B)$ for all $B \in \mathcal{B}$. It is clear that R has a corresponding rank 1 matrix R', as it has only one row (corresponding to x) that is not all zeros.

Let $y \in S'$ be any input such that $R(x, y) > 0$. Then for any $i \in [n]$ such that $x_i \neq y_i$,

$$\theta(x, i) = \frac{\sum_{B \in \mathcal{B}} w(B)}{\sum_{B \in \mathcal{B}: i \in B} w(B)} = \frac{\mathrm{fbs}(f, x)}{\sum_{B \in \mathcal{B}: i \in B} w(B)} \geq \mathrm{fbs}(f),$$

as $0 < \sum_{B \in \mathcal{B}: i \in B} w(B) \leq 1$. On the other hand, note that

$$\theta(y, i) = \frac{w(B)}{w(B)} = 1,$$

where $B = \{i \mid x_i \neq y_i\}$. Therefore, for this R,

$$\min_{\substack{x,y \in S, i \in [n]: \\ R(x,y) > 0, x_i \neq y_i}} \max\{\theta(x, i), \theta(y, i)\} \geq \min_{\substack{y \in S', i \in [n]: \\ R(x,y) > 0, x_i \neq y_i}} \max\{\mathrm{fbs}(f), 1\} = \mathrm{fbs}(f),$$

and the claim follows. □

As mentioned in [LM04], $\mathrm{CRA}(f)$ is a weaker version of $\mathrm{CWA}(f)$. We show that in fact they are exactly equal to each other:

Proposition 27. *Let $f : S \to H$ be a partial Boolean function, where $S \subseteq G^n$. Then*

$$\mathrm{CRA}(f) = \mathrm{CWA}(f).$$

Proof.

- First we show that $\mathrm{CRA}(f) \leq \mathrm{CWA}(f)$.

  Suppose that R is the function for which the relational bound achieves maximum value. Let $w(x, y) = w(y, x) = w(x, y, i) = w(y, x, i) = R(x, y)$ for any x, y, i such that $f(x) \neq f(y)$ and $x_i \neq y_i$. This pair of weight schemes satisfies the conditions of the weighted adversary bound. The value of the latter with w, w' is equal to $\mathrm{CRA}(f)$. As the weighted adversary bound is a maximization measure, $\mathrm{CRA}(f) \leq \mathrm{CWA}(f)$.

- Now we show that $\mathrm{CRA}(f) \geq \mathrm{CWA}(f)$.

  Let w, w' be optimal weight schemes for the weighted adversary bound. Let $R(x,y) = w(x,y)$ for any $x, y \in S$ such that $f(x) \neq f(y)$. Let $S' = f^{-1}(H \setminus f(x))$. Then

$$\theta(x,i) = \frac{\sum_{y \in S'} R(x,y)}{\sum_{y \in S' : x_i \neq y_i} R(x,y)} = \frac{\sum_{y \in S'} w(x,y)}{\sum_{y \in S' : x_i \neq y_i} w(x,y)} \geq \frac{\sum_{y \in S'} w(x,y)}{\sum_{y \in S' : x_i \neq y_i} w'(x,y,i)} = \frac{wt(x)}{v(x,i)},$$

  as $w'(x,y,i) \geq w(x,y)$ by the properties of w, w'. Similarly, $\theta(y,i) \geq \frac{wt(y)}{v(y,i)}$. Therefore, for any $x, y \in S$ and $i \in [n]$ such that $f(x) \neq f(y)$ and $x_i \neq y_i$, we have

$$\max\{\theta(x,i), \theta(y,i)\} \geq \max\left\{\frac{wt(x)}{v(x,i)}, \frac{wt(y)}{v(y,i)}\right\}.$$

  As the relational adversary bound is also a maximization measure, $\mathrm{CRA}(f) \geq \mathrm{CWA}(f)$. □

The proof of this proposition also shows why $\mathrm{CRA}(f)$ and $\mathrm{CWA}(f)$ are equivalent — the weight function w' is redundant in the classical case (in contrast to the quantum setting).

5.6.2 Kolmogorov Complexity and Minimax over Distributions

In this section we prove the equivalence between the mimimax over probability distributions and Kolmogorov complexity adversary bound. It has been shown in the proof of the main theorem of [LM04] that $\mathrm{CMM}(f) = \Omega(\mathrm{CKA}(f))$. Here we show the other direction using a well-known result from coding theory.

Proposition 28 (Kraft's inequality)**.** *Let S be any prefix-free set of finite strings. Then*

$$\sum_{x \in S} 2^{-|x|} \leq 1.$$

Proposition 29. *Let $f : S \to H$ be a partial Boolean function, where $S \subseteq G^n$. Then*

$$\mathrm{CKA}(f) \geq \mathrm{CMM}(f).$$

Proof. Let σ be the binary string for which $\mathrm{CKA}(f)$ achieves the smallest value. Define the set of probability distributions $\{p_x\}_{x \in S}$ on $[n]$ as follows. Let $s_x = \sum_{i \in [n]} 2^{-K(i|x,\sigma)}$ and $p_x(i) = 2^{-K(i|x,\sigma)}/s_x$. The set of programs that print out $i \in [n]$, given x and σ, is prefix-free (by the definition of $\mathcal{S}$), as the information given to all programs is the same. Thus by Kraft's inequality, we have $s_x \leq 1$.

Examine the value of the minimax bound with this set of probability distributions. For any $x, y \in S$ and $i \in [n]$, we have

$$\min\{p_x(i), p_y(i)\} = \min\left\{\frac{2^{-K(i|x,\sigma)}}{s_x}, \frac{2^{-K(i|y,\sigma)}}{s_y}\right\} \geq \min\{2^{-K(i|x,\sigma)}, 2^{-K(i|y,\sigma)}\}.$$

Therefore, $\mathrm{CKA}(f) = \Theta(\mathrm{CMM}(f))$. □

5.6.3 Fractional Block Sensitivity and Minimax over Distributions

Now we proceed to prove that for total functions, fractional block sensitivity is equal to the minimax over probability distributions. The latter has an equivalent form of the following program.

Lemma 30. *For any partial Boolean function $f : S \to H$, where $S \subseteq G^n$,*

$$\mathrm{CMM}(f) = \min_{v} \max_{x \in S} \sum_{i \in [n]} v_x(i)$$

s.t.

$$\forall y \in S \text{ s.t. } f(x) \neq f(y) : \sum_{i : x_i \neq y_i} \min\{v_x(i), v_y(i)\} \geq 1,$$

where $\{v_x\}_{x \in S}$ is any set of weight functions $v_x : [n] \to \mathbb{R}_{\geq 0}$.

Proof. Denote by μ the optimal value of the given program.

- First we prove that $\mu \leq \mathrm{CMM}(f)$.

  Construct a set of weight functions $\{v_x\}_{x \in S}$ by $v_x(i) := p_x(i) \cdot \mathrm{CMM}(f)$, where $\{p_x\}_{x \in S}$ is an optimal set of probability distributions for the minimax bound. Then for any x, y such that $f(x) \neq f(y)$,

  $$\sum_{i : x_i \neq y_i} \min\{v_x(i), v_y(i)\} = \mathrm{CMM}(f) \cdot \sum_{i : x_i \neq y_i} \min\{p_x(i), p_y(i)\} \geq \mathrm{CMM}(f) \cdot \frac{1}{\mathrm{CMM}(f)} = 1.$$

  On the other hand, the value of this solution is given by

  $$\max_{x \in S} \sum_{i \in [n]} v_x(i) = \max_{x \in S} \mathrm{CMM}(f) \cdot \sum_{i \in [n]} p_x(i) = \mathrm{CMM}(f).$$

- Now we prove that $\mu \geq \mathrm{CMM}(f)$.

