

University of Minho
School of Engineering

Bruno Filipe Jardim Machado

Analyzing quantum learning protocols with ZX



University of Minho
School of Engineering

Bruno Filipe Jardim Machado

Analyzing quantum learning protocols with ZX

Master's Dissertation in Informatics Engineering

Formal Methods
Dissertation supervised by
Luís Soares Barbosa

Copyright and Terms of Use for Third Party Work

This dissertation reports on academic work that can be used by third parties as long as the internationally accepted standards and good practices are respected concerning copyright and related rights.

This work can thereafter be used under the terms established in the license below.

Readers needing authorization conditions not provided for in the indicated licensing should contact the author through the RepositóriUM of the University of Minho.

License granted to users of this work:



CC BY

<https://creativecommons.org/licenses/by/4.0/>

Acknowledgements

This dissertation is the product of an unexpected amalgamation of interests, Quantum Computing and Formal Methods. The seed for these interests was sown as an undergraduate of Computer Science by two individuals, those being Luís Barbosa and José Nuno Oliveira.

For that I would like to express my utmost gratitude to my supervisor Luís Barbosa, for the patience, advice, the countless opportunities and for always believing in me and my work, sometimes even more than I did. I would also like to thank professor José Nuno Oliveira for all the conversations, advice, passion for teaching and for guiding me into this direction. This dissertation benefitted immensely from the expertise of several people those being André Sequeira, Jaime Santos and of course professor Luís Paulo Santos.

My appreciation expands, unquestionably, to my friends, namely and in no particular order, David, José, Pedro, Miguel, Simão and Tiago. They accompanied me throughout my academic journey and all the challenges it comes with. Their impact on my life over the last few years is unquestionable and surely I wouldn't be in the place I am without them. I'm extremely proud of everything we've ever achieved together, especially what we did with NECC. Which was at times a second home for me and everyone named.

Finally, I would like to thank my family, specially my parents for their unconditional support and encouragement. For raising me with the assurance that you can never know too much and that hard work never fails. Truly, all they have done for me cannot be expressed through words. And for that, I would like to dedicate this dissertation to my parents

Maria and Joaquim

This work was partially supported by FCT, the Portuguese Foundation for Science and Technology, within the project IBEX, with reference PTDC/CCI-COM/4280/2021.

Statement of Integrity

I hereby declare having conducted this academic work with integrity.

I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, Braga, october 2024

Bruno Filipe Jardim Machado

Abstract

Quantum computing is a rapidly developing field of study, with new and useful applications coming by the day. However, current tools and techniques for the development of quantum software are quite lackluster, as computer scientists are forced to reason with quantum gates.

This dissertation utilizes an emerging graphical language, the *ZX-calculus*. This calculus is formed of two generators, the *spiders*, each one abstracting a linear map over the Z- or X-axis. Accompanying these generators, come a set of rewrite rules that allow for the transformation of ZX-diagrams. The ZX-calculus allows for a reasoning about quantum computation that transcends the circuit model most computer scientists are accustomed to.

The ZX-calculus is used in this dissertation in order to model, analyze and establish properties about a variety of different quantum protocols. These protocols span a plentitude of areas such as: quantum walks, quantum unconstrained binary optimization and quantum machine leaning.

The analysis and circuit optimization conducted in quantum walks proved to be quite rewarding. As the number of gates was severely decreased for the discrete time models, the coined quantum walk and the staggered quantum walk. The staggered variant exhibited an unexpected characteristic, an alternative evolution operator, that is capable of approximating the original operator with a fraction of the gates.

Contrasting the optimization and interesting properties exhibited by the quantum walks, the Quantum Approximate Optimization Algorithm didn't possess any notable qualities. It did however, prove to be a bridge between the classical paradigm of the circuit model and the ever-growing measurement based paradigm.

The final set of algorithms analyzed were a set of hardware efficient ansatzs. Which, in conjunction with novel techniques within the ZX-calculus framework allowed for a completely graphical analytical barren plateau detection. It was concluded that neither of the ansatzs analyzed possessed barren plateaus.

Keywords ZX-calculus, Quantum Computing, PyZX, Quantum Walks, QAOA, QML, Barren Plateau, Barren Plateau Analysis

Resumo

A computação quântica é um campo de estudo em rápido desenvolvimento, com novas e úteis aplicações a surgir diariamente. No entanto, as ferramentas e técnicas atuais para o desenvolvimento de software quântico são bastante insatisfatórias, dado que os cientistas da computação são obrigados a raciocinar com *gates* quânticas.

Esta dissertação utiliza uma linguagem gráfica emergente, o *ZX-calculus*. Este cálculo é formado por dois geradores, as *spiders*, cada uma abstraído um mapa linear sobre o eixo Z ou X. Acompanhando estes geradores, existe um conjunto de regras de reescrita que permitem a transformação de *ZX-diagrams*. O *ZX-calculus* possibilita um raciocínio sobre a computação quântica que transcende o modelo de circuitos ao qual a maioria dos cientistas da computação está habituada.

O *ZX-calculus* é utilizado nesta dissertação para modelar, analisar e estabelecer propriedades sobre uma variedade de diferentes protocolos quânticos. Estes protocolos abrangem várias áreas, tais como: caminhadas quânticas, otimização binária não-constrangida quântica e *machine learning* quântico.

A análise e otimização de circuitos executada nas caminhadas quânticas revelou-se bastante gratificante. O número de *gates* foi significativamente reduzido para os modelos de tempo discreto, a caminhada quântica com moeda e a caminhada quântica escalonada. A variante escalonada apresentou uma característica inesperada, um operador de evolução alternativo, capaz de aproximar o operador original com uma fracção das *gates*.

Contrastando com a otimização e as propriedades interessantes exibidas pelas caminhadas quânticas, o Algoritmo de Otimização Quântica Aproximada não demonstrou características notáveis. Contudo, provou ser uma ponte entre o paradigma clássico do modelo de circuitos e o paradigma, em constante crescimento, baseado em medições.

O último conjunto de algoritmos analisado foi um grupo de *ansatzs* eficientes para hardware. Em conjunto com técnicas novas dentro do quadro do *ZX-calculus*, permitiram uma detecção gráfica completamente analítica de *barren plateaus*. Concluiu-se que nenhum dos *ansatzs* analisados possuía *barren plateaus*.

Palavras-chave ZX-calculus, Computação Quântica, PyZX, Caminhadas Quânticas, QAOA, QML, Barren Plateau, Análise de Barren Plateau

Contents

- 1 Introduction 1**
 - 1.1 Context and Motivation 1
 - 1.2 Objectives 1
 - 1.3 State of the art 2
 - 1.3.1 The ZX-Calculus 2
 - 1.3.2 Quantum Computing and Quantum Machine Learning 3
 - 1.3.3 The barren plateau phenomenon 4

- 2 Background 6**
 - 2.1 Introduction to Quantum Computing 6
 - 2.1.1 States 6
 - 2.1.2 Evolution 7
 - 2.1.3 Composition 7
 - 2.1.4 Measurements 8
 - 2.2 Introduction to ZX 9
 - 2.2.1 Introduction 9
 - 2.2.2 Rewrite Rules 10
 - 2.2.3 Circuit Rewriting 13
 - 2.2.4 T-count reduction 16
 - 2.2.5 PyZX 16

- 3 Modelling algorithms in ZX 17**
 - 3.1 Quantum Walks 17
 - 3.1.1 Introduction 17
 - 3.1.2 Continuous-Time Quantum Walks 18
 - 3.1.3 Coined Quantum Walk 19

3.1.4	Staggered Quantum Walk	19
3.2	Quantum Approximate Optimization Algorithm	21
3.2.1	Introduction	21
3.2.2	Constructing the QAOA Hamiltonian	22
3.2.3	The Ansatz	24
3.3	Hardware Efficient Ansatz	27
3.3.1	Introduction	27
3.3.2	Analytical Barren Plateau Detection	28
3.3.3	Example Ansatz	30
3.3.4	Sim Inspired Ansatz	30
4	Optimization and Analysis	32
4.1	Continuous Time Quantum Walk	32
4.2	Coined Quantum Walk	35
4.3	Staggered Quantum Walk	37
4.3.1	Optimization	37
4.3.2	Alternative Evolution Operator	38
4.4	Quantum Approximate Optimization Algorithm	41
4.5	Hardware Efficient Ansatz	43
4.5.1	Example Ansatz	43
4.5.2	Sim Inspired Ansatz	47
5	Conclusions and Future Work	50
5.0.1	Discussion and Contributions	50
5.0.2	Future Work	51
5.0.3	Measurement Based Quantum Computing	52
I	Appendices	62
A	Models	63

List of Figures

- 1 The repeating pattern of the **QAOA** ansatz for the graph 3.29 26
- 2 Example of a simple **HEA**. The dashed box denotes the repeatable pattern of the circuit.
This is also the ansatz denoted as 'Circuit 1' in the work of Sim et al. (2019). 27
- 3 The alternative evolution operator. 39
- 4 The variation of the alternative evolution operator. 39
- 5 The generalization of the alternative evolution operator. 39
- 6 The state space probabilities for a 4-qubit quantum walk with 4 steps using the classical
operator (red) and the alternative one (blue). 41
- 7 The two equivalent solutions to the MaxCut problem. 42
- 8 The probability distribution of the solution for the MaxCut problem. 42

List of Tables

- 1 Number of total gates in the circuit in relation to the simplification routines used 38
- 2 Number of T gates in the circuit in relation to the simplification routines used 38

Acronyms

HEA Hardware Efficient Ansatz.

MBQC Measurement Based Quantum Computing.

NISQ Noisy Intermediate-Scale Quantum.

PQC Parameterized Quantum Circuit.

QAOA Quantum Approximate Optimization Algorithm.

QUBO Quadratic Unconstrained Binary Optimization.

VQA Variational Quantum Algorithm.

Chapter 1

Introduction

1.1 Context and Motivation

In the quantum, as in the classical domain, software is a critical factor in the reliability of computer systems. Actually, if quantum computing is quickly coming of age, with potential groundbreaking impacts on many different fields, such benefits come at a price: quantum programming is hard and finding new quantum algorithms is far from straightforward. Current methods and tools for quantum software development are still highly fragmentary and fundamentally ‘low-level’. Reasoning directly with quantum gates is as limited as assembling logical gates in classical algorithm design. It sweeps under the carpet all key ingredients of a mature software engineering discipline: compositionality, abstraction, refinement, high-order and property-enforcing type schemes

Thus, the need for suitable formal techniques in quantum software development is even bigger than in classical computation. A lack of reliable approaches to quantum computer programming may put at risk the expected quantum advantage of the new hardware.

This dissertation will explore a graphical language, with a precise mathematical semantics, for reasoning about diagrams representing quantum processes (actually, able to model arbitrary linear maps between qubits) – the ZX Calculus. ZX-diagrams come equipped with a set of useful rewrite rules with which one may reason about quantum theory, as well as optimize and validate quantum programs and protocols.

1.2 Objectives

The main purpose for this dissertation is to use ZX-calculus to model quantum algorithms, and seek for optimized versions of them. The original, motivation example was the barren plateau problem in the context of quantum machine learning algorithms. The problem was first presented by [McClean et al. \(2018\)](#). It

appears throughout the quantum machine learning field.

Another class of circuits addressed in this dissertation are the ones that implement quantum walks, the quantum counterpart of random walks, first introduced by [Aharonov et al. \(1993\)](#) with the *coined quantum walk*. Later expanded in the work of [Childs and Goldstone \(2004\)](#) with the *continuous time quantum walk*. [Portugal \(2016\)](#) introduced an innovative model, the *staggerd quantum walk*.

1.3 State of the art

1.3.1 The ZX-Calculus

ZX-Calculus is part of a broader research area of the use of categorical methods to reason about quantum computing, known as categorical quantum mechanics. From a mathematical standpoint the basic behavior can be captured by the structure of a dagger symmetric monoidal category. Much like ZX-Calculus, categorical quantum mechanics can be fully represented by string diagrams, diagrams originally proposed by Roger Penrose in 1971 as a way to reason about multi linear functions or tensors.

Therefore, ZX-Calculus is a diagrammatic alternative to linear algebra when it comes to reason about quantum gates. It was first introduced in 2008 by Bob Coecke and Ross Duncan, although it wouldn't be until 2017 that its completeness would be proved.

Although ZX-Calculus allows us to reason in a graphical way about linear maps between qubits, it is only one of the languages used for that purpose. Alongside with ZX-Calculus, ZW-Calculus ([Coecke and Kissinger \(2010\)](#)) was also being developed, which provides a smooth way to describe the W-state:

$$|W\rangle = \frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle) \quad (1.1)$$

ZW was the first of these graphical languages to have a complete set of rules. The earliest completeness results for ZX-Calculus were based on the ZW-Calculus.

More recently a new graphical language has been developed to describe in a more natural manner quantum circuits that involve Toffoli gates. It is named the ZH-Calculus ([Backens and Kissinger \(2019\)](#)), as it generalizes the Hadamard gate from ZX-Calculus as a generator. It is, like its counterparts, sound and complete.

The ZX-Calculus has a variety of different applications including the optimization of quantum circuits, which was explored by [Fagan and Duncan \(2019\)](#). Modern quantum computers are still far from being optimal, as they are severely limited by a lot of factors, but mainly they are limited by the coherence time of the qubits. Therefore performing extensive computations becomes almost impossible, as the qubits

may enter a state of decoherence before the computation is complete, rendering the result unusable. To combat this phenomenon it is necessary to perform as many computations as possible in the shortest window of time. Optimization of quantum circuits is a step in this direction

In classical computers, two programs can have the same result without having the same number of computations. The same applies to quantum programs. As long as two programs have the same resulting tensor, they will perform the same computation. Thus, it is only a matter of choosing the one that does it with the least number of gates. The ZX-Calculus allows us to simplify and reduce circuits to their minimal form, although this is not as trivial as one might think, as [de Beaudrap et al. \(2022\)](#) showed.

Until [Duncan et al. \(2020\)](#) the circuit extraction aspect of ZX-diagrams was poorly understood. Articulated with [Kissinger and van de Wetering \(2020b\)](#), a tool for automated reasoning with large scale ZX-diagrams was developed – PyZX. Actually, dealing with large scale ZX-diagrams becomes quickly unreasonable to handle by hand. Although there are tools that are similar to *PyZX* such as *Quantomatic*, they aren't as efficient as *PyZX* since the former was designed with the intent of circuit optimization, and as of today it is the staple for circuit optimization.

Currently the presence of ZX-calculus transcends quantum circuit reasoning and circuit optimization and has been applied to many other areas in quantum computing, such as: quantum error correction, measurement-based quantum computing, quantum natural language processing, among many others.

More specifically ZX-calculus has been used by [Zhao and Gao \(2021\)](#) to analyze the barren plateau phenomenon in training quantum neural networks, and extend the barren plateaus theorem. This is exactly the context for this dissertation.

1.3.2 Quantum Computing and Quantum Machine Learning

Although computer science as a discipline can be traced back to Charles Babbage and Ada Lovelace, it wouldn't be until [Turing \(1936\)](#) that would emerge a formalized understanding of what computer science is.

In tandem the field of study of quantum mechanics was making significant progress. These two fields would be considered completely distinct for the following years. That is until [Benioff \(1980\)](#) introduced the quantum Turing Machine, which uses the theory of quantum mechanics to describe a simple computer. This was the first time these two fields of study converged. With physicists encountering difficulties simulating quantum systems, [Feynman \(1982\)](#) suggested that to efficiently simulate quantum systems one had to use hardware that is built with these intrinsic properties.

The first quantum algorithm was proposed by [Deutsch \(1985\)](#), which was later on expanded by [Deutsch](#)

and Jozsa (1992). They concluded that for a specific class of problems there exists a quantum computation that would solve the problem with an exponential speedup over any classical computation. Although these results had no practical application, they are fundamental for both Shor (1994) and Grover (1996) algorithms both of which attracted a lot of attention to the field of quantum computing.

With these promising results research quickly expanded into a lot of the computer science areas. This is also true of the machine learning area, where Biamonte et al. (2017) postulated that quantum computers would surpass their classical counterpart on machine learning tasks. As of right now, in contrast to machine learning which is its own field of research and has clear economical significance, quantum machine learning is still a mainly theoretical field of study.

1.3.3 The barren plateau phenomenon

The barren plateau phenomenon was first characterized by McClean et al. (2018). It is a phenomenon that appears when training a quantum machine learning algorithm. The optimization gradient quickly disappears, turning the domain of the problem flat, which makes it impossible for the algorithm to find the downward slope. This phenomenon can appear in two different ways:

- Noise Free
- Noise Induced

The former of these surges in PQC¹ that have a large amount of layers and are initialized with random parameters. This leads to the variance of the optimization gradient vanishing exponentially with the number of qubits, making it impossible to train these PQC^s.

The latter occurs in NISQ² (Preskill (2018)) computers. Since the qubits are prone to quantum decoherence as they are strongly susceptible to the environment, the gradient itself vanishes exponentially with the number of qubits, that is the slope itself becomes more and more shallower.

In the context of this dissertation when barren plateaus are mentioned the former variant is the one addressed, unless stated otherwise.

McClean et al. (2018) initially proposed a few methods to avoid the barren plateau phenomenon, such as, using educated guesses for the initial set of parameters and utilizing pre-trained circuits, as they were quite successful combating the same problem in the classical scenario.

¹ A parameterized quantum circuit is specific type of quantum circuit that incorporates gates that have adjustable parameters, making them useful for optimization or machine learning applications.

² The noisy intermediate-scale quantum NISQ era is identified by quantum processors that are not yet capable of fault-tolerance or quantum advantage.

This initial educated guess tries to ensure that the initial conditions of the **PQC** land inside the narrow gorge of the barren plateau, thus ensuring the model is trainable.

Later, [Cerezo et al. \(2021\)](#) proved that not only the initial parameters produced the barren plateaus. The cost function used could also lead to the emergence of this phenomenon.

[Cerezo et al. \(2021\)](#) proved that for a global cost function, this is, when the ansatz is evaluated with respect to all parameters, the barren plateau was guaranteed even for depth 1. Although, if the cost function is only local to a few of the parameters the circuit is able to be trained until a depth of $\log(n)$, with respect to the number of qubits. However, as the depth of the circuit grows from $\log(n)$ to $poly(n)$ the gradients start gradually disappearing in a polynomially, until it reaches a point where they fade exponentially.

Later on [Ragone et al. \(2024\)](#) introduced a framework that unifies all the results about the problem until the date and a Lie algebraic theory to calculate the exact expression of the variance of the cost function.

According to [Ragone et al. \(2024\)](#) the main reasons that originate barren plateaus are the following: excessive circuit expressivity, deeply entangled initial states, global cost functions, and noise channels. Following this work [Cerezo et al. \(2024\)](#) proposed the question ‘Does provable absence of barren plateaus imply classical simulability?’. This work poses a challenge for Quantum Machine Learning as a whole.

The implications of this work could lead to a shift in the Quantum Machine Learning, as **PQCs** that don’t exhibit barren plateaus could be simulable classically. One must then embark in one of two directions following from this work. Either **PQCs** are discarded as a whole when it comes to Quantum Machine Learning, or one tries to find **PQCs** that are hard to simulate classically and don’t present barren plateaus.

Chapter 2

Background

2.1 Introduction to Quantum Computing

2.1.1 States

In classical computing the most basic state of information is the bit, which can either have a value of 0 or 1. The quantum analogue of the bit is the qubit. The qubit exists in the two-dimensional \mathbb{C}^2 Hilbert space, typically represented with the standard basis.

States are usually described in the Dirac *bra-ket* notation, where $\langle\psi|$ stands for the adjoint of the vector $|\psi\rangle$. Notation $\langle\phi|\psi\rangle$ denotes the inner product between ϕ and ψ .

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

These vectors form the *computational basis*. Contrary to the classical bit, qubits can be in any linear combination of these two states.

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

As long as the complex coefficients $\|\alpha\|^2 + \|\beta\|^2 = 1$, it will represent a valid *superposition* of the basis states. This restriction ensures that, when a measurement is done on this qubit one of the two base states $|0\rangle, |1\rangle$ can be obtained with probabilities

$$\|\alpha\|^2 \text{ and } \|\beta\|^2$$

respectively.

2.1.2 Evolution

Evolution proceeds through the application of unitary operators.

An operator is defined as unitary if $U^\dagger U = U U^\dagger = I$. Since the inverse of a unitary operator is also a unitary operator, this means that every computation can be reversed by applying the inverse operators in the inverse order.

A simple example of a unitary matrix is the Pauli-X matrix, which it itself is its own inverse. We can quickly check that by doing:

$$X X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Actually, every Pauli matrix is its own inverse, meaning all of them are involutory.

Evolution is carried out by applying a unitary operator U to a quantum state $|\psi\rangle$, yielding the resulting state $|\psi'\rangle = U |\psi\rangle$. As an example, let us apply the above mentioned Pauli-X matrix to the state $|0\rangle$.

$$X |0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$

As we can see the Pauli-X matrix acts as the quantum equivalent of the NOT operator, that we are so familiar with from classical computing.

2.1.3 Composition

Actually, although a qubit by itself can already perform certain computations that a classical bit simply would not be able to, it is required a larger state space to perform meaningful computations.

This larger state space is created by the composition of two or more qubits. Quantum state spaces combine through the *tensor product* operation \otimes . If we wish to compose n m -dimensional states, the resulting state will be a m^n -dimensional state. Thus the state space of quantum systems scales exponentially with the number of qubits used to represent this state space.

As an example, let us compose the states $|1\rangle$ and $|0\rangle$.

$$|1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10\rangle = |2\rangle$$

Although we can represent a huge amount of states with just the tensor product, most states cannot be described as $|a\rangle \otimes |b\rangle$, for any a and b .

A common example of one such state is the Bell state $|\Phi^+\rangle$.

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

In this case one says that the two qubits are entangled.

2.1.4 Measurements

To obtain information about a quantum system one needs to perform measurements. Quantum measurements are inherently destructive processes, since they collapse the original state into one of the states in the basis used to measure against. This is also known as the *collapse of the wave function*.

Although quantum measurements can be interpreted in many different ways, when it comes to this dissertation it always refers to *projective measurements*, unless stated otherwise.

From a mathematical standpoint measurements are defined by a set of measurement operators that perform an action on the system state space. A measurement, M , must obey the following relation.

$$M^\dagger M = I \quad (2.1)$$

i.e., to be an Hermitian operator. Hermitian operator enjoy the spectral decomposition property:

$$M = \sum_m m P_m \quad (2.2)$$

where m is the eigenvalue of the Hermitian projection operator P_m .

The probability of a measurement having an outcome, m , in a given state, ϕ , is

$$p(m) = \langle \phi | M_m^\dagger M_m | \phi \rangle \quad (2.3)$$

which leaves the system in the following resulting state

$$|\phi'\rangle = \frac{1}{\sqrt{p(m)}} M_m | \phi \rangle \quad (2.4)$$

Projective measurements are usually performed in the *computational basis* of a given qubit.

Let's consider the following measurement operators for the computational basis.

$$M_0 = |0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad (2.5)$$

$$M_1 = |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad (2.6)$$

Now consider the following state $\phi = \alpha |0\rangle + \beta |1\rangle$. We can calculate the probability of each outcome as

$$p(0) = \langle \phi | M_0^\dagger M_0 | \phi \rangle = \langle \phi | M_0 | \phi \rangle = \|\alpha\|^2 \quad (2.7)$$

$$p(1) = \langle \phi | M_1^\dagger M_1 | \phi \rangle = \langle \phi | M_1 | \phi \rangle = \|\beta\|^2 \quad (2.8)$$

2.2 Introduction to ZX

2.2.1 Introduction

ZX-diagrams are composed of *spiders*, of two sorts, Z and X, typically represented by green and red nodes with an associated real number that represents the phase of the *spider*, ranging between $[0, 2\pi[$. Generally, this phase is omitted when zero.

The calculus introduces a set of rewrite rules, which embody two overarching rules:

- Only connectivity matters
- Everything remains equal if you swap red and green nodes

For example, this means that



$$\text{Diagram 1} = \text{Diagram 2} = \text{Diagram 3} \quad (2.9)$$

are all equivalent.

In fact since we can represent the Z *spider* (green) as having one input and two outputs, or two inputs and one output (the same can be said for the X spider), this means that we are able to have a vertical wire without any ambiguity, since it can be represented in either of these two ways.

This means that we can represent multi-qubit gates in ZX-diagrams. Actually, the gate described in (2.9) is the representation of a CNOT gate as a ZX-diagram.

All quantum gates have an equivalent representation as a ZX-diagram. A few of those are the following:

$$\begin{array}{ccc}
 \boxed{X} \rightarrow \text{red circle } \pi & \boxed{Z} \rightarrow \text{green circle } \pi & \boxed{H} \rightarrow \text{yellow square} \\
 \boxed{S} \rightarrow \text{green circle } \frac{\pi}{2} & \boxed{T} \rightarrow \text{green circle } \frac{\pi}{4} & \text{CNOT} \rightarrow \text{green circle } \frac{\pi}{2} \text{ and red circle } \frac{\pi}{2}
 \end{array}
 \tag{2.10}$$

Therefore transforming a quantum circuit into a ZX-diagram is as simple as replacing the gates with their ZX counterpart.

2.2.2 Rewrite Rules

The calculational power of ZX-calculus comes from the powerful set of rewrite rules it provides. The basic rewrite rules are the following:

- Spider Fusion

$$\text{Spider } \alpha \text{ and } \beta \text{ fused} = \text{Spider } \alpha + \beta
 \tag{2.11}$$

- Identity Removal

$$\text{Wire with } \pi \text{ circle} = \text{Wire}
 \tag{2.12}$$

- Hadamard cancellation

$$\text{Two yellow squares} = \text{Wire}
 \tag{2.13}$$

- π commutation

$$\text{Red circle } \pi \text{ then Green circle } \alpha = \text{Green circle } -\alpha \text{ then Red circle } \pi
 \tag{2.14}$$

- State copy

$$(2.15)$$

- Color change

$$(2.16)$$

- Strong complementarity

$$(2.17)$$

- Hopf

$$(2.18)$$

The Hadamard gate is present in the rules above, represented by a yellow box. This is merely a notation used out of convenience, as the Hadamard gate can be decomposed using Euler angles. This notation is simplified even further, in the coming ZX-diagrams as a dashed blue edge/wire, leaving us with:

$$(2.19)$$

There are other rewrite rules in ZX-calculus, all derived from the rules above. One of those derived rules is the following *hadamard pushing*, which is derived from 2.16 and 2.13.

$$(2.20)$$

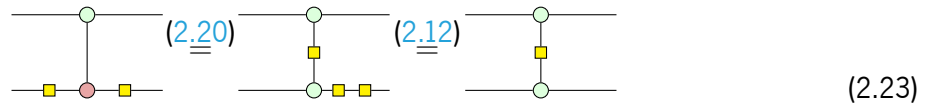
With these representations and the rewrite rules we can already infer other results, such as the representation of CZ gates, with just 2 rules: *hadamard pushing* and *hadamard cancellation*.

$$(2.21)$$

First we need to represent the circuit as a ZX-diagram, yielding.



Applying the rules described above we get the following:



ZX-calculus has a set of rewrite rules that, with a valid ZX-diagram, lead to a new ZX-diagram equivalent to the one we started with. However it is not guaranteed that the resulting ZX-diagram is still a valid circuit. Actually, every circuit has a valid ZX counterpart, but not every ZX-diagram has a direct circuit equivalent. The reason for this is that the ZX-calculus allows for an expressiveness that quantum circuits are unable to reproduce.

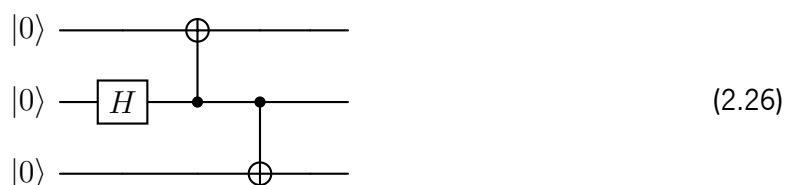
Let's take the following ZX-diagram for example:



Which has not a direct mapping to a circuit. But it does represent three qubits entangled in the Z-basis, that is, these three qubits are in the GHZ (Greenberger–Horne–Zeilinger) state.

$$|GHZ\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}} \quad (2.25)$$

Let's now take a look the circuit that implements the GHZ state.

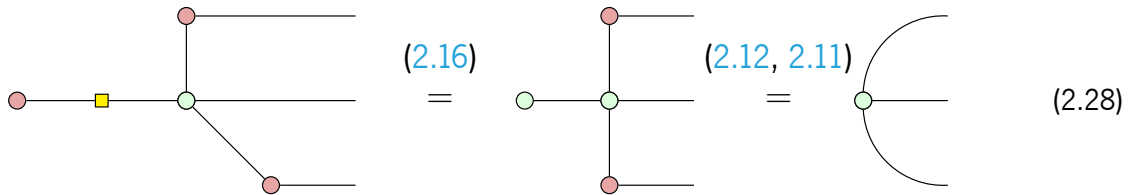


Since ZX-calculus is sound and complete [Jeandel et al. \(2020\)](#) this circuit and the ZX-diagram [\(2.24\)](#) above must represent the same tensor.

Using the rewrite rules [\(2.10\)](#) above we can represent the circuit that implements the GHZ state [\(2.26\)](#) as the following ZX-diagram:



Now, using a few of the rewrite rules from the ZX-calculus (more specifically the fusion, color changing and identity removal rules), we get the following diagram.



Thus, as a first advantage of ZX-calculus, it allows us to have a simplified notation to represent and reason about quantum circuits.

Actually, one of the biggest advantages of reasoning with ZX-diagrams instead of circuits is that a lot of equalities that one has to be aware when reasoning with circuits are just given as *free properties*. But what does that mean exactly?