  Let $\{v_x\}_{x \in S}$ be an optimal solution for the given program. Set $s_x = \sum_{i \in [n]} v_x(i)$. Construct a set of probability distributions $\{p_x\}_{x \in S}$ by $p_x(i) = v_x(i)/s_x$. Then for any x, y such that $f(x) \neq f(y)$, we have

  $$\sum_{i : x_i \neq y_i} \min\{p_x(i), p_y(i)\} = \sum_{i : x_i \neq y_i} \min\left\{\frac{v_x(i)}{s_x}, \frac{v_y(i)}{s_y}\right\} \geq \frac{1}{\mu} \cdot \sum_{i : x_i \neq y_i} \min\{v_x(i), v_y(i)\} \geq \frac{1}{\mu}.$$

  Therefore, $\mathrm{CMM}(f) \leq \mu$. □

In this case we prove that for total functions the minimax over probability distributions is equal to the fractional certificate complexity $\mathrm{FC}(f)$. The result follows since $\mathrm{FC}(f) = \mathrm{fbs}(f)$. The proof of this claim is almost immediate in light of the following "fractional certificate intersection" lemma by Kulkarni and Tal:

Proposition 31 ([KT16], Lemma 6.2)**.** *Let $f : G^n \to H$ be a total function*[2] *and $\{v_x\}_{x \in G^n}$ be a feasible solution for the $\mathrm{FC}(f)$ linear program. Then for any two inputs $x, y \in G^n$ such that $f(x) \neq f(y)$, we have*

$$\sum_{i : x_i \neq y_i} \min\{v_x(i), v_y(i)\} \geq 1.$$

[2]Kulkarni and Tal prove the lemma for Boolean functions, but it is straightforward to check that their proof also works for functions with arbitrary input and output alphabets.

Let f be a total function. Suppose that $\{v_x\}_{x\in G^n}$ is a feasible solution for the $\mathrm{CMM}(f)$ program. Then for any $x, y \in G^n$ such that $f(x) \neq f(y)$,

$$\sum_{i:x_i\neq y_i} v_x(i) \geq \sum_{i:x_i\neq y_i} \min\{v_x(i), v_y(i)\} \geq 1.$$

Hence this is also a feasible solution for the $\mathrm{FC}(f)$ linear program. On the other hand, if $\{v_x\}_{x\in G^n}$ is a feasible solution for $\mathrm{FC}(f)$ linear program, then it is also a feasible solution for the $\mathrm{CMM}(f)$ program by Proposition 31. Therefore, $\mathrm{CMM}(f) = \mathrm{FC}(f)$.

5.7 Separations for Partial Functions

In this section, we show examples of partial Boolean functions that give asymptotic separations between different classical adversary bounds.

5.7.1 Fractional Block Sensitivity vs. Adversary Bounds

Here we show an example of a partial function that provides an unbounded separation between the adversary measures and fractional block sensitivity.

Theorem 32. *There exists a partial Boolean function $f : S \to \{0,1\}$, where $S \subseteq \{0,1\}^n$, such that $\mathrm{fbs}(f) = O(1)$ and $\mathrm{CRA}_1(f), \mathrm{CRA}(f), \mathrm{CWA}(f), \mathrm{CKA}(f), \mathrm{CMM}(f) = \Omega(n)$.*

Proof. Let n be an even number and $S = \{x \in \{0,1\}^n \mid |x| = 1\}$ be the set of bit strings of Hamming weight 1. Define the "greater than half" function $\textsc{Gth}_n : S \to \{0,1\}$ to be 1 iff $x_i = 1$ for $i > n/2$.

For the first part, the certificate complexity is constant $\mathrm{C}(\textsc{Gth}_n) = 1$. To certify the value of greater than half, it is enough to certify the position of the unique i such that $x_i = 1$. The claim follows, as $\mathrm{C}(f) \geq \mathrm{fbs}(f)$ for any f.

For the second part, by Theorem 25, it suffices to show that $\mathrm{CRA}_1(\textsc{Gth}_n) = \Omega(n)$. Let $X = f^{-1}(0)$ and $Y = f^{-1}(1)$. Let $R(x,y) = 1$ for all $x \in X, y \in Y$. Suppose that $x \in X, y \in Y, i \in [n]$ are such that $x_i = 1$ (and thus $y_i = 0$). Then

$$\theta(x,i) = \frac{\sum_{y^*\in Y} R(x,y^*)}{\sum_{y^*\in Y:x_i\neq y_i^*} R(x,y^*)} = \frac{n/2}{n/2} = 1,$$
$$\theta(y,i) = \frac{\sum_{x^*\in X} R(x^*,y)}{\sum_{x^*\in X:x_i^*\neq y_i} R(x^*,y)} = \frac{n/2}{1} = n/2.$$

Therefore, $\max\{\theta(x,i), \theta(y,i)\} = n/2$. Similarly, if i is such an index that $y_i = 1$ and $x_i = 0$, we also have $\max\{\theta(x,i), \theta(y,i)\} = n/2$. Also note that R has a corresponding rank 1 matrix R', hence $\mathrm{CRA}_1(f) \geq n/2 = \Omega(n)$. □

We note that this function is essentially the Boolean variant of the permutation inversion. By construction, a lower bound on $\textsc{Gth}_n$ also gives a lower bound on inverting a permutation.

5.7.2 Weighted Adversary vs. Kolmogorov Complexity Bound

Here we show that, for a variant of the ordered search problem, the Kolmogorov complexity bound gives a tight logarithmic lower bound, while the weighted adversary gives only a constant value lower bound.

Theorem 33. *There exists a partial Boolean function $f : S \to \{0,1\}$, where $S \subseteq \{0,1\}^n$, such that* $\mathrm{CRA}_1(f), \mathrm{CRA}(f), \mathrm{CWA}(f) = O(1)$ *and* $\mathrm{CKA}(f), \mathrm{CMM}(f) = \Omega(\log n)$.

Proof. Let $S = \{x \in \{0,1\}^n \mid \exists i \in [0;n] : x_1 = \ldots x_i = 0 \text{ and } x_{i+1} = \ldots = x_n = 1\}$. In other words, x is any string starting with some number of 0s followed by all 1s. Define the "ordered search parity" function $\mathrm{OSP}_n : S \to \{0,1\}$ to be $\textsc{Ind}(x)$ mod 2, where $\textsc{Ind}(x)$ is the last index i such that $x_i = 0$ (in the special case $x = 1^n$, assume that $i = 0$).

For simplicity, further assume that n is even. First, we prove that $\mathrm{CKA}(f) = \Omega(\log n)$. We use the argument of Laplante and Magniez and the distance scheme method they have adapted from [HNS02]:

Proposition 34 ([LM04], Theorem 5)**.** *Let $f : S \to \{0,1\}$ be a Boolean function, where $S \subseteq \{0,1\}^n$. Let D be a non-negative integer function on S^2 such that $D(x,y) = 0$ whenever $f(x) = f(y)$. Let $W = \sum_{x,y:D(x,y)\neq 0} \frac{1}{D(x,y)}$. Define the* right load $\mathrm{RL}(x,i)$ *to be the maximum over all values d, of the number of y such that $D(x,y) = d$ and $x_i \neq y_i$. The* left load $\mathrm{LL}(y,i)$ *is defined similarly, inverting x and y. Then*

$$\mathrm{CKA}(f) = \Omega\left(\frac{W}{|S|} \min_{\substack{x,y,i \\ D(x,y)\neq 0, x_i \neq y_i}} \max\left\{\frac{1}{\mathrm{RL}(x,i)}, \frac{1}{\mathrm{LL}(y,i)}\right\}\right).$$

For each pair x, y such that $f(x) \neq f(y)$ and $\textsc{Ind}(x) > \textsc{Ind}(y)$, let $D(x,y) = \textsc{Ind}(x) - \textsc{Ind}(y)$. Then we have

$$W = \sum_{k=1}^{n/2}((n+1) - (2k-1))\frac{1}{2k-1} = (n+1)\sum_{k=1}^{n/2}\frac{1}{2k-1} - \frac{n}{2}.$$

Since $\sum_{k=1}^{n/2} 1/(2k-1) > \sum_{k=1}^{n/2} 1/2k = \frac{1}{2}\cdot\sum_{k=1}^{n/2} 1/k = H_{n/2} = \Theta(\log n)$ as a harmonic number, we have that $W > (n+1)H_{n/2} - n/2 = \Theta(n \log n)$.

On the other hand, since for every $x \in S$ and positive integer d there is at most one y such that $D(x,y) = d$, we have that $\mathrm{RL}(x,i) = \mathrm{LL}(y,i) = 1$ for any x, y such that $f(x) \neq f(y)$ and $x_i \neq y_i$. Since $|S| = n+1$, by Proposition 34,

$$\mathrm{CKA}(\mathrm{OSP}_n) = \Omega\left(\frac{n \log n}{n}\right) = \Omega(\log n).$$

Now we prove that $\mathrm{CRA}(\mathrm{OSP}_n) \leq 2$. Let $N = n/2$; we start by fixing an enumeration of S. By $x^{(i)}$, $i \in [N+1]$, we denote the unique element of S satisfying $\textsc{Ind}(x^{(i)}) = 2i - 2$ (it is a negative input for OSP_n); by $y^{(j)}$, $j \in [N]$, we denote the unique element of S satisfying $\textsc{Ind}(y^{(j)}) = 2j - 1$ (it is a positive input for OSP_n).