For example, an equality one has to be aware when reasoning with circuits is that gates with a Z-phase commute with the control of a CNOT gate. This becomes trivial when expressed in ZX.

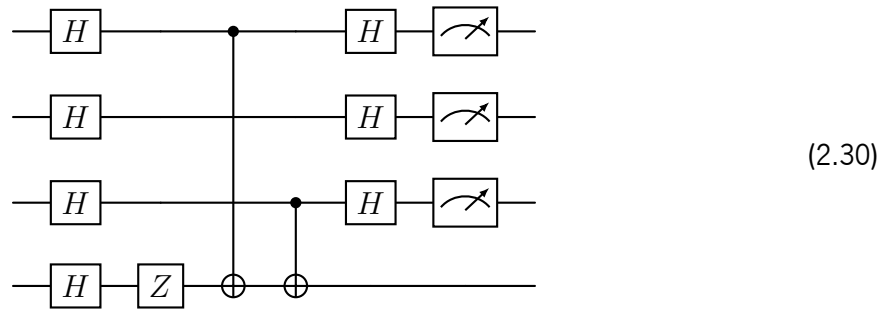


That is because green *spiders* commute through one another, and because of the second overarching rule of ZX-calculus (*Everything is remains equal if you swap red and green nodes*). The same applies to red *spiders*. This subtle rule is able to abstract a lot of equalities.

2.2.3 Circuit Rewriting

Bernstein-Vazirani

ZX-calculus can be used for rewriting or even simplifying quantum circuits. Let us look at some examples. The following circuit implements the Bernstein-Vazirani 3 qubit circuit. This algorithm tries to learn an encoded binary string in a function, which, in this example, is the binary string $s = 101$.

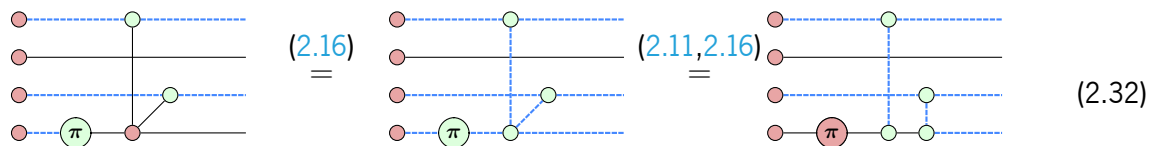


Although ZX-calculus won't be able to simplify this circuit much further, as it already is quite simple, we can infer an equivalent circuit that utilizes CZ gates instead of CNOT gates.

First we need to represent the circuit as a ZX-diagram. Using the equivalences established in 2.10 this is quite trivial.

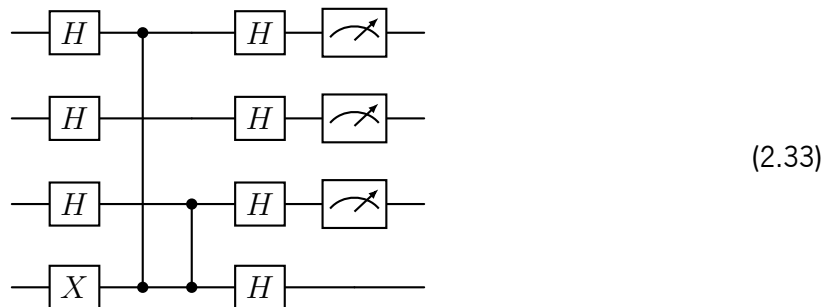


In the following diagrams the edges with Hadamards gates have been exchanged for dotted blue lines. There were also applied 2 rules between the diagrams; *spider fusion* 2.11 and *Hadamard cancellation* 2.13.



As a first step we apply the *color change* rule 2.16. Then we *unfuse the spiders* 2.11 and apply a final *color change* 2.16 to reduce the number of Hadamard gates in the resulting circuit.

Again using the equivalences from 2.10 we get the following circuit.

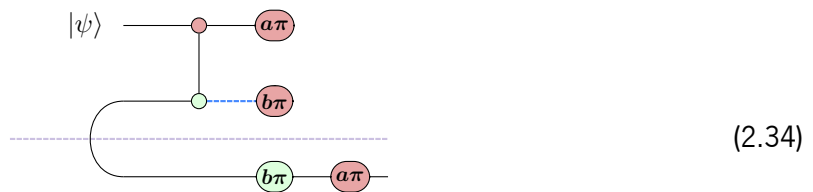


In fact if we compare both tensor products resulting from the circuits we will see that both circuits represent the same tensor.

Note that the rules applied here were chosen with the exact purpose of resembling a circuit, so that the circuit extraction in the end could be a simple task.

Quantum Teleportation

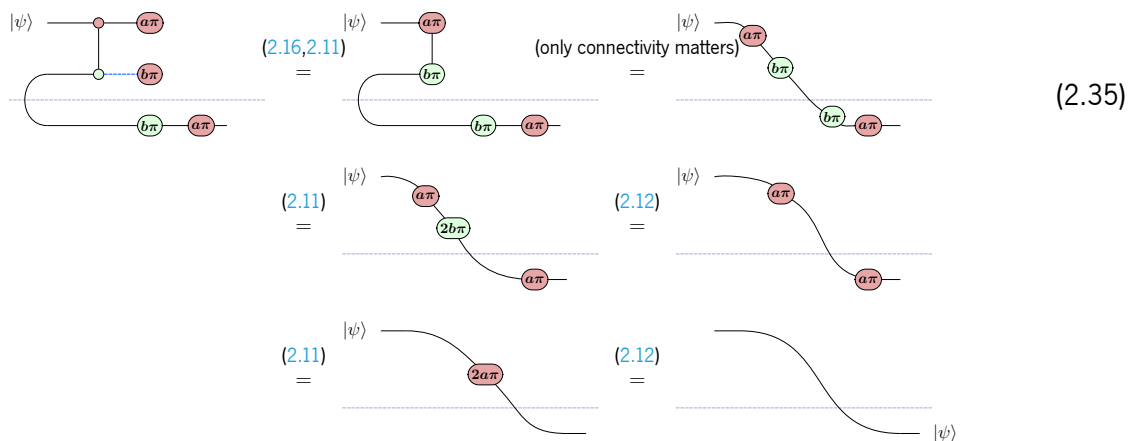
There are a lot of different implementations for quantum teleportation, but they all follow the same principle: There is a quantum state $|\psi\rangle$ that we want to send to another partner in the protocol. For that we need an entangled pair of qubits that act as a quantum channel. Finally, some corrections that will need to be sent over a classical channel. This can be represented by the following ZX-diagram.



The scenario includes an entangled pair of qubits denoted by the curved wire, and the measurements denoted the X-spiders on the top two qubits. These measurements will then be applied to the destination qubit in order to guarantee the state remains the same. It is important to note that the measurements will force the quantum state to collapse resulting in either $|0\rangle$ or $|1\rangle$.

This preserves the no-cloning theorem as the state is destroyed in the qubit it originates from, but this result will also aid us in the following proof, through the *identity removal* rule 2.12.

This is due to the phases of the *spiders* in the ZX-diagram being a multiple of 2π , which allows to reason as follows:



As a first step we perform a *color change* 2.16 followed by two *spider fusions* 2.11. In the following step no rewrite rule was applied, the diagram was simply bent. Afterwards we perform back to back *spider fusion* 2.11 and *identity removal* 2.12.

This is one example of the abstraction power provided by the ZX-calculus. The result is more intuitive than its circuit counterpart, as we can literally visualize the wire bending and the destination qubit becoming the original state $|\psi\rangle$.

2.2.4 T-count reduction

As stated by [Eastin and Knill \(2009\)](#) it is impossible to have a universal gate set without any non-transversal operations. Thus, at least one operation has to be non-transversal. The usual set of operators for universal computation is the *Clifford + T*, that is the Clifford gates (a set of transversal gates) and the T-gate $R_Z = (\frac{\pi}{4})$ (a non-transversal gate). Since the T-gate is non-transversal its implementation is extremely expensive, due to the process necessary to obtain a pure "magic" state. This pure "magic" state needs to be *distilled* from other noisy "magic" states in a process called *magic state distillation*. Finally, it is injected into the circuit in order to act as a T-gate. Since the implementation of such gates is so expensive, a fundamental aspect of circuit optimization is the reduction of the amount of T-gates contained in the circuit.

This is achieved, in the ZX-calculus, through the set of rewrite rules. Namely the spider fusion rule [2.11](#), making two T-gates fused together to become a S-gate $R_Z = (\frac{\pi}{2})$. These S-gates are Clifford, and as such, are much more efficient to implement than their square root counterpart. The other scenario where we can use the spider fusion rule is with a T and a T^\dagger , that is because $T^\dagger T = T T^\dagger = I$.

In fact, these simple rewrite rules are just the beginning of T-count optimization with the ZX-calculus, a starting point for many more sophisticated optimization techniques.

2.2.5 PyZX

Reasoning with ZX-diagrams quickly becomes difficult, because of the need to deal by hand with huge diagram, as the circuits grow in depth. A number of tools like *PyZX* are capable of assisting us when it comes to circuit optimization.

As [Kissinger and van de Wetering \(2020b\)](#) stated, *PyZX* is capable of simplifying ZX-diagrams that contain thousands of vertices in a few seconds. Due to the nature of ZX-calculus *PyZX* is best suited to deal with Clifford circuits as well with phase gates. This makes it the ideal tool to deal with quantum machine learning algorithms, as they are mostly composed of rotations on the Z-axis.

Although the ZX-calculus is very versatile, it cannot directly reason with Toffoli and CCZ gates, making *PyZX* not as efficient in dealing with circuits that heavily rely on these types of gates.

Chapter 3

Modelling algorithms in ZX

This chapter aims at describing the different quantum protocols that were object of study in this dissertation as ZX-diagrams. Such protocols range through three different categories: *quantum walks*, the *quantum approximate optimization algorithm* and *hardware efficient ansatzs*.

Their ZX representations, or models, will then be used in the Chapter 4 in two distinct manners. On the one hand, quantum gate optimization routines will be applied to the *quantum walks* and the *quantum approximate optimization algorithm* to seek for corresponding optimized versions.

On the other hand, ZX models of *hardware efficient ansatzs* will be analyzed to detect the presence of barren plateaus. This resorts to analytical barren plateau detection techniques derived from the ZXW-calculus (a variant that incorporates both the ZX- and the ZW-calculus). These techniques allow for the characterization of variance of the parameters as a function of the number of qubits in the ansatz, thus, making possible the detection of barren plateau phenomena.

3.1 Quantum Walks

3.1.1 Introduction

Thought of as the quantum counterpart to classical random walks, quantum walks as per [Venegas-Andraca \(2012\)](#) provide an interesting technique in algorithmic design, with applications in unstructured search, graph algorithms and communication protocols.

Differently from the classical case, where the walker next move follows the result of some sort of random choice, in a quantum setting evolution typically proceeds in an equally weighed superposition of possible moves through the iteration of a unitary operator, without resorting to intermediate measurements. This results in a very rich dynamics, in which the design of the evolution operator, and even seemingly innocent differences in its phase and in the initial state, determine complex 'walking patterns' which differ

greatly both among them and in contrast to the classical setting.

The relevance of quantum walks as a tool for algorithmic design justifies both a better understanding of their behavior and the optimization of their implementation, namely to increase resilience to decoherence phenomena.

As such, the following sections are dedicated to modelling the *continuous time quantum walk*, *coined quantum walk* and the *staggered quantum walk* as ZX-diagrams, according to their specifications.

3.1.2 Continuous-Time Quantum Walks

Continuous-time quantum walks were first considered as a quantum analogue of the classical continuous time random walks. Such a walk is determined, in contrast to a discrete time quantum walk, by a time-varying unitary matrix instead of a, e.g., coin operator, as is the case on coined quantum walks. This is what dictates how the walker evolved over the given graph.

Implementation

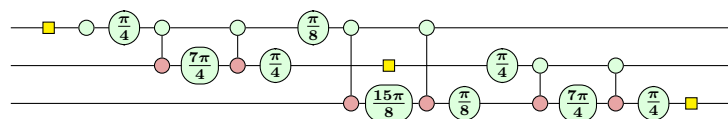
The Continuous-Time Quantum Walk can be easily translated to a quantum circuit. It is simply the Quantum Fourier Transform over n qubits, followed by the time-varying unitary evolution operator, followed by the Inverse Quantum Fourier Transform.

Thus we have the following circuit:

$$\begin{array}{c}
 \text{---} \\
 \text{---} \\
 \vdots \\
 \boxed{QFT} \quad \boxed{U} \quad \boxed{QFT^\dagger} \\
 \vdots \\
 \text{---}
 \end{array} \tag{3.1}$$

The following implementation is directly derived from the work of Santos (2021).

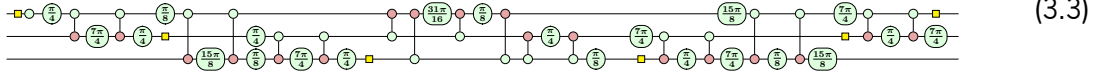
The Quantum Fourier Transform has a well known circuit representation. The following is a ZX-diagram representation of that same 3-qubit circuit with the decomposition of the controlled-phase gates into its basic gates.



$$\tag{3.2}$$

One can also derive the circuit for the Inverse Quantum Fourier Transform just from this diagram. All one needs to do is apply the control-phase gates and the Hadamard gates in the reverse order with the reverse phase.

Thus a 3 qubit implementation of this quantum walk yields the following ZX-diagram.



3.1.3 Coined Quantum Walk

Contrasting classical random walks the coined quantum walk walker does not have its position determined by stochastic transitions over the state space, but through the properties of quantum mechanics. The Coined Quantum Walk is a discrete time implementation of a quantum walk.

An implementation is characterized by the use of a unitary operator that dictates how the walker evolves over time. The position of the walker is specified by a coin operator, usually the Hadamard operator as it is an unbiased operator, and a shift operator that is dependant on the coin operator.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Although the Hadamard operator is unbiased it is not necessary that the coin operator being unbiased. Using different operators as a coin allows for other probabilities distributions.

The shift operator is simply an increment and decrement layer controlled on the coin operator.

A one-step model can be found in the appendix (A.1) as it is too extensive to be detailed in the main text.

3.1.4 Staggered Quantum Walk

In contrast to conventional, coin-based quantum walks, which proceed straightforwardly from one vertex to another, the staggered variant [Portugal et al. \(2016\)](#) takes advantage of forming partitions of graph cliques¹ over the graph structure of the walking space. Each partition forms a tessellation whose elements do not overlap. The set of cliques in each tessellation must cover all vertices of the graph, and the set of tessellations $\{T_1, T_2, \dots, T_k\}$ chosen must cover all the edges.

Then a unit vector, typically encoding a uniform superposition, is associated to each clique so that the vector belongs to the subspace spanned by the corresponding vertices. I.e.,

$$|u_j^k\rangle = \frac{1}{\sqrt{|\alpha_j^k|}} \sum_{l \in \alpha_j^k} |l\rangle, \quad (3.4)$$

¹ A clique is a subset of vertices of an undirected graph such that every two distinct vertices are adjacent.

where α_j^k is the j^{th} polygon in the k^{th} tessellation.