We claim that for every $R = (r_{ij})$, $i \in [N+1], j \in [N]$, with nonnegative entries we have

$$\min_{\substack{(i,j)\in[N+1]\times[N]: \\ r_{ij}>0}} \min_{\substack{t\in[n]: \\ x_t^{(i)} \neq y_t^{(j)}}} \max\{\theta(x^{(i)}, t), \theta(y^{(j)}, t)\} \leq 2,$$

unless $r_{ij} = 0$ for all i, j. Since $\mathrm{CRA}(\mathrm{OSP}_n)$ is defined only for R which are not identically zero, we conclude that $\mathrm{CRA}(f) \le 2$.

For all $i \in [N+1]$, $j = [N]$ we set

$$t_{ij} = \min\{t : x_t^{(i)} \neq y_t^{(j)}\} = 1 + \min\{\mathrm{IND}(x^{(i)}), \mathrm{IND}(y^{(j)})\} = \begin{cases} 2i-1, & i \le j, \\ 2j, & i > j. \end{cases}.$$

We shall show that, unless $R \equiv 0$, there is a pair (i, j) satisfying

$$r_{ij} > 0 \quad \text{and} \quad \max\{\theta(x^{(i)}, t_{ij}), \theta(y^{(j)}, t_{ij})\} \le 2. \tag{5.1}$$

Consider $i \in \{2, 3, \dots, N+1\}$ and $j \in [i-1]$. Then we have $t_{ij} = 2j$ and

$$\theta(x^{(i)}, t_{ij}) = \frac{\sum_{k=1}^{N} r_{ik}}{\sum_{k=1}^{j} r_{ik}}, \quad \theta(y^{(j)}, t_{ij}) = \frac{\sum_{l=1}^{N+1} r_{lj}}{\sum_{l=j+1}^{N+1} r_{lj}}. \tag{5.2}$$

Now consider $i \in [N]$ and $j \in \{i, i+1, \dots, N\}$. Then we have $t_{ij} = 2i - 1$ and

$$\theta(x^{(i)}, t_{ij}) = \frac{\sum_{k=1}^{N} r_{ik}}{\sum_{k=i}^{N} r_{ik}}, \quad \theta(y^{(j)}, t_{ij}) = \frac{\sum_{l=1}^{N+1} r_{lj}}{\sum_{l=1}^{i} r_{lj}}. \tag{5.3}$$

We introduce the following notation:

- $\alpha_{ij} = \sum_{k=j+1}^{N} r_{ik}$ and $\beta_{ij} = \sum_{k=1}^{j} r_{ik}$, for $i \in [N+1]$ and $j \in \{0, 1, \dots, i-1\}$;
- $\gamma_{ij} = \sum_{l=i+1}^{N+1} r_{lj}$ and $\delta_{ij} = \sum_{l=1}^{i} r_{lj}$ for $i \in [N]$, $j \in \{i, i+1, \dots, N\}$.

By convention, $\beta_{10} = \alpha_{N+1,N} = 0$. Then (5.2)–(5.3) can be rewritten as follows:

$$\theta(x^{(i)}, t_{ij}) = \begin{cases} 1 + \alpha_{ij}/\beta_{ij}, & j < i, \\ 1 + \beta_{i,i-1}/\alpha_{i,i-1}, & j \ge i, \end{cases} \qquad \theta(y^{(j)}, t_{ij}) = \begin{cases} 1 + \delta_{jj}/\gamma_{jj}, & j < i, \\ 1 + \gamma_{ij}/\delta_{ij}, & j \ge i. \end{cases}$$

Consequently, (5.1) holds if there is a pair $(i, j) \in [N+1] \times [N]$ such that $r_{ij} > 0$ and

$$\begin{cases} (\alpha_{ij} \le \beta_{ij}) \wedge (\delta_{jj} \le \gamma_{jj}), & j < i, \\ (\beta_{i,i-1} \le \alpha_{i,i-1}) \wedge (\gamma_{ij} \le \delta_{ij}), & j \ge i. \end{cases}$$

Suppose the contrary: for all $(i, j) \in [N+1] \times [N]$ we have

$$i > j \Rightarrow (r_{ij} = 0) \vee (\alpha_{ij} > \beta_{ij}) \vee (\delta_{jj} > \gamma_{jj}) \tag{C1}$$

and

$$i \le j \Rightarrow (r_{ij} = 0) \vee (\beta_{i,i-1} > \alpha_{i,i-1}) \vee (\gamma_{ij} > \delta_{ij}). \tag{C2}$$

We shall show by induction that for all $i \in \{0, 1, \dots, N\}$, $j \in [N]$ the following holds:

$$\alpha_{i+1,i} \ge \beta_{i+1,i} \quad \text{and} \quad \gamma_{jj} \ge \delta_{jj}. \tag{5.4}$$

When that is established, it follows that all r_{ij} must be zero. To see that, recall $\alpha_{N+1,N} = 0$. Since (5.4) implies $\beta_{N+1,N} \le \alpha_{N+1,N} = 0$, we obtain $\beta_{N+1,N} = \sum_{k=1}^{N} r_{N+1,k} \le 0$. However,

all r_{lk} are nonnegative, hence $r_{N+1,k} = 0$ for all $k \in [N]$. That, in turn, implies $\sum_{l=1}^{N} r_{lN} = \delta_{NN} \leq \gamma_{NN} = r_{N+1,N} = 0$, where we have used (5.4) again. Now $r_{lN} = 0$ for all $l \in [N]$ (and also for $l = N+1$), thus $\alpha_{N,N-1} = r_{NN} = 0$. Continue inductively to obtain that $\alpha_{i+1,i} = \beta_{i+1,i} = 0$ and $\gamma_{jj} = \delta_{jj} = 0$ (and $r_{ij} = 0$) for all i, j. It remains to show (5.4).

The base case: we already have $\alpha_{10} \geq 0 = \beta_{10}$. For **the inductive step**, suppose that (5.4) holds for all $i \in \{0, 1, \ldots, p-1\}$ and $j \in [p-1]$, for some $p \in [N]$ (for $p = 1$, the inequality $\gamma_{jj} \geq \delta_{jj}$ remains unproven for all j). We shall show that both inequalities hold also with $i = j = p$. The proof is by contradiction.

Suppose that $\gamma_{pp} < \delta_{pp}$. From (C2) it follows that either $r_{pp} = 0$ or $\beta_{p,p-1} > \alpha_{p,p-1}$. The latter is false by the inductive hypothesis, thus $r_{pp} = 0$. But then

$$\gamma_{p-1,p} = \sum_{l=p}^{N+1} r_{lp} = \gamma_{pp} \quad \text{and} \quad \delta_{p-1,p} = \sum_{l=1}^{p-1} r_{lp} = \delta_{pp}.$$

Thus we have $\gamma_{p-1,p} < \delta_{p-1,p}$. Again, from (C2) it follows that either $r_{p-1,p} = 0$ or $\beta_{p-1,p-2} > \alpha_{p-1,p-2}$. The latter is false, thus $r_{p-1,p} = 0$, which implies $\gamma_{p-2,p} = \gamma_{p-1,p} < \delta_{p-1,p} = \delta_{p-2,p}$. Continuing similarly, we obtain $r_{1p} = r_{2p} = \ldots = r_{pp} = 0$. However, then $\delta_{pp} = 0$ and the inequality $\gamma_{pp} < \delta_{pp}$ is impossible, a contradiction.

Suppose that $\beta_{p+1,p} > \alpha_{p+1,p}$. From (C1) it follows that either $r_{p+1,p} = 0$ or $\delta_{pp} > \gamma_{pp}$. As shown previously, the latter is false, thus $r_{p+1,p} = 0$. But then we have

$$\alpha_{p+1,p-1} = \sum_{k=p}^{N} r_{p+1,k} = \alpha_{p+1,p} \quad \text{and} \quad \beta_{p+1,p-1} = \sum_{k=1}^{p-1} r_{p+1,k} = \beta_{p+1,p}.$$

Hence we also have $\beta_{p+1,p-1} > \alpha_{p+1,p-1}$. Then again from (C1) we either have $\delta_{p-1,p-1} > \gamma_{p-1,p-1}$, or $r_{p+1,p-1} = 0$. The former is false by the inductive hypothesis, the latter implies $\beta_{p+1,p-2} = \beta_{p+1,p-1} > \alpha_{p+1,p-1} = \alpha_{p+1,p-2}$. Continuing similarly, we obtain $r_{p+1,1} = \ldots = r_{p+1,p} = 0$. But then $\beta_{p+1,p} = 0 \leq \alpha_{p+1,p}$, a contradiction. This completes the inductive step. □

5.7.3 Rank-1 Relational Adversary vs. Weighted Adversary

The following result was proven by Zajakins [Zaj18]. We describe the function that achieves the separation and state the result without proof.

Theorem 35. *There exists a partial Boolean function $f : S \to \{0,1\}$, where $S \subseteq \{0,1\}^n$, such that $\mathrm{CRA}_1(f) \leq \sqrt{n}$ and $\mathrm{CRA}(f) = \Omega(n)$.*

Let n be a perfect square and $N := \sqrt{n}$. Now we define S. Split the n positions into N consecutive blocks of size N. An input x belongs to S iff for all N blocks, there is a single bit set to 1 and all others are 0. The function f is defined as follows:

$$f(x) = \left(\sum_{i=1}^{n} i \cdot x_i \right) \bmod 2.$$

Chapter 6

Fractional Block Sensitivity

In this chapter we describe the results concerning the relationship between block sensitivity and fractional block sensitivity. These results have been presented in the following papers:

[AKPV18] Andris Ambainis, Martins Kokainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. All Classical Adversary Methods are Equivalent for Total Functions. In *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*, volume 96 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[APV18] Andris Ambainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. On block sensitivity and fractional block sensitivity. *Lobachevskii Journal of Mathematics*, 39(7):967––969, 2018.

6.1 Introduction

In this chapter we study the relationship between block sensitivity $\mathrm{bs}(f)$ and fractional block sensitivity $\mathrm{fbs}(f)$. The latter measure was introduced as a generalization of block sensitivity independently by Tal and Gilmer, Saks and Srinivasan [Tal13, GSS16]. While block sensitivity is a lower bound on both deterministic and randomized query complexities, fractional block sensitivity is a tighter randomized lower bound (see Section 4.3).

The fractional block sensitivity can be viewed as a real-valued linear program relaxation of block sensitivity, which itself is an integer-valued program. Similarly, one can define a similar relaxation of the certificate complexity, fractional certificate complexity $\mathrm{FC}(f)$. By duality of linear programming, it follows that $\mathrm{fbs}(f) = \mathrm{FC}(f)$. It also turns out that this measure is asymptotically equivalent to the randomized certificate complexity $\mathrm{RC}(f)$ introduced earlier by Aaronson [Aar08]. Its quantum counterpart is the quantum certificate complexity $\mathrm{QC}(f)$, which abides $\mathrm{QC}(f) = \Theta(\sqrt{\mathrm{RC}(f)})$. Thus, $\mathrm{fbs}(f)$ is a fundamental randomized and quantum lower bound technique.

This measure has found a variety of applications. Using it, Aaronson has proved that $\mathrm{R}_0(f) = O(\mathrm{Q}(f)^2\,\mathrm{Q}_0(f)\log n)$ [Aar08]. Kulkarni and Tal proved that the behaviour of $\mathrm{fbs}(f)$ with the approximate degree $\widetilde{\deg}(f)$ is similar to $\mathrm{bs}(f)$, that is, $\mathrm{fbs}(f) = O(\widetilde{\deg}(f)^2)$ (which implies that $\widetilde{\deg}(f)$ is a stronger quantum lower bound than $\mathrm{QC}(f)$) [KT16]. They also proved that $\mathrm{R}_0(f) = O(\mathrm{R}(f)^2 \log \mathrm{R}(f))$, which is tight up to the logarithmic factor [ABB+17].

Finally, an open question by Aaronson asks whether $\mathrm{R}(f) = O(\mathrm{fbs}(f)^2)$ (the best upper bound now is $\mathrm{R}(f) = O(\mathrm{fbs}(f)^3)$); if true, that would imply a stronger relation between randomized and quantum query complexities, $\mathrm{R}(f) = O(\mathrm{Q}(f)^4)$ (right now the best upper bound is $\mathrm{R}(f) = O(\mathrm{Q}(f)^6)$).

The relationship between $\mathrm{bs}(f)$ and $\mathrm{fbs}(f)$ is partially understood. By definition, $\mathrm{bs}(f) \leq \mathrm{fbs}(f)$. For total functions, it is known that if f is a function with constant input size, then both measures asymptotically converge when composing f with itself any number of times [Tal13]. Tal also shows that for any measure M such that for all f and g, we have $M(f \circ g) \leq M(f) \cdot M(g)$, then an upper bound $\mathrm{bs}(f) \leq M(f)^\alpha$ also implies $\mathrm{fbs}(f) \leq M(f)^\alpha$. Thus these two measures often behave very similarly.

On the other hand, there are examples of total functions where the two are asymptotically different. The best known separation gives $\mathrm{fbs}(f) = \left(\frac{1}{3\sqrt{2}} + o(1)\right) \mathrm{bs}(f)^{3/2}$ [GSS16]. The best known upper bound is $\mathrm{fbs}(f) \leq \mathrm{C}(f) \leq \mathrm{bs}(f)^2$ [BdW02]. For partial functions, the relation between them has not been understood until now. In this work, we study this problem for partial and total functions.

6.2 Overview of The Results

In this chapter, we show the following results.

- A complete characterization of the relationship between $\mathrm{bs}(f)$ and $\mathrm{fbs}(f)$ for partial functions. We prove that $\mathrm{fbs}(f) \leq \sqrt{n \cdot \mathrm{bs}(f)}$ (Section 6.3) and give an example where this inequality is tight (Section 6.4). Hence, we completely solve the prolem for partial functions.
- A slightly improved separation between the two measures for total functions that achieves $\mathrm{fbs}(f) = \left(\frac{1}{\sqrt{6}} - o(1)\right) \mathrm{bs}(f)^{3/2}$ (Section 6.5).

6.3 Upper Bound in Terms of Block Sensitivity

In this section, we show that for partial functions, $\mathrm{fbs}(f) \leq \sqrt{n \cdot \mathrm{bs}(f)}$. We begin with a weaker statement that shows the same asymptotic result but has an easier proof.

Theorem 36. *For any partial function $f : S \to H$, where $S \subseteq G^n$, and any $x \in S$,*

$$\mathrm{fbs}(f) = O(\sqrt{n \cdot \mathrm{bs}(f)}).$$

Proof. We show that for all $x \in S$, we have $\mathrm{FC}(f, x) = O(\sqrt{n \cdot \mathrm{bs}(f, x)})$. The claim then follows as $\mathrm{fbs}(f, x) = \mathrm{FC}(f, x)$.

Since $\mathrm{FC}(f, x)$ is a minimization linear program, it suffices to show a fractional certificate v of size at most $O(\sqrt{n \cdot \mathrm{bs}(f, x)})$. Let k be a parameter between 1 and n. Let $\mathcal{B} = \{B \subseteq [n] \mid f(x) \neq f(x^B), |B| \leq k\}$ be a maximum set of non-overlapping sensitive blocks of x of size at most k. Then $|\mathcal{B}| \leq \mathrm{bs}(f)$. Let $S = \{i \in [n] \mid \exists B \in \mathcal{B} : i \in B\}$ be the set of all positions in blocks of $\mathcal{B}$. We construct the fractional certificate v by setting $v(i) = 1$ for all $i \in S$, and $v(i) = 1/k$ for all $i \notin S$.

Let B be any sensitive block of x of size at most k. As $\mathcal{B}$ is a maximum set of non-overlapping sensitive blocks, there must exist a $B' \in \mathcal{B}$ such that $B \cap B' \neq \varnothing$. Therefore,

$\sum_{i\in B} v(i) \geq |B \cap B'| \geq 1$ On the other hand, if $|B| \leq k$, then $\sum_{i\in B} v(i) \geq |B|/k \geq 1$. Hence v is a feasible fractional certificate. The size of v is $\sum_{i\in[n]} v(i) \leq |\mathcal{B}|\cdot k + n/k \leq \mathrm{bs}(f)\cdot k + n/k$. The last expression asymptotically reaches the minimum when $\mathrm{bs}(f)\cdot k = n/k$, which happens if $k = \sqrt{n/\mathrm{bs}(f)}$. In that case $\mathrm{FC}(f,x) = O(\sqrt{n\cdot \mathrm{bs}(f)})$, q.e.d. □

Now we show an optimal upper bound.