This way each tessellation k gives rise to an operator

$$H_k = 2 \sum_{j=1}^p |u_j^k\rangle \langle u_j^k| - I. \quad (3.5)$$

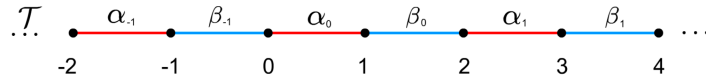
which propagates the probability amplitude locally, in each clique. The composition of all such operators defines the evolution operator, which, by solving the the time-independent Schrodinger equation, is equivalent to

$$U = e^{i\theta_k H_k} \dots e^{i\theta_2 H_2} e^{i\theta_1 H_1}, \text{ where } e^{i\theta_k H_k} = \cos(\theta_k)I + i \sin(\theta_k)H_k \quad (3.6)$$

since $H_k^2 = I$, meaning that the Hamiltonian is a reflection operator that, when expanded in a Taylor series, generates a local operator.

As an elementary example consider a line where the following two tessellations (depicted in red and blue below) are defined

$$T_\alpha = \{\{2x, 2x + 1\} : x \in \mathbb{Z}\} \text{ and } T_\beta = \{\{2x + 1, 2x + 2\} : x \in \mathbb{Z}\}. \quad (3.7)$$



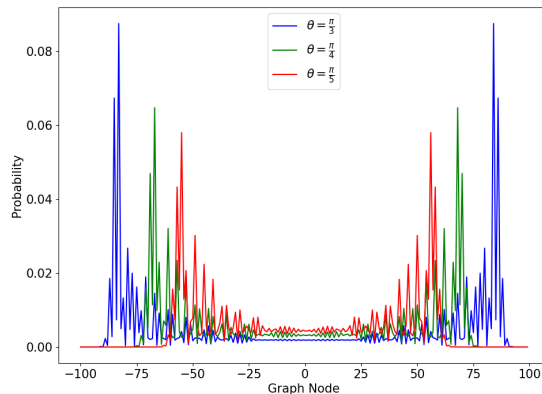
Thus,

$$|\alpha_x\rangle = \frac{|2x\rangle + |2x + 1\rangle}{\sqrt{2}} \text{ and } |\beta_x\rangle = \frac{|2x + 1\rangle + |2x + 2\rangle}{\sqrt{2}}, \quad (3.8)$$

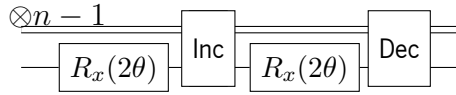
yielding Hamiltonians

$$H_\alpha = 2 \sum_{x=-\infty}^{+\infty} |\alpha_x\rangle \langle \alpha_x| - I \text{ and } H_\beta = 2 \sum_{x=-\infty}^{+\infty} |\beta_x\rangle \langle \beta_x| - I. \quad (3.9)$$

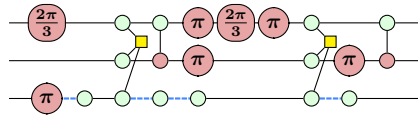
Therefore, $U = e^{i\theta H_\beta} e^{i\theta H_\alpha}$ is the evolution operator. The probability distribution on a line after 50 steps, starting at $|+\rangle$, for different values of θ , is depicted below, noticing that the walker is more likely to be found further away from the origin as the angle increases.



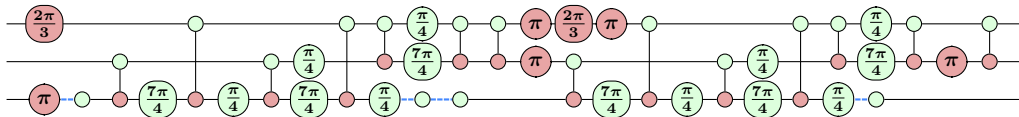
A circuit implementation of the staggered model can be found in Santos (2021). As expected, it resorts to the joint tessellation dynamics, in contrast to a coined quantum walk that uses a coin as a shift operator. For the example discussed above, it yields



where $R_x(\theta) = e^{-\frac{i\theta X}{2}}$ and the Inc(rement) and Dec(rement) circuits have the usual implementation through generalized Toffoli gates. An implementation for a 3 qubit staggered quantum walk, starting at $|4\rangle$ and taking $\theta = \frac{\pi}{3}$, which maximizes propagation, is represented in a ZX diagram as



which takes advantage of the ZH-calculus H-box notation for representing Toffoli gates in a concise way. Expanding Tofolli into its basic gates leads to



3.2 Quantum Approximate Optimization Algorithm

3.2.1 Introduction

Fault-tolerant quantum computers capable of executing the most promising quantum algorithms are still a far away effort. Even so, the development of useful quantum algorithms to be executed in NISQ has had significant progress.

Variational Quantum Algorithms (VQA) are one of such algorithms which take advantage of the current landscape in the realm of quantum computing.

All **VQA** share a common structure: a task is encoded into a parameterized cost function that is evaluated using a quantum computer, and a classical optimizer which trains the relevant parameters.

A **VQA** is a hybrid loop of a parameterized quantum circuit that runs in quantum computer, followed by an optimizer that runs on a classical computer. This optimizer updates the parameters in order to minimize the cost function, based on the outputs of the quantum circuit.

This hybrid loop allows the circuit to stay shallow, thus somehow controlling the noise from **NISQ** devices.

The Quantum Approximate Optimization Algorithm (**QAOA**) is one of the most promising algorithms in the category of **VQA**. As it aims to solve combinatorial optimization problems that were previously unsolvable with classical methods.

The **QAOA** is capable of finding good approximations to many optimization problems in the Quadratic Unconstrained Binary Optimization (**QUBO**) category. A few examples of problems in this category are the Maximum Cut (MaxCut), Minimum Vertex Cover, Maximum Independent Set (MIS), among many others.

3.2.2 Constructing the QAOA Hamiltonian

Let's take the Maximum Cut problem as an example, as this problem is *NP-complete* and as such any **QUBO** problem is at most as difficult as the MaxCut.

We can formulate the MaxCut as an optimization problem with the following objective function:

$$\max_v \frac{1}{2} \sum_{ij \in E} (1 - v_i v_j) \quad v_i, v_j \in \{-1, +1\} \quad (3.10)$$

where ij represents an edge belonging to the edge set E of a given graph, v_i and v_j represent the cut a vertex is part of. Thus an edge will only be cut if the vertices are in distinct cuts.

In order to solve an optimization problem in a quantum computer, one needs to characterize a quantum Hamiltonian that encodes the solution to the problem in its highest energy eigenstate.

Therefore, it is necessary to convert the objective function in 3.10 into a Hamiltonian that encodes the solution². This Hamiltonian is diagonal with the values of the diagonal corresponding to the values of the objective function, e.g.

² The eigenstate of this Hamiltonian can be measured in the computational basis. Thus one may obtain the solution with certainty.

$$C = \begin{pmatrix} f(0\dots 00) & \dots & 0 & 0 & 0 \\ 0 & f(0\dots 01) & \dots & 0 & 0 \\ \vdots & & \ddots & & \\ 0 & 0 & \dots & f(1\dots 10) & 0 \\ 0 & 0 & 0 & \dots & f(1\dots 11) \end{pmatrix} \quad (3.11)$$

Therefore,

$$C |x\rangle = f(x) |x\rangle \quad \forall x \in \{0, 1\}^n \quad (3.12)$$

of course, this represents a major challenge, as the Hamiltonian we are describing is indeed too large ($2^n \times 2^n$) to be constructed explicitly and constructing it would mean solving the optimization problem. Alternatively we need to construct the Hamiltonian in a more compact way. The MaxCut Hamiltonian is obtained mapping the variables v_i onto the eigenvalues of Z . Thus yielding

$$C = \frac{1}{2} \sum_{ij \in E} (I - Z_i Z_j) \quad (3.13)$$

This comes from the nature of the Pauli-Z operator.

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (3.14)$$

Whose eigenvalues, -1 and $+1$, correspond to the following eigenvectors in the computational basis.

$$Z |0\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle \quad (3.15)$$

$$Z |1\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} = (-1) |1\rangle \quad (3.16)$$

This means that when we apply the Pauli-Z operator to state $|0\rangle$ it acts as an identity operator, but when we apply it to state $|1\rangle$ we also obtain state $|1\rangle$ although affected by a phase factor.

Thus, when we use the notation $Z_i |x_0 \dots x_n\rangle$ it means we are applying a Pauli-Z operator to the i -th qubit, leaving all others unaffected. Similarly the notation $Z_i Z_j |x_0 \dots x_n\rangle$ means we are applying a Pauli-Z operator to the i -th and j -th qubit, leaving all others unaffected.

Therefore, the following variable change

$$x_i = \frac{1}{2}(1 - v_i) \quad (3.17)$$

specifies a move between binary variables and spin variables, as appropriate, as shown in

$$v_i = 1 \mapsto x_i = 0, \quad (-1)^{x_i} = 1 = v_i \quad (3.18)$$

$$v_i = -1 \mapsto x_i = 1, \quad (-1)^{x_i} = -1 = v_i \quad (3.19)$$

Such a variable change leads to the following objective function.

$$\max_x \frac{1}{2} \sum_{ij \in E} (1 - (-1)^{x_i} (-1)^{x_j}) \quad x_i \in \{0, 1\} \quad (3.20)$$

Actually, $(-1)^x$ is just a compact way to describe the Pauli-Z operator that we saw in 3.16 as $Z|x\rangle = (-1)^x|x\rangle$. Then this yields the Hamiltonian described in 3.13.

This process allows us to describe an Hamiltonian that encodes the solution with a linear amount of terms in respect to the edges of the graph.

The Hamiltonian just built is specific to the MaxCut problem, although there is a way to implement any Boolean function, as extensively discussed in the work of Hadfield (2021).

3.2.3 The Ansatz

The **QAOA** prepares a parameterized ansatz state composed of the problem hamiltonian C , described in 3.2.2, and the mixer Hamiltonian B , which can be simply described as

$$B = \sum_i X_i \quad (3.21)$$

where each X_i is a parameterized R_X in the i -th qubit.

This parameterized ansatz has the following form

$$|\psi(\beta, \gamma)\rangle = e^{-i\beta_p B} e^{-i\gamma_p C} \dots e^{-i\beta_1 B} e^{-i\gamma_1 C} H^{\otimes n} |0\rangle \quad (3.22)$$

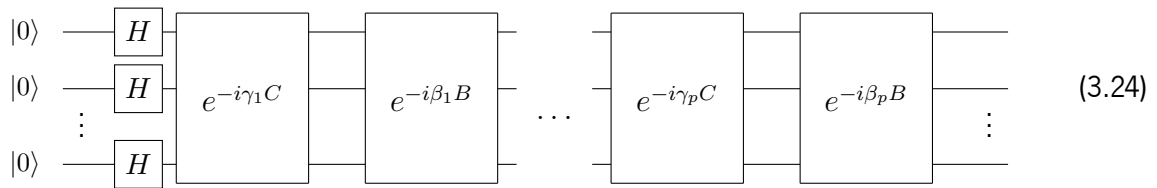
where B and C are the mixer and the problem Hamiltonians, respectively. The layer i of the mixer Hamiltonian is parameterized by β_i , similarly the problem Hamiltonian is parameterized by γ_i for the i -th layer. For $1 \leq i \leq p$.

Then a classical optimizer is used in order to vary the parameters of both the problem Hamiltonian and the mixer Hamiltonian and find the optimum parameters that maximize the function

$$f(\beta, \gamma) = \langle \psi(\beta, \gamma) | C | \psi(\beta, \gamma) \rangle \quad (3.23)$$

For a number p , such that $p \rightarrow \infty$, of layers of the **QAOA** ansatz it can find the optimal solution. Running a quantum algorithm with an infinite amount of layers is untractable, for a value of p that is small there is evidence for quantum advantage.

The **QAOA** ansatz can be represented in the following circuit form.



The phase-separation Hamiltonian (problem Hamiltonian) is composed of the following 2-qubit circuit.



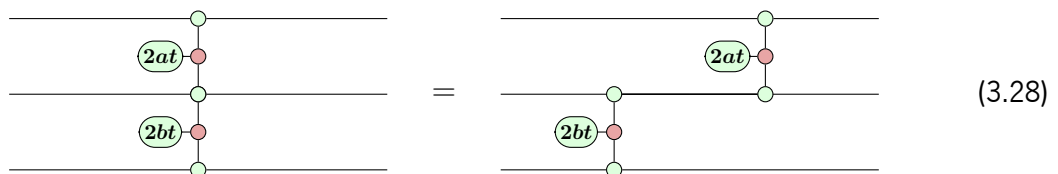
These phase-separation Hamiltonians are typically represented in the form of a ZX-diagram as a phase-gadget.



A property of the circuit 3.25 is that it commutes through other phase-separation circuits. For example, consider.



This property means that we are free to move these phase gadgets as we see fit.



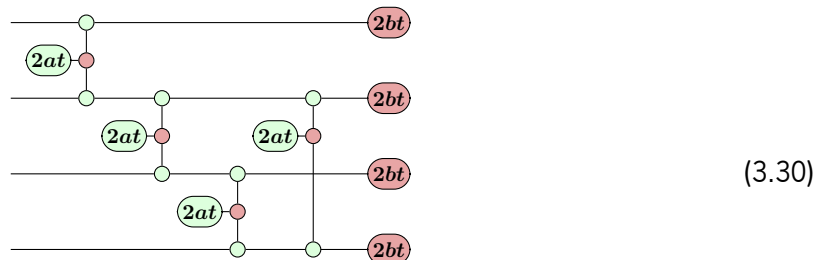
Although this wasn't trivial to observe in the circuit form, it is clear that in the ZX-diagram form that the phase-gadgets simply commute through one another. This is due to the fusion and un-fusion (2.11) rule on the qubit belonging to both phase-gadgets.

What was described is only half of our QAOA circuit, as we are still missing the mixer Hamiltonian. Fortunately it is simply a sum of Pauli-Xs on all qubits, and, therefore, it can be represented in the form of a ZX-diagram as an arbitrary single qubit rotation along the X-axis on all qubits.

With these building blocks we can derive the QAOA MaxCut implementation for any graph we desire. Let's use the following graph as an example.



Then using the circuit in 3.25 for each edge, followed by the mixer Hamiltonian on all qubits we get the following ZX-diagram.



We can further use the rewrite rules (2.2.2) to obtain the following diagram.

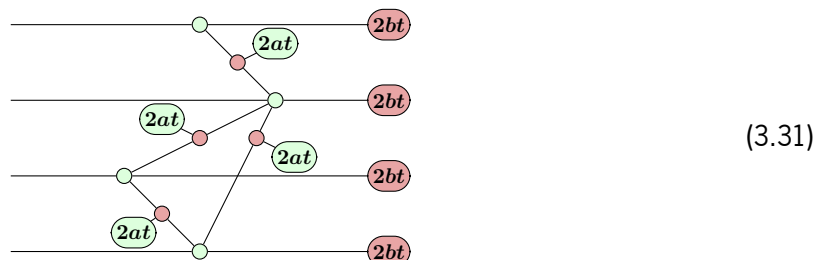


Figure 1: The repeating pattern of the QAOA ansatz for the graph 3.29

This shows us, once again, how much more expressive the ZX-calculus is in comparison to the circuit model. As the graph for the problem is quite literally encoded into the ZX-diagram. This also shows us one of the main characteristics of NISQ algorithms, that is, being able to solve the problem at hand only using the necessary amount of qubits to encode the problem itself.