Theorem 37. *For any partial function $f : S \to H$, where $S \subseteq G^n$, and any $x \in S$,*

$$\mathrm{fbs}(f) \leq \sqrt{n \cdot \mathrm{bs}(f)}.$$

Proof. We will prove that $\mathrm{fbs}(f,x) \leq \sqrt{n\cdot \mathrm{bs}(f,x)}$ for any $x \in S$. First we introduce a parametrized version of the fractional block sensitivity. Let $x \in S$ be any input, $\mathcal{B}$ the set of sensitive blocks of x and $N \leq n$ a positive real number. Define

$$\mathrm{fbs}_N(f,x) = \max_w \sum_{B\in\mathcal{B}} w(B) \qquad \text{s.t.} \quad \forall i \in [n] : \sum_{B\in\mathcal{B}: i\in B} w(B) \leq 1,$$
$$\sum_{B\in\mathcal{B}} |B|\cdot w(B) \leq N.$$

where $w : \mathcal{B} \to [0;1]$. If we let $N = n$, then the second condition becomes redundant and $\mathrm{fbs}_n(f,x) = \mathrm{fbs}(f,x)$.

For simplicity, let $k = \mathrm{bs}(f,x)$. We will prove by induction on k that $\mathrm{fbs}_N(f,x) \leq \sqrt{Nk}$. If $k = 0$, the claim obviously holds, so assume $k > 0$. Let ℓ be the length of the shortest block in $\mathcal{B}$. Then

$$\sum_{B\in\mathcal{B}} \ell\cdot w(B) \leq \sum_{B\in\mathcal{B}} |B|\cdot w(B) \leq N$$

and $\mathrm{fbs}_N(f,x) = \sum_{B\in\mathcal{B}} w(B) \leq N/\ell$.

On the other hand, let D be any shortest sensitive block. Let f' be the restriction of f where the variables with indices in D are fixed to the values of x_i for all $i \in D$. Note that $\mathrm{bs}(f',x) \leq k-1$, as we have removed all sensitive blocks that overlap with D. Let $\mathcal{B}'$ be the set of sensitive blocks of x on f' and let $\mathcal{T} = \{B \in \mathcal{B} \mid B \cap D \neq \varnothing\}$, the set of sensitive blocks that overlap with D (including D itself). Then no $T \in \mathcal{T}$ is a member of $\mathcal{B}'$, therefore

$$\sum_{B'\in\mathcal{B}'} |B'|\cdot w(B') \leq N - \sum_{T\in\mathcal{T}} |T|\cdot w(T) \leq N - \ell\cdot\sum_{T\in\mathcal{T}} w(T).$$

Denote $t = \sum_{T\in\mathcal{T}} w(T)$. We have that $t \leq |D| = \ell$, as any $T \in \mathcal{T}$ overlaps with D. By combining the two inequalities we get

$$\begin{aligned}\mathrm{fbs}_N(f,x) &\leq \max_{\ell\in[0;n]} \min\left\{\frac{N}{\ell}, \max_{t\in[0;\ell]}\{t + \mathrm{fbs}_{N-\ell t}(f',x)\}\right\}\\ &\leq \max_{\ell\in[0;n]} \min\left\{\frac{N}{\ell}, \max_{t\in[0;\ell]}\left\{t + \sqrt{(N-\ell t)(k-1)}\right\}\right\}.\end{aligned}$$

If $N/\ell \leq \sqrt{Nk}$, we are done. Thus further assume that $\ell < \sqrt{N/k}$.

Denote $g(t) = t + \sqrt{(N-\ell t)(k-1)}$. We need to find the maximum of this function on the interval $[0;\ell]$ for a given ℓ. Its derivative,

$$g'(t) = 1 - \frac{\ell}{2}\sqrt{\frac{k-1}{N-\ell t}},$$

is a monotone function in t. Thus it has exactly one root, $t_0 = N/\ell - (k-1)\cdot\ell/4$. Therefore, $g(t)$ attains its maximum value on $[0;\ell]$ at one of the points $\{0, t_0, \ell\}$.

- If $t = 0$, then $g(0) = \sqrt{N(k-1)} \le \sqrt{Nk}$.
- If $t = t_0$, then, as $t \le \ell < \sqrt{N/k}$,

$$\sqrt{Nk} - \frac{k-1}{4}\cdot\sqrt{\frac{N}{k}} < \frac{N}{\ell} - (k-1)\frac{\ell}{4} < \sqrt{\frac{N}{k}}$$
$$\sqrt{k} - \frac{k-1}{4\sqrt{k}} < \sqrt{\frac{1}{k}}$$
$$3k < 0.$$

  The last inequality has no solutions in natural numbers for k, so this case is not possible.

- If $t = \ell$, then $g(t) = \ell + \sqrt{(N-\ell^2)(k-1)}$.

Now it remains to find the maximum value of $h(k) = \ell + \sqrt{(N-\ell^2)(k-1)}$ on the interval $[0;\sqrt{N/k}]$. The derivative is equal to

$$h'(\ell) = 1 - \ell\cdot\sqrt{\frac{k-1}{N-\ell^2}}.$$

The only non-negative root of $h'(\ell)$ is equal to $\ell_0 = \sqrt{N/k}$. Then $h(\ell)$ is monotone on the interval $[0;\sqrt{N/k}]$. Thus $h(\ell)$ attains its maximal value at one of the points $\{0, \sqrt{N/k}\}$.

- If $\ell = 0$, then $h(\ell) = \sqrt{N(k-1)} < \sqrt{Nk}$.
- If $\ell = \ell_0 = \sqrt{N/k}$, then

$$h(\ell) = \sqrt{\frac{N}{k}} + \sqrt{\left(N - \frac{N}{k}\right)(k-1)} = \sqrt{N}\left(\sqrt{\frac{1}{k}} + (k-1)\sqrt{\frac{1}{k}}\right) = \sqrt{Nk}.$$

Thus, $h(\ell) \le \sqrt{Nk}$ and that concludes the induction.

Therefore, $\mathrm{fbs}(f,x) = \mathrm{fbs}_n(f,x) \le \sqrt{n\cdot\mathrm{bs}(f,x)}$, hence also $\mathrm{fbs}(f) \le \sqrt{n\cdot\mathrm{bs}(f)}$ and we are done. □

6.4 Optimal Separation for Partial Functions

In this section we show an example of function that matches the upper bound of Theorem 37.

Theorem 38. *For any $k \in \mathbb{N}$, there exists a partial Boolean function $f : S \to \{0,1\}$, where $S \subseteq \{0,1\}^n$, such that $\mathrm{bs}(f) = k$ and $\mathrm{fbs}(f) = \Omega(\sqrt{n\cdot\mathrm{bs}(f)})$.*

Proof. Take any finite projective plane of order t, then it has $\ell = t^2 + t + 1$ many points. Let $n = k\ell$ and enumerate the points with integers from 1 to ℓ. Let $X = \{0^\ell\}$ and $Y = \{y \mid$ there exists a line L such that $y_i = 1$ iff $i \in L\}$. Define the (partial) finite projective plane function $\textsc{Fpp}_t : X \cup Y \to \{0,1\}$ as $\textsc{Fpp}_t(y) = 1 \iff y \in Y$.

We can calculate the 1-sided block sensitivity measures for this function:

- $\mathrm{fbs}^0(\textsc{Fpp}_t) \geq (t^2+t+1)\cdot\frac{1}{t+1} = \Omega(t)$, as each line gives a sensitive block for 0^n; since each point belongs to $t+1$ lines, we can assign weight $1/(t+1)$ for each sensitive block and that is a feasible solution for the fractional block sensitivity linear program.
- $\mathrm{bs}^0(\textsc{Fpp}_t) = 1$, as any two lines intersect, so any two sensitive blocks of 0^n overlap.
- $\mathrm{bs}^1(\textsc{Fpp}_t) = 1$, as there is only one negative input.

Next, define $f : S^{\times k} \to \{0,1\}$ as the composition of Or with the finite projective plane function, $f = \textsc{Or}_k(\textsc{Fpp}_t(x^{(1)}), \ldots, \textsc{Fpp}_t(x^{(k)}))$.