3.3 Hardware Efficient Ansatz

3.3.1 Introduction

In Quantum Machine Learning not all problems have a well defined and known ansatz that solves the specific problem, or, in a more general sense, there isn't a way to derive an Hamiltonian that encodes the solution to the problem.

When we have such problems we might need to use a broader, more general ansatz that is able to represent and explore the state space necessary to solve the problem at hand. This is where one must resort to Hardware Efficient Ansatzs (**HEA**). These are parameterized quantum circuits which aim to use the resources available on the quantum computer on which they are being executed, taking into consideration the limitations of the hardware itself. This is, they utilize all the resources available to them and construct a general circuit that might contemplate the solution to the problem at hand, and then through gradient descent find the optimal solution.

An example of one such ansatz is the following circuit.

$$\begin{array}{ccccccc} |0\rangle & \text{---} & \boxed{R_x} & \text{---} & \boxed{R_z} & \text{---} & \\ |0\rangle & \text{---} & \boxed{R_x} & \text{---} & \boxed{R_z} & \text{---} & \\ |0\rangle & \text{---} & \boxed{R_x} & \text{---} & \boxed{R_z} & \text{---} & \\ |0\rangle & \text{---} & \boxed{R_x} & \text{---} & \boxed{R_z} & \text{---} & \end{array} \quad (3.32)$$

Figure 2: Example of a simple **HEA**. The dashed box denotes the repeatable pattern of the circuit. This is also the ansatz denoted as 'Circuit 1' in the work of [Sim et al. \(2019\)](#).

These ansatzs may also utilize 2-qubit gates as a way to perform the entanglement of two or more qubits, although, as stated above, this is limited to the capabilities of the quantum hardware.

However, these **HEA** tend to exhibit the barren plateau phenomenon. Thus limiting severely their usefulness. Therefore, this warranted a change of analysis conducted comparatively to previously shown protocols. The **HEA** were subjected to analytical detection of barren plateaus, a technique that aims to characterize the existence of a barren plateau, without the execution of the circuit in a quantum computer. Up until now, every ZX-diagram in this dissertation conveniently ignored every scalar factor respective to itself, as they are irrelevant for the analysis conducted. However, this is not the case for the analytical detection of barren plateaus.

3.3.2 Analytical Barren Plateau Detection

Zhao and Gao (2021) introduced a framework for calculating the variance of parameters in a PQC utilizing ZX-calculus. They derived the parameter-shift in the form of a ZX-diagram, which when plugged into the parameters allows to retrieve the corresponding parameter variance. In conjunction with a set of rules that allows for splitting diagrams, made possible to convert the ZX-diagrams into *tensor networks*, and consequently calculate the variance of a parameter. This was the first time that the barren plateau problem was analyzed with a framework of the ZX-calculus.

For this however, it still required to leave ZX-calculus and go into *tensor networks* to proceed with the computation.

Following this work Wang et al. (2024) introduced a framework that allows for the calculus of the parameter variance exclusively within the framework of the ZXW-calculus.

As it turns out, ZX-calculus, just by itself, is very useful in Quantum Circuit Optimization and Measurement-Based Quantum Computing, ZW-calculus is very good at Summing, Products and Quantum Optics. When both are combined new properties emerge that are extremely useful for Hamiltonian Exponentiation and more importantly, at least in the context of this dissertation, Diagram Differentiation, which has several applications in the realm of Quantum Machine Learning.

The work of Wang et al. (2024) introduces a new generator, the *triangle*

$$\begin{array}{c} \leftarrow \\ \leftarrow \end{array} = 2 \begin{array}{c} \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} \quad (3.33)$$

which enjoys a set of rewrite rules. The following, capturing the change of basis are used below

$$\begin{array}{cc} \text{---} \circ \leftarrow = \text{---} \circ & \text{---} \circ \leftarrow = \text{---} \circ \\ \text{---} \circ \rightarrow = \text{---} \circ & \text{---} \circ \rightarrow = \text{---} \circ \end{array} \quad (3.34)$$

The triangle construction is essential for the following theorem presented in the same work:

$$\text{Var}\left(\frac{\partial \langle H \rangle}{\partial \theta_i}\right) = \text{Diagram} \quad (3.35)$$

which allows, when computing the variance for an arbitrary number of qubits, to characterize the variance of the parameters in relation to the number of qubits and, consequently, the existence of a barren plateau.

This construction is composed of two distinct building blocks denoted *cycles*

$$\text{Diagram 1} \quad \text{Diagram 2} \quad (3.36)$$

The one on the left is plugged into the variance parameter θ_i and the one on the right is plugged into every other parameter θ_j , such that, $j \neq i$.

This type of analysis necessitates a different representation of the red spiders, the *pink spider*, in the following diagram is due to the work of Koch (2022).

$$\text{Diagram} = n \left\{ \text{Diagram} \right\} m = 2^{-\frac{m+n-2}{2}} \text{Diagram} \quad (3.37)$$

The reason for this extended notation is that some rewrite rules introduced scalars. With this notation, only the color change rule introduces scalars. This makes it much easier to account for all the scalars during the analysis mentioned.

In spite of this, the fusion of *pink spiders* also introduces a scalar.

$$\text{Diagram} = 2^{n-1} \text{Diagram} \quad (3.38)$$

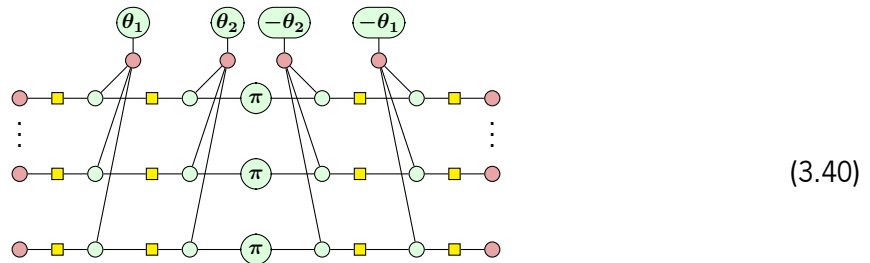
3.3.3 Example Ansatz

The following ansatz will be used as an early example to employ the techniques for analytical barren plateau detection.



Compared to other HEA this one lacks a lot of expressivity that is required to perform meaningful computations. This is due to all the qubits being rotated according to the same parameters θ_1, θ_2 . However, this makes for a great example on how to calculate the variance of this ansatz.

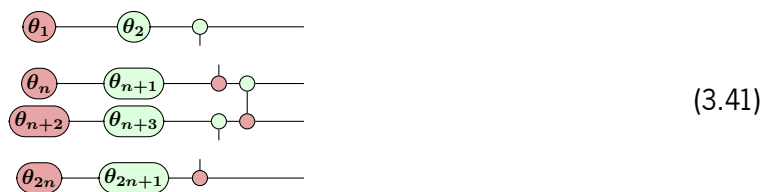
Nonetheless, the ansatz, just by itself, is insufficient to calculate its variance. This is due to the fact that the variance isn't intrinsic to the ansatz itself. But instead, to the Hamiltonian that it is being measured against. Thus we must construct the following circuit



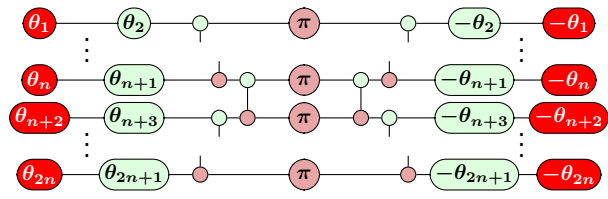
Where we evaluate the expectation value $\langle H \rangle$ for the Hamiltonian, $H = Z^{\otimes n}$

3.3.4 Sim Inspired Ansatz

We also utilize these same analytical barren plateau detection techniques to an ansatz inspired in the work of Sim et al. (2019).



Again, we must construct the following circuit



(3.42)

Where we evaluate the expectation value $\langle H \rangle$ for the Hamiltonian, $H = X^{\otimes n}$

Chapter 4

Optimization and Analysis

The goal of this chapter is to use the ZX models presented in Chapter 3 to explore their optimization (namely in the cases related to the *quantum walk* and *quantum approximate optimization algorithm*) and/or analysis. The resulting circuits are then analyzed for emergent properties and the impact of the optimization in the original protocol discussed.

The case related to *hardware efficient ansatz* will be subject to the techniques presented in the Subsection 3.3.2 to detect the presence of barren plateau phenomena.

4.1 Continuous Time Quantum Walk

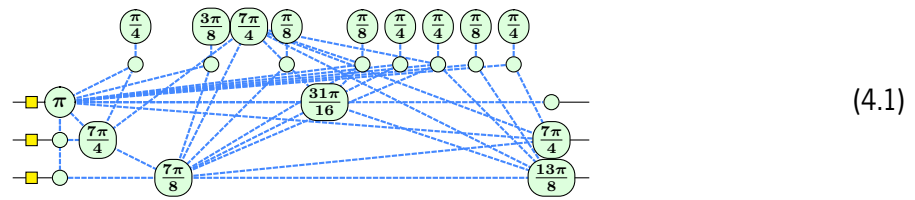
Consider the ZX-diagram in 3.3 corresponding to a 3-qubit Continuous Time Quantum Walk. It contains 46 gates of which

- 22 are T-gates
- 24 are Cliffords, of which
 - 18 are CNOTs
 - 6 are Hadamards

This specific implementation of a quantum walk might not benefit from the optimization routines that *PyZX* (2.2.5) provides.

This is largely due to the nature of this implementation itself. Finding a more optimized version would require the existence of a simpler, more gate efficient, instance of the Quantum Fourier Transform. Moreover, differently from what happens in other quantum walk implementations, the Continuous Time Quantum Walk encodes the length into its Unitary Operator. Thus, having a longer walk does not mean an increase in the number of total gates in the circuit.

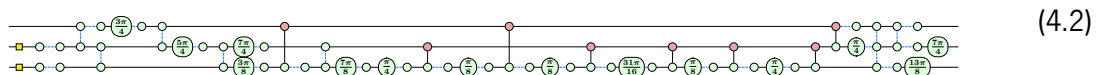
Nonetheless, when we apply the optimization routines that *PyZX* provides we obtain the following ZX-diagram.



The diagram no longer resembles the circuit model necessary to execute this specific instance of a Quantum Walk. Thus, we need to extract a *circuit-like* ZX-diagram from this *graph-like*¹ diagram.

Unfortunately, circuit extraction is not a trivial problem², which explains why the circuit extracted, most likely, will not be the most optimal circuit that represents the same tensor as 4.1.

This is, in fact, the case for this diagram as the circuit extraction yields the following *circuit-like* ZX-diagram.



This ZX-diagram contains 55 gates of which

- 14 are T-gates
- 41 are Cliffords, of which
 - 8 are CNOTs
 - 7 are CZs
 - 26 are Hadamards

The resulting circuit is able to reduce the T-count by 8, and the total amount of 2-qubit gates from 18 to 15, but, in turn, adds 14 extra gates to the circuit, most of them Hadamards.

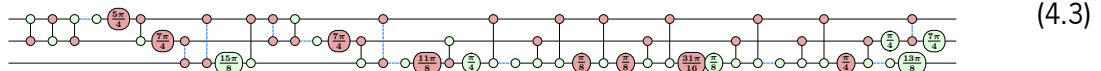
At first glance this optimization made the original circuit more inefficient. However, when it comes to circuit optimization targeting **NISQ** devices, one of the most important metrics, if not the most important one, is the T-count. Therefore, a reduction on the T-count, even if small, has a tremendous effect when it comes to the execution of the circuit. This is due to the fact that T-gates are so expensive to implement, comparatively to the Clifford set which can be implemented pretty much for ‘free’.

¹ A *graph-like* ZX-diagram is a ZX-diagram that only contains Z-spiders, all the Z-spiders are connected through Hadamard edges, there are no parallel edges, every input/output is connected to a Z-spider and each Z-spider is connected to at most one input/output.

² In fact, it can be #P-Hard as [de Beaudrap et al. \(2022\)](#) stated

Note that, the resulting ZX-diagram still contains a considerable amount of Z-spiders and Hadamard gates. This suggests that the resulting *circuit-like* ZX-diagram still exhibits some of the *graph-like* nature prior to the circuit extraction. And as a consequence, it can be further optimized, mainly by resorting to the color-change rule (2.16), along with (2.11,2.12).

Applying those rules, yields the following diagram.



This ZX-diagram contains 33 gates of which

- 14 are T-gates
- 19 are Cliffords, of which
 - 9 are CNOTs
 - 6 are XCXs
 - 4 are Hadamards

XCX gates are not commonly used for describing a quantum process. Despite this, *PyZX* utilizes this denomination quite often. This gate is simply a CNOT with an Hadamard on each side of the control qubit. This final circuit can be considered as ‘fully-optimized’. This term is used rather loosely, as there are many different definitions of what a ‘fully-optimized’ circuit is. These definitions take into consideration different metrics that can be used to optimize against, e.g. total number of gates, total number of 2-qubit gates, T-count, circuit depth, among many others. In the context of this dissertation, the expression refers to a circuit that *PyZX* is unable to simplify further, which presents, in general, a balance between T-count and the total amount of gates.

Many of the concerns prior to the optimization routines, e.g., the existence of a simpler Quantum Fourier Transform, proved to be, in some form, true. The Continuous Time Quantum Walk does not benefit hugely from the optimizations *PyZX* provides.

An interesting property does however emerge from this optimization. The rotations relative to the unitary operator (and most other rotations for that matter) are now on the third qubit. As for the reason for this, it is inconclusive.

4.2 Coined Quantum Walk

Contrasting to the Continuous Time Quantum Walk discussed in the previous section, the Coined Quantum Walk necessitates an increase in the total number of gates to yield a longer walk. Therefore, it yields the longest circuits out of all the quantum walks studied in this dissertation. Therefore, it leads to the most effective returns under the optimization techniques that *PyZX* implements.

The model in [A.1](#) is an implementation of a one-step Coined Quantum Walk.

This model possesses a total of 440 gates, of which

- 136 are T-gates
- 304 are Cliffords, of which
 - 124 are CNOTs
 - 29 are Hadamards

The gates not explicitly described above are either Z or X rotations that are multiples of $\frac{\pi}{2}$.

Since this quantum walk works in a repetitive manner, and each step used in this specific implementation adds 439 gates to the total amount of gates in the circuit, it is reasonable to think that there are some simple optimizations that could yield a simpler circuit. When ‘simple optimizations’ are mentioned it is strictly limited to non-transforming *PyZX* optimizations such as spider-simplifications ([2.11](#)) and id-simplifications ([2.12](#)).

The model in [A.2](#) is the circuit extracted after these optimizations. This model possesses a total of 417 gates, of which

- 122 are T-gates
- 295 are Cliffords, of which
 - 122 are CNOTs
 - 27 are Hadamards

This yields an optimization of around 5% (5.227%), which just by itself doesn’t mean much.

However, as soon as we utilize more elaborate (and transforming) optimization routines that *PyZX* offers a giant gain suddenly appears.

The model in [A.3](#) is the circuit extracted after these optimizations.

These optimizations bring the total amount of gates to a total of 256 gates, of which

- 64 are T-gates
- 192 are Cliffords, of which
 - 74 are CNOTs
 - 19 are CZs
 - 114 are Hadamards

Much like the Continuous Time Quantum Walk case, a lot of Hadamard gates are introduced to perform the optimization routine. These gates persist in the circuit after it is extracted.

This optimization yields a gain of around 41%(41.18%) which is closer to what *PyZX* may achieve, as per [Kissinger and van de Wetering \(2020a\)](#). This is already an incredible result, as this algorithm is already one of the most promising algorithms in the **NISQ** era. The result further cements this reputation by increasing its resilience to decoherence and noisy devices.

Even so, it is possible to go even further and optimize the Hadamard gates by applying color-changes (2.16), spider-fusions (2.11) and id-simplifications (2.12).

The resulting ZX-diagram of the circuit can be found in A.4. This circuit has a total of 158 gates, of which

- 64 are T-gates
- 94 are Cliffords, of which
 - 54 are CNOTs
 - 1 is CZ
 - 23 are XCXs
 - 12 are Hadamards

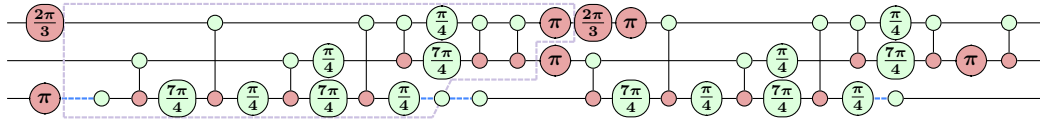
The extracted circuit has a gain of 64%(64.09%) over the original implementation.

However, as a consequence of these transforming optimizations, the resulting circuit has no resemblance to the original one. Actually we have a much smaller circuit, but it seems to amount to a set of random gates.

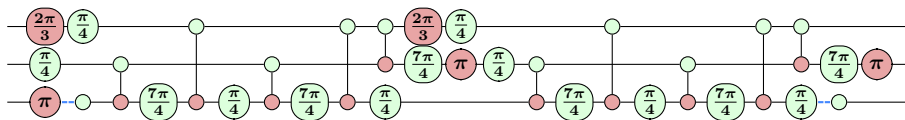
4.3 Staggered Quantum Walk

4.3.1 Optimization

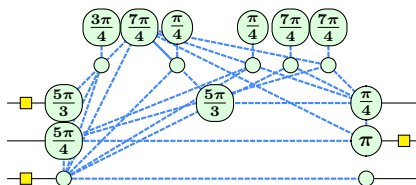
Consider the following ZX-diagram which highlights the expanded the increment operator.



Some simple optimizations can be considered to deal with the CNOT gate at the end of the expansion of the Toffoli gate and the CNOT belonging to the increment operator, and similarly, but for a X-spider between the CNOT targets, in the decrement operator. Moreover, one may cancel the two X-spiders with phase π in the first qubit between the increment and decrement layers, and the two consecutive Hadamards in the last qubit. The result is the following ZX-diagram,



These optimizations slightly reduced the number of Clifford gates in the diagram. More advanced techniques, described in [Kissinger and van de Wetering \(2020a\)](#) and directly implemented in *PyZX* as the `full_reduce` method, may reduce the circuit T-count in about 50% [Kissinger and van de Wetering \(2020a\)](#). Although this is not the case for our small example, when we start applying such simplifications to staggered models with larger amounts of steps the T-count reduction can reach approximately 60-70%. Back to the example, this simplification yields



This diagram no longer resembles a circuit, making difficult a comparison with the original one. The circuit extracted [de Beaudrap et al. \(2022\)](#) by *PyZX* has more gates than the one obtained from the simple optimizations mentioned above, although the T-count is indeed smaller.

In fact, the `full_reduce` method introduces several additional Hadamard gates to make all the edges Hadamard-edges, and color-change all the X -spiders. The subsequent circuit extraction 'preserves' the nature of the *graph-like ZX*-diagram. As such there are a lot of Hadamard gates and Z -spiders that could cancel-out or color-change. Following the extraction with a small set of simplifications, basically resorting to fusion rewrite rules followed by color-changes, we get a much smaller circuit.

This fully-simplified circuit now surpasses the original circuit in both the total amount of gates and T-count, but not the one obtained with the simple optimizations above. Although the reduction of both these metrics are not that significant in this example, when applying the same techniques in models with a greater number of steps reductions in the number of gates and T-count become quite clear. The following tables show, respectively, the total number of gates and the T-count value induced by the different optimization procedures used.

	Number of steps in the staggered quantum walk:			
Optimizations used:	1	2	4	8
None	39	77	153	305
Simple	31	59	115	227
Full-reduce + fusion/id/to_rg	37	47	72	118

Table 1: Number of total gates in the circuit in relation to the simplification routines used

	Number of steps in the staggered quantum walk:			
Optimizations used:	1	2	4	8
None	16	32	64	128
Simple	16	32	64	128
Full-reduce + fusion/id/to_rg	10	16	28	52

Table 2: Number of T gates in the circuit in relation to the simplification routines used

4.3.2 Alternative Evolution Operator

When analyzing the ZX diagram for long staggered quantum walk (i.e. with more than 5 steps) a pattern starts to emerge, repeating itself as many times as the number of steps in the quantum walk. It seems able to represent, or at least to approximate, both the increment and decrement layers of the evolution operator. Its ZX representation is

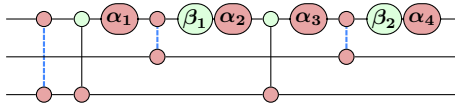


Figure 3: The alternative evolution operator.

where $\alpha_n = \pm \frac{\pi}{4}$ and $\beta_n = \frac{2\pi}{3} + m\pi$, with $m = 0$ or $m = 1$.

There is also a slight variation of this operator, where a CNOT gate between the first and last qubit appears right after the β_1 Z-spider.

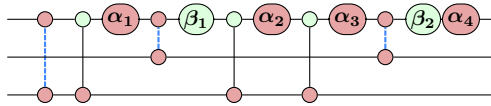


Figure 4: The variation of the alternative evolution operator.

This diagram does not fully capture the staggered model we started with, but, once suitably enveloped, it captures the exact same tensor as the original circuit. The set of gates to be placed as an envelop, in the beginning and the end of the diagram, does not exhibit a specific structure, but for 4 rotations combined with a seemingly random arrangement of other phase-less gates.

This construction appeared when optimizing the 3 qubit staggered quantum walk. However, it can be generalized for an n qubit implementation, yielding the following operator, in the form of a ZX-diagram:

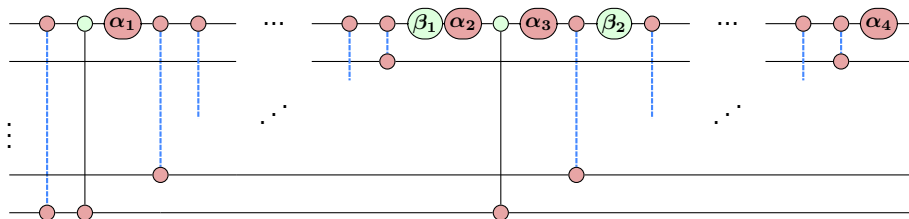


Figure 5: The generalization of the alternative evolution operator.

This operator is defined by entangling the first and last qubits and placing them in superposition. This is achieved by the XCX-gate (which is a CNOT with a Hadamard on both sides of the control), followed by a CNOT gate between the first and last qubits and a rotation over the X-axis on the first one. Then follows a ladder of XCX-gates starting with the first qubit and the $n - 1^{th}$ qubits and descending all the way to the second one. This is followed by a rotation over the Z-axis along with a X-axis rotation. Then a CNOT is applied to the first and last qubits, followed by a X-axis rotation and a XCX-gate over the first qubit and

the $n - 1^{th}$ one. Finally, there is a second Z-axis rotation followed by the same ladder of XCX-gates, now appearing between the first and the $n - 2^{th}$ qubits and descending all the way to the second. And, finally, a last X-axis rotation.

The rationale behind this operator is easy to explain: it creates a uniform distribution over a certain number of states, applies a rotation that makes some states more likely than others and then spreads these probabilities over the remaining states using CNOT gates. This also explains why the pattern only shows up in staggered quantum walks over a certain length. The classical evolution operator resorting to the increment and decrement layers, needs to be repeated a number of times to be able to spread the probability distributions over the whole state space. This is what this version does on the first layer. Thus, it cannot approximate staggered quantum walks with a small number of steps as well as it can for longer ones.

In any case, this alternative implementation has a number of advantages. First and foremost it reduces the total amount of gates needed to represent the evolution of the quantum walk. With the number of qubits increasing so does the cost of the increment and decrement layers, as a n qubit staggered quantum walk needs to implement MCX gates with $n - 1$ controls. Then, such MCX gates need to be expanded into their basic gates representation, as [Slepoy \(2006\)](#) stated. The alternative operator uses at most gates controlled by at most 1 qubit. Moreover, to go from a n to a $n + 1$ qubit quantum walk, all that needs to be done is to add two more XCX-gates, one to each ladder of XCX-gates. Such is not the case of the evolution operator based on increment and decrement layers.

In general, this makes the alternative operator much more efficient with respect to the total number of gates used, leading to lower depth and, therefore, potentially less error-prone circuits.

As mentioned above, just by itself this operator is able of approximating the evolution of a staggered quantum walk. Although the approximation is not perfect it can yield results which are quite similar to the ones obtained with the original implementation of the staggered model, as shown in the graph below.

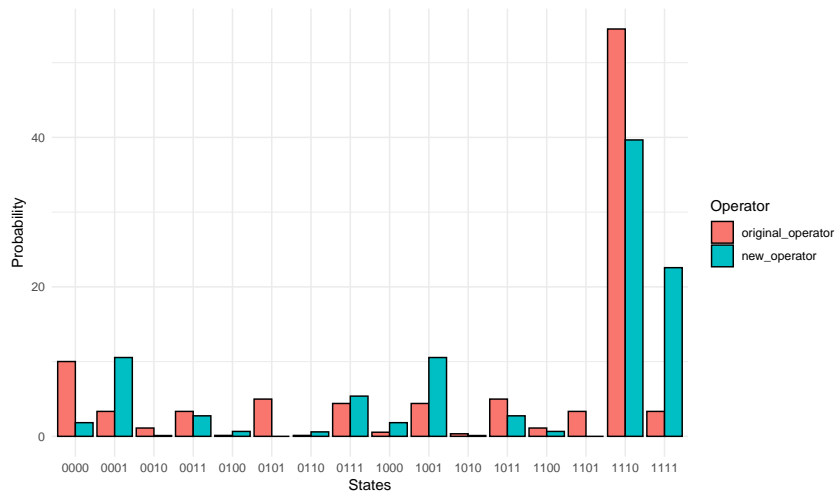


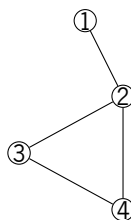
Figure 6: The state space probabilities for a 4-qubit quantum walk with 4 steps using the classical operator (red) and the alternative one (blue).

One particular advantage of this alternative evolution operator is that it can work quite well on a quantum processor with limited connectivity. This is due to the fact that all the qubits used in the staggered quantum walk only need to be strongly connected to the first qubit, thus minimizing possible errors occurring from having to operate on two qubits with poor connectivity.

However, a number of challenges remain, requiring further investigation. These concern the most suitable choice of parameters for α_n and β_n , as well as whether and how they depend on the number of qubits used in a particular staggered walk. Actually, when optimizing the 4 qubit implementation of this circuit the resulting parameters did not seem to follow any regular pattern.

4.4 Quantum Approximate Optimization Algorithm

Using the model from 3.31 we are able to solve the MaxCut problem for the following graph.



For a small graph like this the MaxCut problem is quite trivial and can be solved directly, the solutions being:

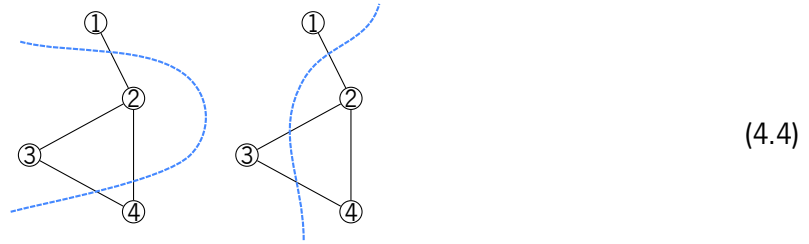


Figure 7: The two equivalent solutions to the MaxCut problem.

Encoding the solutions into *bitstrings*, with one bit for each node of the graph, we get $s_1 = 0110$ and $s_2 = 0101$. The solutions $s_1 = 1001$ and $s_2 = 1010$ are also valid as they are the dual to the ones above. Therefore, when the circuit is executed those should appear the most probable outputs, and as the number layers of the **QAOA** tend to infinity, the optimal solution will be obtained with certainty. However, for such a simple example a couple of layers are sufficient.

Since the **QAOA** is a hybrid loop, a part of the algorithm runs on a classical optimizer. This is where the parameters for the phase-separation and mixer Hamiltonians are determined.

After optimizing the parameters the circuit can be finally executed in order to yield the solution.

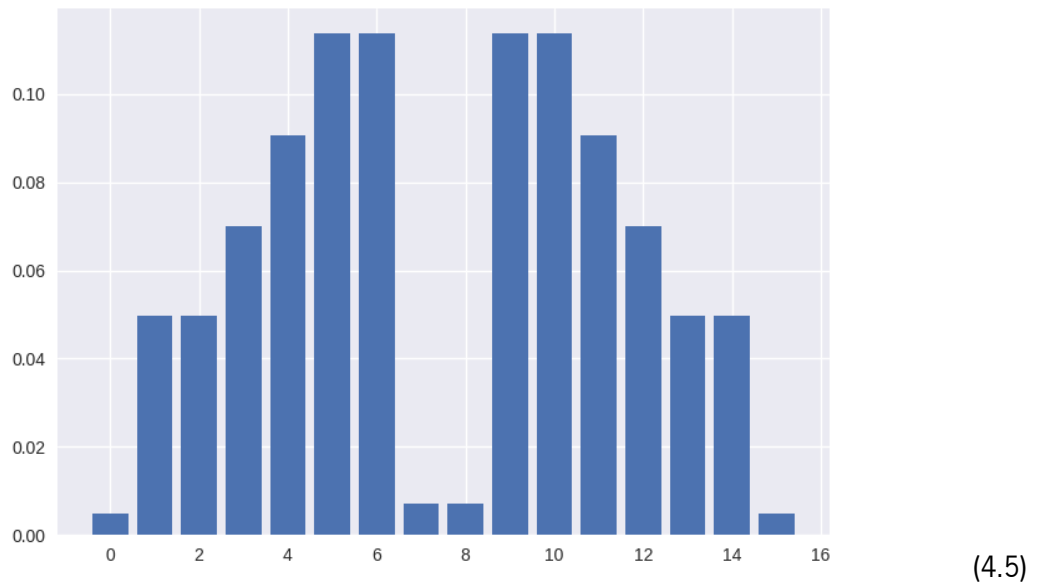


Figure 8: The probability distribution of the solution for the MaxCut problem.