To obtain the final result, we use the following lemma:

Lemma 39 (Proposition 31 in [GSS16])**.** *Let g be a non-constant Boolean function and*

$$f = \textsc{Or}(g^{(1)}, \ldots, g^{(m)}),$$

an Or *composed with m copies of g. Then for complexity measures $M \in \{\mathrm{bs}, \mathrm{fbs}\}$, we have*

$$M_1(f) = M_1(g)$$
$$M_0(f) = m \cdot M_0(g).$$

Then, by Lemma 39, we have

- $\mathrm{fbs}(f) = \max\{\mathrm{fbs}^0(f), \mathrm{fbs}^1(f)\} \geq \mathrm{fbs}^0(f) = \mathrm{fbs}^0(\textsc{Fpp}_t)\cdot k = \Theta(t)\cdot k = \Theta(t\cdot n/t^2) = \Theta(n/t)$,
- $\mathrm{bs}(f) = \max\{\mathrm{bs}^0(f), \mathrm{bs}^1(f)\} = \mathrm{bs}^0(\textsc{Fpp}_t)\cdot k = k = \Theta(n/t^2)$.

As $\sqrt{n\cdot n/t^2} = n/t$, we have $\mathrm{fbs}(f) = \Omega(\sqrt{n\cdot \mathrm{bs}(f)})$ and hence the result. □

Note that our example is also tight in regards to the multiplicative constant, since t can be unboundedly large (and the constant arbitrarily close to 1).

6.5 Improved Separation for Total Functions

In this section, we give a slightly improved separation between $\mathrm{fbs}(f)$ and $\mathrm{bs}(f)$ for total functions. The separation in [GSS16] composes a graph property Boolean function (namely, whether a given graph is a star graph) with the Or function. We build on these ideas and define a new graph property g for the composition that gives a larger separation.

Theorem 40. *There exists a family of Boolean functions such that*

$$\mathrm{fbs}(f) = \left(\frac{1}{\sqrt{6}} - o(1)\right)\mathrm{bs}(f)^{3/2}.$$

Proof. Let $N \geq 12$ be a multiple of 3. An input on $\binom{N}{2}$ variables $(x_{1,2}, x_{1,3}, \ldots, x_{N-1,N})$ encodes a graph G on N vertices. Let $x_{i,j} = 1$ iff the vertices i and j are connected by an edge in G.

We define an auxiliary function $g : \{0,1\}^{\binom{N}{2}} \to \{0,1\}$. Partition $[N]$ into three sets S_0, S_1, S_2 such that $S_r = \{i \in [N] \mid i \equiv r \pmod 3\}$. Let $g(x) = 1$ iff:

1. there is some vertex i that is connected to every other vertex by an edge (a star graph);
2. for any $r \in \{0,1,2\}$, no two vertices $j,k \neq i$ such that $j,k \in S_r$ are connected by an edge.

Formally, $g(x) = 1$ iff x satisfies one of the following 1-certificates $C_1, \ldots, C_N$: C_i assigns 1 to every edge in $\{x_{j,k} \mid j = i \vee k = i\}$, and assigns 0 to every edge in $\{x_{jk} \mid j \neq i, k \neq i, j \equiv k \pmod 3\}$.

Now we calculate the values of $\mathrm{bs}_0(g), \mathrm{bs}_1(g), \mathrm{fbs}_0(g)$.

1. $\mathrm{bs}_0(g) = 3$.

   Consider an input x describing a triangle graph between vertices i,j,k. For this input $g(x) = 0$. Let x' be an input obtained from x by removing the edge $x_{i,j}$ and adding all the missing edges $x_{k,l}$, for all $l \neq i,j$. The corresponding graph is a star graph, therefore, $g(x') = 1$. Let B_k be the sensitive block that flips x to x'. Similarly define B_i and B_j. None of the three blocks overlap, hence $\mathrm{bs}_0(g,x) \geq 3$.

   Now we prove that $\mathrm{bs}_0(g) \leq 3$. Assume the contrary, that there exists an input $x \in f^{-1}(0)$ with $\mathrm{bs}(g,x) \geq 4$. Then x has (at least) 4 non-overlapping sensitive blocks $B_1, \ldots, B_4$. Each x^{B_i} satisfies one of the 1-certificates, each a different one. There are 4 such certificates, therefore at least two of them require a star at vertices i,j belonging to the same S_r. The corresponding certificates C_i and C_j both assign 1 at the edge $x_{i,j}$. On the other hand, every other C_k assigns 0 at $x_{i,j}$. Therefore, of the 4 certificates corresponding to $B_1, \ldots, B_4$, two assign 1 to this edge and two assign 0 to this edge. Then, regardless of the value of $x_{i,j}$, we would need to flip it in two of the blocks $B_1, \ldots, B_4$: a contradiction, since the blocks don't overlap. Therefore no such x exists.

2. $\mathrm{bs}_1(g) = \frac{N^2}{6} + \frac{N}{6}$.

   Examine any 1-certificate C_i. Find three indices $j,k,l \equiv i \pmod 3$ (this is possible, as $N \geq 12$). Any input x that satisfies C_i has $x_{i,j} = x_{i,k} = x_{i,l} = 1$. On the other hand, any other 1-certificate C_t requires at least two of the variables $x_{i,j}, x_{i,k}, x_{i,l}$ to be 0. Hence, the Hamming distance between C_i and C_t is at least two. Therefore, flipping any position of x that is fixed in C_i changes the value of the function as well. Thus, we have

   $$\mathrm{bs}(f,x) = \mathrm{C}(f,x) = |C_i| = 3\binom{N/3}{2} + \frac{2N}{3} = \frac{N^2}{6} + \frac{N}{6}.$$

3. $\mathrm{fbs}_0(g) \geq \frac{N}{2}$.

   Examine the all zeros input $0^{\binom{N}{2}}$. Any sensitive block B of this input flips the edges on a star from some vertex. Therefore, any position is flipped by exactly two of the sensitive blocks. The weights $w_B = \frac{1}{2}$ for each sensitive block B then give a feasible solution for the fractional block sensitivity linear program. As there are N sensitive blocks, $\mathrm{fbs}(g, 0^{\binom{N}{2}}) = \frac{N}{2}$.

To obtain the final function we use Lemma 39. Let $m = \mathrm{bs}_1(g)/\mathrm{bs}_0(g) = \frac{N^2}{18} + \frac{N}{18}$. Then $\mathrm{bs}(f) = \mathrm{bs}_0(f) = \mathrm{bs}_1(f) = \mathrm{bs}_1(g) = \frac{N^2}{6} + \frac{N}{6}$. On the other hand, $\mathrm{fbs}(f) \geq \mathrm{fbs}_0(f) = m \cdot \mathrm{fbs}_0(g) \geq m \cdot \frac{N}{2} = \frac{N^3}{36} + \frac{N^2}{36}$. Therefore, we have $\mathrm{fbs}(f) \geq \left(\frac{N^2}{6} + \frac{N}{6}\right) \cdot \frac{N}{6} = \mathrm{bs}(f) \cdot \left(\frac{1}{\sqrt{6}} - o(1)\right)\sqrt{\mathrm{bs}(f)} = \left(\frac{1}{\sqrt{6}} - o(1)\right)\mathrm{bs}(f)^{3/2}$. □

Conclusion

In this thesis, we have explored a number of ways of limitations of quantum and randomized algorithms.

We have introduced oscillatory localization of Grover's quantum walk and developed mathematical tools to estimate the amount of localization exhibited by the quantum walk depending on the starting state. We have shown a variety of examples of such localization, which is not present in the classical random walks.

Next, we have investigated the stationary states of the quantum search. If the starting state of the search is close to a stationary state, the quantum walk does not give any quantum advantage over the classical algorithms. We have given a complete characterization of such states and have shown condition on the existence of such states in a graph. As a result, we have described a family of configurations of multiple marked vertices where Grover's quantum search is not efficient and thus other quantum algorithms are needed to solve the problem.

Both localization and stationary properties of the quantum walk involve analyzing eigenvectors with eigenvalue 1 of the corresponding walk operators. Since many of quantum walk-based algorithm employ phase estimation of eigenvectors of some unitary operator, a possible application of our results is to develop new algorithms to test some graph properties. This is an open question as of the moment of this research.

We then have studied the relationship between classical adversary lower bound methods. It has been known that the quantum adversary bounds are all asymptotically equivalent. In this work, we have proven the equivalence between the randomized adversary lower bounds for total functions, and also have shown that they are equal to the fractional block sensitivity. On the other hand, we have shown separations between them for partial functions, unlike in the quantum regime. We also have introduced a new classical adversary method, rank-1 relational bound. Thus we have simplified the overall picture of lower bounds in the randomized query model.

Finally, we have studied the relationship between block sensitivity and fractional block sensitivity. We have proved an optimal separation between them for partial functions. For total functions, we have shown a slight improvement over the best known separation example. In general, the relationship between the two for total functions remains an open question. In fact, proving that the current examples are asymptotically optimal would imply new lower bounds for block sensitivity of transitive Boolean functions. This is another open direction of this research.

Bibliography

[AA18] Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. *SIAM J. Comput.*, 47(3):982–1038, 2018.

[Aar06] Scott Aaronson. Lower bounds for local search by quantum arguments. *SIAM Journal on Computing*, 35(4):804–824, 2006.

[Aar08] Scott Aaronson. Quantum certificate complexity. *Journal of Computer and System Sciences*, 74(3):313–322, 2008.

[ABB+17] Andris Ambainis, Kaspars Balodis, Aleksandrs Belovs, Troy Lee, Miklos Santha, and Juris Smotrovs. Separations in query complexity based on pointer functions. *J. ACM*, 64(5):32:1–32:24, 2017.

[ABDK16] Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC'16, pages 863–876, New York, NY, USA, 2016. ACM.

[ABI+19] Andris Ambainis, Kaspars Balodis, Jānis Iraids, Martins Kokainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. Quantum speedups for exponential-time dynamic programming algorithms. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1783–1793, 2019.

[ABN+13] Andris Ambainis, Artūrs Bačkurs, Nikolajs Nahimovs, Raitis Ozols, and Alexander Rivosh. Search by quantum walks on two-dimensional grid without amplitude amplification. In *Theory of Quantum Computation, Communication, and Cryptography*, pages 87–97, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[ACR+07] Andris Ambainis, Andrew M. Childs, Ben Reichardt, Robert Špalek, and Shengyu Zhang. Any AND-OR formula of size n can be evaluated in time $n^{1/2+o(1)}$ on a quantum computer. *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 363–372, 2007.

[ADZ93] Yakir Aharonov, Luis Davidovich, and Nicim Zagury. Quantum random walks. *Phys. Rev. A*, 48:1687–1690, 1993.

[AF02] David Aldous and James Allen Fill. Reversible markov chains and random walks on graphs, 2002.

[AGM13] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Spectral sparsification in dynamic graph streams. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques. Proceedings of the 16th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, and the 17th International Workshop on Randomization and Computation*, APPROX/RANDOM 2013, pages 1–10, Berlin, Heidelberg, 2013. Springer.

[AK17] Andris Ambainis and Martins Kokainis. Quantum algorithm for tree size estimation, with applications to backtracking and 2-player games. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 989–1002, New York, NY, USA, 2017. ACM.

[AKPV18] Andris Ambainis, Martins Kokainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. All Classical Adversary Methods are Equivalent for Total Functions. In *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*, volume 96 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[AKR05] Andris Ambainis, Julia Kempe, and Alexander Rivosh. Coins make quantum walks faster. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 1099–1108, Philadelphia, PA, USA, 2005. SIAM.

[AL97] Emile Aarts and Jan K. Lenstra, editors. *Local Search in Combinatorial Optimization*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1997.

[All96] Eric Allender. Circuit complexity before the dawn of the new millennium. In V. Chandru and V. Vinay, editors, *Foundations of Software Technology and Theoretical Computer Science*, pages 1–18, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[Amb99] Andris Ambainis. A better lower bound for quantum algorithms searching an ordered list. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS '99, pages 352–, Washington, DC, USA, 1999. IEEE Computer Society.

[Amb00] Andris Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, STOC'00, pages 636–643, New York, NY, USA, 2000. ACM.

[Amb03a] Andris Ambainis. Polynomial degree vs. quantum query complexity. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, FOCS'03, pages 230–239, Washington, DC, USA, 2003. IEEE Computer Society.

[Amb03b] Andris Ambainis. Quantum walks and their algorithmic applications. *Int. J. Quantum Inf.*, 01(04):507–518, 2003.

[Amb04] Andris Ambainis. Quantum walk algorithm for element distinctness. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '04, pages 22–31. IEEE Computer Society, 2004.

[APV16] Andris Ambainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. Sensitivity versus certificate complexity of Boolean functions. In *Computer Science – Theory and Applications*, pages 16–28, Cham, 2016. Springer International Publishing.

[APV18] Andris Ambainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. On block sensitivity and fractional block sensitivity. *Lobachevskii Journal of Mathematics*, 39(7):967––969, 2018.

[APVW16] Andris Ambainis, Krišjānis Prūsis, Jevgēnijs Vihrovs, and Thomas G. Wong. Oscillatory localization of quantum walks analyzed by classical electric circuits. *Phys. Rev. A*, 94:062324, 2016.

[AS04] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, 2004.

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001.

[BCJ+13] Aleksandrs Belovs, Andrew M. Childs, Stacey Jeffery, Robin Kothari, and Frédéric Magniez. Time-efficient quantum walks for 3-distinctness. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming*, ICALP '13, pages 105–122, Berlin, Heidelberg, 2013. Springer.

[BCN89] Andries E. Brouwer, Arjeh M. Cohen, and Arnold Neumaier. *Distance-Regular Graphs*. Number 18 in Ergebnisse der Mathematik und ihrer Grenzgebiete. Springer-Verlag, 1989.

[BdW01] Harry Buhrman and Ronald de Wolf. Communication complexity lower bounds by polynomials. In *Proceedings of the 16th Annual Conference on Computational Complexity*, CCC '01, pages 120–, Washington, DC, USA, 2001. IEEE Computer Society.

[BdW02] Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21 – 43, 2002.

[BHMT00] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *arXiv:quant-ph/0005055*, 2000.

[Bol79] Béla Bollobás. *Graph theory: an introductory course*. Springer-Verlag, New York, NY, USA, 1979.

[BR12] Aleksandrs Belovs and Ben W. Reichardt. Span programs and quantum algorithms for st-connectivity and claw detection. In *Proceedings of the 20th Annual European Conference on Algorithms*, ESA'12, pages 193–204, Berlin, Heidelberg, 2012. Springer-Verlag.

[BVW18] Balthazar Bauer, Jevgēnijs Vihrovs, and Hoeteck Wee. On the inner product predicate and a generalization of matching vector families. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2018)*, volume 122 of *Leibniz International Proceedings*

in Informatics (LIPIcs), pages 41:1–41:13, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[CFG02] Andrew M. Childs, Edward Farhi, and Sam Gutmann. An example of the difference between quantum and classical random walks. *Quantum Information Processing*, 1(1):35–43, 2002.

[CG04] Andrew M. Childs and Jeffrey Goldstone. Spatial search by quantum walk. *Phys. Rev. A*, 70:022314, 2004.

[Chi10] Andrew M. Childs. On the relationship between continuous- and discrete-time quantum walk. *Commun. Math. Phys.*, 294(2):581–603, 2010.

[CRR+96] Ashok K. Chandra, Prabhakar Raghavan, Walter L. Ruzzo, Roman Smolensky, and Prasoon Tiwari. The electrical resistance of a graph captures its commute and cover times. *Comput. Complexity*, 6(4):312–340, 1996.

[DS84] Peter G. Doyle and James L. Snell. *Random walks and electric networks*. Number 22 in Carus Mathematical Monographs. Mathematical Association of America, 1984.

[FG98] Edward Farhi and Sam Gutmann. Quantum computation and decision trees. *Phys. Rev. A*, 58:915–928, 1998.

[Fos49] Ronald M. Foster. *The average impedance of an electrical network*, pages 333–340. Edwards, 1949.

[Fre77] Rusins Freivalds. Probabilistic machines can use less running time. In *IFIP Congress*, pages 839–842, 1977.

[GGLR98] Oded Goldreich, Shafi Goldwasser, Eric Lehman, and Dana Ron. Testing monotonicity. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, FOCS '98, pages 426–, Washington, DC, USA, 1998. IEEE Computer Society.

[Gia08] Douglas C. Giancoli. *Physics for Scientists & Engineers with Modern Physics*. Pearson, 4 edition, 2008.

[GLM08] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, 2008.

[GPW17] M. Göös, T. Pitassi, and T. Watson. Query-to-communication lifting for BPP. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 132–143, 2017.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. ACM.

[GSS16] Justin Gilmer, Michael Saks, and Srikanth Srinivasan. Composition limits and separating examples for some Boolean function complexity measures. *Combinatorica*, 36(3):265–311, 2016.

[HBF03] Mark Hillery, Janos Bergou, and Edgar Feldman. Quantum walks based on an interferometric analogy. *Phys. Rev. A*, 68:032314, 2003.