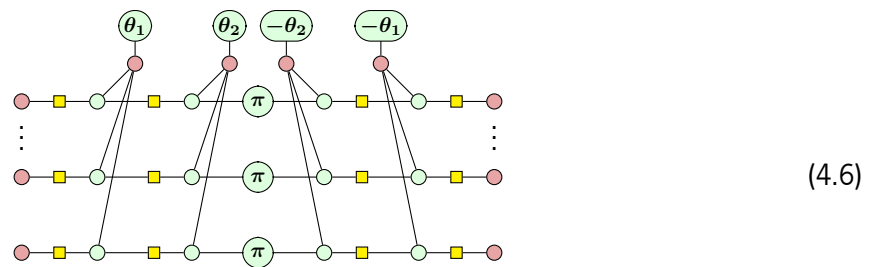
As can be in figure 8 the states $|5\rangle$ and $|6\rangle$ (along with $|9\rangle$, $|10\rangle$) are the most probable. After converting those states to *bitstrings*, it yields $s_1 = 0110$ and $s_2 = 0101$, which coincides with the proposed MaxCut in 4.4.

Due to the way the circuit for the **QAOA** was constructed 3.2.2, there isn't much room for further optimization. The circuit takes such a simple and almost irreducible form that any optimization routine from *PyZX* yields a longer circuit.

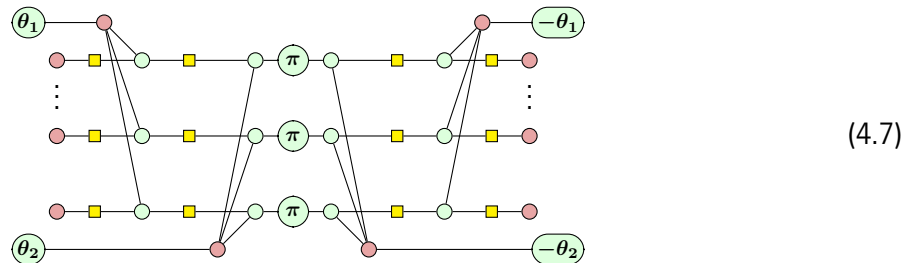
4.5 Hardware Efficient Ansatz

4.5.1 Example Ansatz

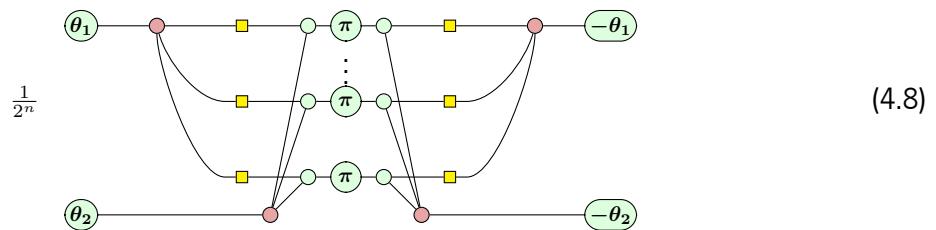
The first ansatz where this analysis was conducted was the one introduced in 3.40, as it is quite compact.



Bending the diagram, it yields:



Applying color-change (2.16) and spider-fusion (2.11) on the inputs and outputs, leads to:



Pushing the Hadamard gates through the second phase-gadget changes the color of the observable along with the legs of the phase-gadget itself. Thus,

(4.9)

One can now move the observable out of the center of the diagram, resulting in the $n\pi$ X-spider next to the parameter θ_1 , where n is the number of qubits present in the ansatz:

(4.10)

Applying fusion for pink-spiders (3.38), yields

(4.11)

and consequently

(4.12)

This is where this analysis starts to branch out, depending on whether n (the number of qubits) is even or odd. The Hopf rule (2.18) leads to:

(4.13)

On the even branch a lot of scalars start to emerge, which can be simplified as follows:

$$\circ = 2 \quad \bullet = 1 \quad \textcircled{\pi} = \textcircled{\pi} = 0 \quad \bullet\textcircled{\alpha} = 1 \quad \textcircled{\pi}\textcircled{\alpha} = e^{i\alpha} \quad (4.14)$$

Thus,

$$\left\{ \begin{array}{l} 0 \quad \text{if } n \text{ even} \\ \frac{1}{4} \begin{array}{l} \textcircled{\theta_1} \textcircled{n\pi} \textcircled{-\theta_1} \\ \textcircled{\theta_2} \textcircled{\square} \textcircled{\square} \textcircled{-\theta_2} \end{array} \quad \text{if } n \text{ odd} \end{array} \right. \quad (4.15)$$

This means that if n is even, the expectation value will be 0, and in consequence the partial derivative and the variance for both parameters.

Utilizing the theorem 3.35 we can calculate the variance of the parameters.

$$\text{Var}\left(\frac{\partial \langle H \rangle}{\partial \theta_1}\right) = \frac{1}{16} \quad (4.16)$$

Using the following Lemma from the work of Koch (2022) one eliminates 2 edges, along with a π X-spider

$$\begin{array}{c} \textcircled{\pi} \\ \diagup \quad \diagdown \\ \textcircled{x\pi} \quad \textcircled{y\pi} \\ \diagdown \quad \diagup \\ \textcircled{\alpha} \end{array} = \begin{array}{c} \textcircled{\pi} \\ \diagup \quad \diagdown \\ \textcircled{x\pi} \quad \textcircled{y\pi} \\ \diagdown \quad \diagup \\ \textcircled{\alpha + \pi} \end{array} \quad \begin{array}{c} \textcircled{\pi} \\ \diagup \quad \diagdown \\ \textcircled{\alpha} \quad \textcircled{\beta} \\ \diagdown \quad \diagup \\ \textcircled{x\pi} \end{array} = \begin{array}{c} \textcircled{\pi} \\ \diagup \quad \diagdown \\ \textcircled{\alpha} \quad \textcircled{\beta} \\ \diagdown \quad \diagup \\ \textcircled{x\pi + \pi} \end{array} \quad (4.17)$$

This coupled with the identity removal rule (2.12), yields

$$\frac{1}{16} \text{ (Diagram with } \pi \text{ nodes and squares) } \quad (4.18)$$

Simplifying the inner loop with the spider fusion rule (2.11) and identity removal (2.12) results in

$$\frac{1}{16} \text{ (Diagram with self-loop) } \quad (4.19)$$

To simplify the self-loop an identity must be introduced in order to utilize the Hopf rule (2.18) and thus break up the self-loop. This procedure however introduces a scalar. Applying the color-change and id-removal rule, brings

$$\frac{1}{16} \text{ (Diagram with self-loop and central node) } \quad (4.20)$$

In order to break apart the diagram, the spider fusion along with the Hopf rule must be used leading to the following diagram

$$\frac{1}{16} \text{ (Diagram with three nodes) } \quad (4.21)$$

Resorting to the triangle related rules introduced in the work of Wang et al. (2024), followed by spider fusion and scalar representation as above, yields

$$\frac{1}{16} \left[\begin{array}{c} \circ \\ \circ \\ \circ \\ \circ \end{array} \right] = \frac{1}{16} \left[\begin{array}{c} \circ \\ \circ \\ \circ \\ \circ \end{array} \right] = \frac{1}{2} \quad (4.22)$$

This result proves that for this example ansatz the $\text{Var}\left(\frac{\partial \langle H \rangle}{\partial \theta_1}\right) = \frac{1}{2}$. And as such it does not contain barren plateaus. This is most likely to the lack of expressiveness this ansatz possesses as all the qubits are rotated along the same parameters.

The same analysis could be conducted to the parameter θ_2 , however its variance will be 0. This is due to the fact that θ_2 is in a phase-gadget along the Z-axis and, as such, when measuring with an observable $Z^{\otimes n}$ it is able to commute through the observable and cancel out with its counterpart. Thus the parameter θ_2 does not contribute to the expectation value.

4.5.2 Sim Inspired Ansatz

This ansatz is a slight variation of the Sim 2 ansatz presented in the work of [Sim et al. \(2019\)](#). Instead of a ladder of CNOTs this variation has a pyramid of CNOTs that converges in the center qubits. This, this architecture imposes a restriction in the number of qubits utilized in the ansatz, which must conform to a power of 2.

$$\quad (4.23)$$

In order to conduct the analysis of this ansatz the observable will have to commute with the layer of CNOTs on the left. The following ZX-diagram is an abstraction of the top half of the CNOT pyramid with the observable.

$$\quad (4.24)$$

Applying the the pi-commutation rule (2.14) to the observable's π X-spiders through the control of the CNOT gates, yields the following diagram:

The pi-commutation rule (2.14) introduces a π X-spider on the left of the control of the CNOT and a π X-spider on the target of the CNOT. This means that for a qubit between the second and the $(2^{n-1} + 1)th$ all the π X-spiders will cancel each other out through spider fusion (2.11), resulting in the following ZX-diagram

Since the X-spiders from the observable do not commute with the previous qubits due to the orientation of the CNOT gates the same proof from 4.25 can be used for the qubits between the $(2^{n-1} + 2)th$ and $(2^{n-1} + 2^{n-2} + 1)th$. Actually, this proof can be used recursively for the remaining qubits. This means that the observable commuting with the CNOT pyramid will yield an observable with a π X-spider in the following positions

$$1, \quad 2^{n-1} + 2, \quad 2^{n-1} + 2^{n-2} + 2, \quad 2^{n-1} + 2^{n-2} + 2^{n-3} + 2, \quad \dots, \quad 2^n \quad (4.27)$$

Applying this process to the ansatz yields the following ZX-diagram

This way the pyramids of CNOTs can cancel out through spider fusion (2.11), Hopf (2.18) and the id-removal (2.12) rules, resulting in

This ansatz is equivalent to the Sim 1 ansatz with a restriction on the observable used, which must be in the specific qubits described in 4.27.

Koch (2022) conducted this same analysis in the Sim 1 ansatz and concluded that the variance for all parameters is either 0 or

$$\frac{1}{2^{2h_X+h_Y+h_Z}} \quad (4.30)$$

where h_P is is is number of occurrences of the Pauli $P \in \{X, Y, Z\}$ in the observable.

Note that this specific instance only contains observables along the X-axis and the number of observables has an upper bound of $\log_2(n)$, where n is the total number of qubits, which prevents the variance from vanishing exponentially.

An analogous proof of the commutation (4.25) of the observable with the pyramid of CNOTs could be constructed for an observable $Z^{\otimes n}$, yielding an observable with a π Z-spider on the following qubits:

$$2^n, \quad 2^{n-1} - 1, \quad 2^{n-2} - 1, \quad 2^{n-3} - 1, \quad \dots, \quad 1 \quad (4.31)$$

The amount of measurements in this case is also limited with an upper bound of $\log_2(n)$, where n is the total number of qubits, preventing again the exponential vanishing of the gradients.

Thus, the Sim Inspired Ansatz does not have barren plateaus for either σ_X or σ_Z measurements, which is most likely assured because of the ‘local’ measurements that this ansatz guarantees.

Chapter 5

Conclusions and Future Work

5.0.1 Discussion and Contributions

This dissertation aimed at exploring ZX in modelling and optimizing quantum algorithms, namely in the area of quantum data science.

Chapter 1 introduced the context and motivation of the dissertation along with its objectives. This was followed by an overview of the state of the art of both the ZX-calculus, Quantum Computing and Quantum Machine Learning and finally the Barren Plateau phenomenon.

Chapter 2 provides a brief overview of the mathematical foundation for the work presented in the dissertation. In particular, it explored a small set of use cases of the ZX-calculus and derivations. It was also detailed how the ZX-calculus is able, resorting to tools like *PyZX*, to reduce considerably the amount of T-gates in circuits.

Chapter 3 had the purpose of presenting the diverse set of algorithms that were object of study in the form of ZX-diagrams. These would be later exposed to diverse techniques within the framework of the ZX-calculus.

The principal contribution of this dissertation resides in Chapter 4, A first set discussing possible circuit optimizations deals with the implementation of three different types of quantum walks. The first of these, the Continuous Time Quantum Walk, yielded the least interesting results. This is mainly due to the nature of the protocol itself. The following two, as discrete time variants, were much more amenable to the optimization routines employed, making these more resilient to decoherence and noise originated from the quantum hardware. For the Staggered Quantum Walk an interesting alternative evolution operator, was proposed, as documented in the paper ‘Reconfiguring staggered quantum walks with ZX’ ([Jardim et al. \(2024\)](#)) written in conjunction with Jaime Santos and Luís Soares Barbosa. This article was submitted and accepted to the International Workshop on Reconfigurable Transition Systems: Semantics, Logics and Applications – *ReacTS’24* – and will be presented in the following 5th of November 2024. Comparatively

the results obtained on the Quantum Approximate Optimization Algorithm utilizing the ZX-calculus were quite lackluster, at least in terms of circuit optimization.

We have also addressed some applications in the context of Measurement Based Quantum Computing (**MBQC**), which, due to their preliminary character weren't included in the main part of the dissertation. They are nonetheless discussed below in Section 5.0.3. The chapter closes with an analysis of the barren plateau phenomenon in two different ansatzs. The first, an introductory example, was proven to not have barren plateaus. The second was a variation of the ansatzs present in the work of Sim et al. (2019). The reason for this choice was the fact that a combination of the work of Koch (2022) had already conducted similar analysis on the ansatzs denoted *Sim 1*, *Sim 2* and *Sim 9*. Other sorts of ansatz include parameterized controlled gates or parameterized R_Y gates, which would make this type of analysis unfeasible, due to the complexity of representing these gates as ZX-diagrams, in the first case, or a lack of research documented in the literature, in the second. This prompted the creation of an ansatz that resembled the nature of Sim et al. (2019)'s work and allowed for this kind of analysis. This ansatz was proven to not have barren plateaus through a reduction to the ansatz denoted *Sim 1*.

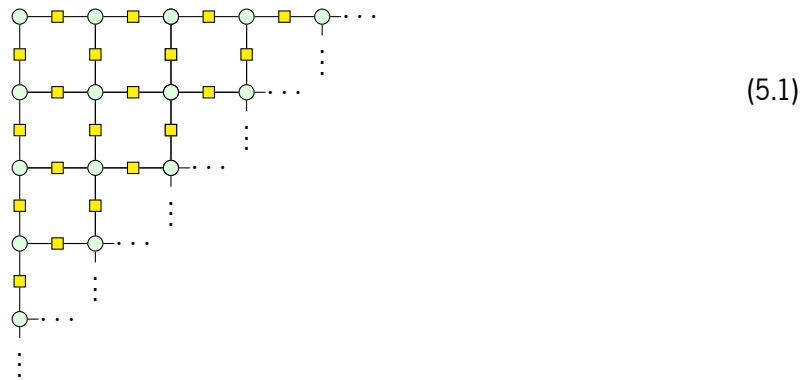
5.0.2 Future Work

This dissertation calls for further research in several directions. Therefore the following items are suggested as future work!