[HLŠ07] Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC'07, pages 526–535, New York, NY, USA, 2007. ACM.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc. (N.S.)*, 43(4):439–561, 2006.

[HNS02] Peter Høyer, Jan Neerbek, and Yaoyun Shi. Quantum complexities of ordered searching, sorting, and element distinctness. *Algorithmica*, 34(4):429–448, 2002.

[Hoa61] C. A. R. Hoare. Algorithm 64: Quicksort. *Commun. ACM*, 4(7):321–, 1961.

[IKK04] Norio Inui, Yoshinao Konishi, and Norio Konno. Localization of two-dimensional quantum walks. *Phys. Rev. A*, 69:052323, 2004.

[JK10] Rahul Jain and Hartmut Klauck. The partition bound for classical communication complexity and query complexity. In *Proceedings of the 2010 IEEE 25th Annual Conference on Computational Complexity*, CCC'10, pages 247–258, Washington, DC, USA, 2010. IEEE Computer Society.

[JKK+18] Rahul Jain, Hartmut Klauck, Srijita Kundu, Troy Lee, Miklos Santha, Swagato Sanyal, and Jevgēnijs Vihrovs. Quadratically tight relations for randomized query complexity. In *Computer Science – Theory and Applications*, pages 207–219, Cham, 2018. Springer International Publishing.

[KKJ15] Bálint Kollár, Tamás Kiss, and Igor Jex. Strongly trapped two-dimensional quantum walks. *Phys. Rev. A*, 91:022308, 2015.

[KR93] Douglas J. Klein and Milan Randić. Resistance distance. *J. Math. Chem.*, 12(1):81–95, 1993.

[KRBD10] Takuya Kitagawa, Mark S. Rudner, Erez Berg, and Eugene Demler. Exploring topological phases with quantum walks. *Phys. Rev. A*, 82:033429, 2010.

[KT16] Raghav Kulkarni and Avishay Tal. On fractional block sensitivity. *Chicago Journal Of Theoretical Computer Science*, 8:1–16, 2016.

[LG14] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 296–303, New York, NY, USA, 2014. ACM.

[LL93] Arjen K. Lenstra and Hendrik W.Jr. Lenstra. *The Development of the Number Field Sieve*. Number 1554 in Lecture Notes in Mathematics. Springer Science & Business Media, 1993.

[LM04] Sophie Laplante and Frédéric Magniez. Lower bounds for randomized and quantum query complexity using Kolmogorov arguments. In *Proceedings of the 19th IEEE Annual Conference on Computational Complexity*, CCC'04, pages 294–304, Washington, DC, USA, 2004. IEEE Computer Society.

[Lov93] László Lovász. Random walks on graphs: A survey, 1993.

[LV08] Ming Li and Paul M. B. Vitnyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Publishing Company, Incorporated, 3 edition, 2008.

[MBSS02] Troy D. Mackay, Stephen D. Bartlett, Leigh T. Stephenson, and Barry C. Sanders. Quantum walks in higher dimensions. *J. Phys. A: Math. Gen.*, 35(12):2745, 2002.

[Mey96] David A. Meyer. From quantum cellular automata to quantum lattice gases. *J. Stat. Phys.*, 85(5-6):551–574, 1996.

[MN07] Frederic Magniez and Ashwin Nayak. Quantum complexity of testing group commutativity. *Algorithmica*, 48(3):221–232, 2007.

[MNSX11] Frédéric Magniez, Ashwin Nayak, Miklos Santha, and David Xiao. Improved bounds for the randomized decision tree complexity of recursive majority. In *Automata, Languages and Programming*, pages 317–329, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[Mon15] Ashley Montanaro. Quantum walk speedup of backtracking algorithms, 2015.

[Mon16] Ashley Montanaro. Quantum algorithms: an overview. *Npj Quantum Information*, 2:15023, 2016.

[MPAD08] Franklin L. Marquezino, Renato Portugal, Gonzalo Abal, and Raul Donangelo. Mixing times in quantum walks on the hypercube. *Phys. Rev. A*, 77:042312, 2008.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1995.

[MSS05] Frédéric Magniez, Miklos Santha, and Mario Szegedy. Quantum algorithms for the triangle problem. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 1109–1117, Philadelphia, PA, USA, 2005. SIAM.

[Nis89] Noam Nisan. CREW PRAMs and decision trees. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, STOC'89, pages 327–335, New York, NY, USA, 1989. ACM.

[NR16] Nikolajs Nahimovs and Alexander Rivosh. Exceptional configurations of quantum walks with Grover's coin. In *Mathematical and Engineering Methods in Computer Science*, pages 79–92, Cham, 2016. Springer International Publishing.

[NS17] Nikolajs Nahimovs and Raqueline A. M. Santos. Adjacent vertices can be hard to find by quantum walks. In *SOFSEM 2017: Theory and Practice of Computer Science*, pages 256–267, Cham, 2017. Springer International Publishing.

[PVW16a] Krišjānis Prūsis, Jevgēnijs Vihrovs, and Thomas G. Wong. Doubling the success of quantum walk search using internal-state measurements. *J. Phys. A: Math. Theor.*, 49(45):455301, 2016.

[PVW16b] Krišjānis Prūsis, Jevgēnijs Vihrovs, and Thomas G. Wong. Stationary states in quantum walk search. *Phys. Rev. A*, 94:032334, 2016.

[Rei09] Ben W. Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every Boolean function. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS'09, pages 544–551, Washington, DC, USA, 2009. IEEE Computer Society.

[San08] Miklos Santha. Quantum walk based search algorithms. In Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li, editors, *Theory and Applications of Models of Computation*, pages 31–46, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[Sim83] Hans-Ulrich Simon. A tight $\Omega(\log\log n)$-bound on the time for parallel ram's to compute nondegenerated Boolean functions. In *Proceedings of the 1983 International FCT-Conference on Fundamentals of Computation Theory*, pages 439–444, London, UK, 1983. Springer-Verlag.

[Sim97] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997.

[SKW03] Neil Shenvi, Julia Kempe, and K. Birgitta Whaley. Quantum random-walk search algorithm. *Phys. Rev. A*, 67:052307, 2003.

[ŠS06] Robert Špalek and Mario Szegedy. All quantum adversary methods are equivalent. *Theory of Computing*, 2(1):1–18, 2006.

[Sze04] Mario Szegedy. Quantum speed-up of markov chain based algorithms. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '04, pages 32–41, Washington, DC, USA, 2004. IEEE Computer Society.

[Tal13] Avishay Tal. Properties and applications of Boolean function composition. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS'13, pages 441–454, New York, NY, USA, 2013. ACM.

[Tar89] Gabor Tardos. Query complexity, or why is it difficult to separate $\mathbf{NP}^A \cap \mathbf{coNP}^A$ from $\mathbf{P}^A$ by random oracles? *Combinatorica*, 9(4):385–392, 1989.

[Tet91] Prasad Tetali. Random walks and the effective resistance of networks. *J. Theor. Probab.*, 4(1):101–109, 1991.

[TFMK03] Ben Tregenna, Will Flanagan, Rik Maile, and Viv Kendon. Controlling discrete quantum walks: coins and initial states. *New J. Phys.*, 5(1):83, 2003.

[VA12] Salvador E. Venegas-Andraca. Quantum walks: a comprehensive review. *Quantum Information Processing*, 11(5):1015–1106, 2012.

[VFQF17] Ihor Vakulchyk, Mikhail V. Fistul, Pinquan Qin, and Sergej Flach. Anderson localization in generalized discrete-time quantum walks. *Phys. Rev. B*, 96:144204, 2017.

[WA15] Thomas G. Wong and Andris Ambainis. Quantum search with multiple walk steps per oracle query. *Phys. Rev. A*, 92:022338, 2015.

[Wat18] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018.

[Weg87] Ingo Wegener. *The Complexity of Boolean Functions*. John Wiley & Sons, Inc., New York, NY, USA, 1987.

[Won15] Thomas G. Wong. Grover search with lackadaisical quantum walks. *J. Phys. A: Math. Theor.*, 48(43):435304, 2015.

[Zaj18] Aleksejs Zajakins. Apakšējie novērtējumi varbūtiskajiem vaicājumalgoritmiem. *Course thesis, Faculty of Computing, University of Latvia*, 2018.

[Zha09] Shengyu Zhang. Tight bounds for randomized and quantum local search. *SIAM Journal on Computing*, 39(3):948–977, 2009.