1. Characterize precisely the gates that envelop the alternative evolution operator in the Staggered Quantum Walk, improving the approximation already obtained.
2. Formalize the parameters α_n and β_n used in the alternative evolution operator. This will make the alternative operator more reliable, as, in its current form, it does not guarantee that the next step of a given walk can be approximated.
3. Expand the work presented in Section 5.0.3 to accommodate different classes of problems besides the **QUBO**.
4. Convert Quantum Machine Learning ansatzs into **MBQC** protocols, so that these protocols could be executed in photonic hardware.
5. Integrate the work of Yeung (2020) within the framework for analytical barren plateau analysis to better accommodate ansatzs that include parameterized R_y rotations.

5.0.3 Measurement Based Quantum Computing

As explained above, the following section discusses some, still preliminary, work in the **MBQC** domain. Measurement Based Quantum Computing is a rather novel, comparatively to unitary evolution quantum computing, method of performing quantum computations. First introduced by [Raussendorf and Briegel \(2001\)](#), it entirely relies on single-qubit measurements over a heavily entangled initial state, usually denoted by the cluster-state. These measurements are an inherently destructive and random process. However, they are correlated and can dictate the program evolution. The processing of the outcomes of such measurements in a specific order and in a specific basis, makes it to be propagated through the cluster. This cluster state is created by entangling pairs of qubits in a grid-like fashion with CZ-gates. This can be represented with the following ZX-diagram.



One can interpret the number of qubits in each column as the number of qubits available to perform computations and assume that information flows from the left to the right.

This setting, in conjunction with an universal set of gates, such as $\{CZ, R_X(\theta), R_Z(\phi)\}$ for all θ, ϕ is denoted the *one way quantum computer*, whose concept relies on the following facts, that are elaborated further in the work of [Jozsa \(2005\)](#)

- Any permutation of gates can be implemented exclusively as 1-qubit measurements, as long as the two dimensional cluster state is large enough;
- All the measurements are in the basis $M_z = \{|0\rangle, |1\rangle\}$ and $M(\theta) = \{|0\rangle \pm e^{i\theta} |1\rangle\}$ for some θ ;
- Measurements are always labelled 0 or 1 and uniformly random;
- Quantum gates are implemented only up to Pauli corrections $X^a Z^b$, where a and b are dependent on the measurement outcome.

$$\text{---} \beta \text{---} = \begin{array}{c} \text{---} \text{---} m_v \pi \\ | \\ \text{---} \text{---} (-1)^{m_v} \beta \text{---} m'_v \pi \\ | \\ \text{---} \text{---} m'_v \pi \text{---} m_v \pi \end{array} \quad (5.5)$$

This time there is the need for two ancilla qubits. As the input qubit is measured its information is transferred to the second ancilla qubit where it is corrected for the measurements performed in the input and the first ancilla qubit.

Let us complete a derivation for an arbitrary Z-rotation yielding

$$\text{---} \gamma \text{---} = \begin{array}{c} \text{---} \text{---} \gamma \text{---} m \pi \\ | \\ \text{---} \text{---} m \pi \end{array} \quad (5.6)$$

Thus, one is able to construct an implementation of the **QAOA** protocol in the paradigm of Measurement Based Quantum Computing.

Consider the following example:

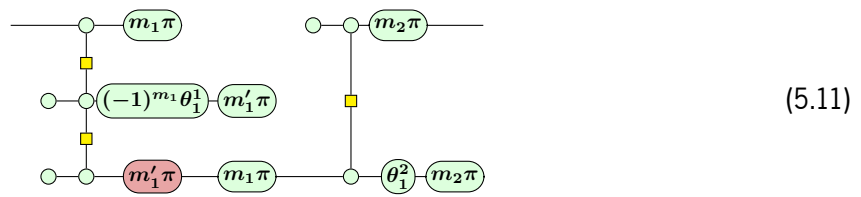
$$\begin{array}{c} \text{---} \text{---} \beta \text{---} \pi \text{---} -\beta \text{---} \\ | \quad | \quad | \\ \text{---} \text{---} \gamma \text{---} \beta \text{---} \pi \text{---} -\beta \text{---} \text{---} -\gamma \text{---} \\ | \quad | \quad | \\ \text{---} \text{---} \gamma \text{---} \beta \text{---} -\beta \text{---} \text{---} -\gamma \text{---} \end{array} \quad (5.7)$$

This is an implementation for the MaxCut problem in a K_3 graph with an observable on the first and second qubits, as any combination of 2 nodes is a solution to the MaxCut problem.

Utilizing π -commutation (2.14), spider fusion (2.11) and color change (2.16) rules yields the following ZX-diagram.

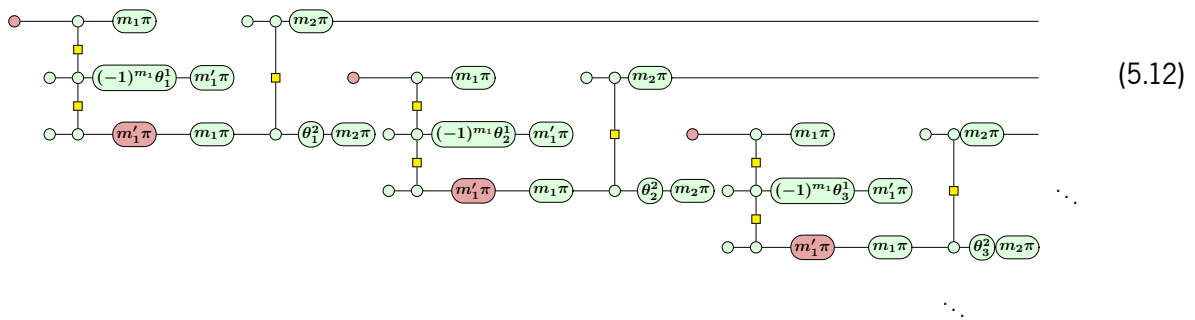
$$\begin{array}{c} \text{---} \text{---} \pi \text{---} \text{---} 2\beta \text{---} \\ | \quad | \\ \text{---} \text{---} \pi \text{---} \text{---} 2\beta \text{---} \text{---} -\gamma \text{---} \\ | \quad | \\ \text{---} \text{---} \text{---} -\gamma \text{---} \end{array} \quad (5.8)$$

This circuit now allows us to implement the equivalence between the circuit model and the **MBQC** paradigm described above, resulting in



Note that the construction that represents the R_Z rotation was simply inverted to assure the output qubit is in the same row as the input.

This construction can then be repeated in a ladder like fashion to yield the complete ansatz



This example illustrates how Quantum Machine Learning protocols run on a completely different paradigm and a plentitude of different quantum processors.

References

- Y. Aharonov, L. Davidovich, and N. Zagury. Quantum random walks. *Phys. Rev. A*, 48:1687–1690, Aug 1993. doi: 10.1103/PhysRevA.48.1687. URL <https://link.aps.org/doi/10.1103/PhysRevA.48.1687>.
- Miriam Backens and Aleks Kissinger. Zh: A complete graphical calculus for quantum computations involving classical non-linearity. *Electronic Proceedings in Theoretical Computer Science*, 287:23–42, January 2019. ISSN 2075-2180. doi: 10.4204/eptcs.287.2. URL <http://dx.doi.org/10.4204/EPTCS.287.2>.
- Paul Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics*, 22:563–591, 1980. URL <https://api.semanticscholar.org/CorpusID:122949592>.
- Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, September 2017. ISSN 1476-4687. doi: 10.1038/nature23474. URL <http://dx.doi.org/10.1038/nature23474>.
- M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12(1), March 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-21728-w. URL <http://dx.doi.org/10.1038/s41467-021-21728-w>.
- M. Cerezo, Martin Larocca, Diego García-Martín, N. L. Diaz, Paolo Braccia, Enrico Fontana, Manuel S. Rudolph, Pablo Bermejo, Aroosa Ijaz, Supanut Thanasilp, Eric R. Anschuetz, and Zoë Holmes. Does provable absence of barren plateaus imply classical simulability? or, why we need to rethink variational quantum computing, 2024. URL <https://arxiv.org/abs/2312.09121>.
- Andrew M. Childs and Jeffrey Goldstone. Spatial search by quantum walk. *Physical Review A*, 70(2), August 2004. ISSN 1094-1622. doi: 10.1103/physreva.70.022314. URL <http://dx.doi.org/10.1103/PhysRevA.70.022314>.

- Bob Coecke and Aleks Kissinger. The compositional structure of multipartite quantum entanglement, 2010. URL <https://arxiv.org/abs/1002.2540>.
- Niel de Beaudrap, Aleks Kissinger, and John van de Wetering. Circuit extraction for zx-diagrams can be #p-hard. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPICS.ICALP.2022.119. URL <https://drops.dagstuhl.de/entities/document/10.4230/LIPICS.ICALP.2022.119>.
- David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400:117 – 97, 1985. URL <https://api.semanticscholar.org/CorpusID:1438116>.
- David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439:553 – 558, 1992. URL <https://api.semanticscholar.org/CorpusID:121702767>.
- Ross Duncan, Aleks Kissinger, Simon Perdrix, and John van de Wetering. Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus. *Quantum*, 4:279, June 2020. ISSN 2521-327X. doi: 10.22331/q-2020-06-04-279. URL <https://doi.org/10.22331/q-2020-06-04-279>.
- Bryan Eastin and Emanuel Knill. Restrictions on transversal encoded quantum gate sets. *Physical Review Letters*, 102(11), March 2009. ISSN 1079-7114. doi: 10.1103/physrevlett.102.110502. URL <http://dx.doi.org/10.1103/PhysRevLett.102.110502>.
- Andrew Fagan and Ross Duncan. Optimising clifford circuits with quantomatic. *Electronic Proceedings in Theoretical Computer Science*, 287:85–105, January 2019. ISSN 2075-2180. doi: 10.4204/eptcs.287.5. URL <http://dx.doi.org/10.4204/EPTCS.287.5>.
- R. P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6): 467–488, 1982.
- Lov K. Grover. A fast quantum mechanical algorithm for database search, 1996.
- Stuart Hadfield. On the representation of boolean and real functions as hamiltonians for quantum computing. *ACM Transactions on Quantum Computing*, 2(4):1–21, December 2021. ISSN 2643-6817. doi: 10.1145/3478519. URL <http://dx.doi.org/10.1145/3478519>.

- Bruno Jardim, Jaime Santos, and Luís S. Barbosa. Reconfiguring staggered quantum walks with ZX. *ReactS'24, Springer Lecture Notes in Computer Science, 2024 (to appear)*, November 2024.
- Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Completeness of the ZX-Calculus. *Logical Methods in Computer Science*, Volume 16, Issue 2, June 2020. doi: 10.23638/LMCS-16(2:11)2020. URL <https://lmcs.episciences.org/6532>.
- Richard Jozsa. An introduction to measurement based quantum computation, 2005. URL <https://arxiv.org/abs/quant-ph/0508124>.
- Aleks Kissinger and John van de Wetering. Reducing the number of non-clifford gates in quantum circuits. *Physical Review A*, 102(2), August 2020a. ISSN 2469-9934. doi: 10.1103/physreva.102.022406. URL <http://dx.doi.org/10.1103/PhysRevA.102.022406>.
- Aleks Kissinger and John van de Wetering. Pyzx: Large scale automated diagrammatic reasoning, 2020b.
- Mark Koch. Quantum machine learning using the zxw-calculus, 2022. URL <https://arxiv.org/abs/2210.11523>.
- Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1), November 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-07090-4. URL <http://dx.doi.org/10.1038/s41467-018-07090-4>.
- Renato Portugal. Establishing the equivalence between szegedy's and coined quantum walks using the staggered model. *Quantum Information Processing*, 15(4):1387–1409, January 2016. ISSN 1573-1332. doi: 10.1007/s11128-015-1230-7. URL <http://dx.doi.org/10.1007/s11128-015-1230-7>.
- Renato Portugal, Raqueline A. M. Santos, Tharso D. Fernandes, and Demerson N. Gonçalves. The staggered quantum walk model. *Quantum Inf. Process.*, 15(1):85–101, 2016. doi: 10.1007/S11128-015-1149-Z. URL <https://doi.org/10.1007/s11128-015-1149-z>.
- John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, August 2018. ISSN 2521-327X. doi: 10.22331/q-2018-08-06-79. URL <http://dx.doi.org/10.22331/q-2018-08-06-79>.

- Michael Ragone, Bojko N. Bakalov, Frédéric Sauvage, Alexander F. Kemper, Carlos Ortiz Marrero, Martin Larocca, and M. Cerezo. A lie algebraic theory of barren plateaus for deep parameterized quantum circuits, 2024. URL <https://arxiv.org/abs/2309.09342>.
- Robert Raussendorf and Hans J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, 86:5188–5191, May 2001. doi: 10.1103/PhysRevLett.86.5188. URL <https://link.aps.org/doi/10.1103/PhysRevLett.86.5188>.
- Jaime Santos. Quantum random walks: simulations and physical realizations. MSc Thesis in Engineering Physics, DI, Universidade do Minho, 2021.
- P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994. doi: 10.1109/SFCS.1994.365700.
- Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019. doi: <https://doi.org/10.1002/qute.201900070>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/qute.201900070>.
- Alexander Slepoy. Quantum gate decomposition algorithms. Sandia Report SAND2006-3440, Sandia National Laboratories, 2006. URL <https://www.osti.gov/biblio/889415>.
- Tobias Stollenwerk and Stuart Hadfield. Measurement-based quantum approximate optimization, 2024. URL <https://arxiv.org/abs/2403.11514>.
- Alan Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(1):230–265, 1936. doi: 10.2307/2268810.
- Salvador Venegas-Andraca. Quantum walks: a comprehensive review. *Quantum Information Processing*, 11(5):1015–1106, July 2012. ISSN 1573-1332. doi: 10.1007/s11128-012-0432-5. URL <http://dx.doi.org/10.1007/s11128-012-0432-5>.
- Quanlong Wang, Richie Yeung, and Mark Koch. Differentiating and integrating zx diagrams with applications to quantum machine learning, 2024. URL <https://arxiv.org/abs/2201.13250>.
- Richie Yeung. Diagrammatic design and study of ansätze for quantum machine learning, 2020. URL <https://arxiv.org/abs/2011.11073>.

Chen Zhao and Xiao-Shan Gao. Analyzing the barren plateau phenomenon in training quantum neural networks with the zx-calculus. *Quantum*, 5:466, June 2021. ISSN 2521-327X. doi: 10.22331/q-2021-06-04-466. URL <http://dx.doi.org/10.22331/q-2021-06-04-466>.

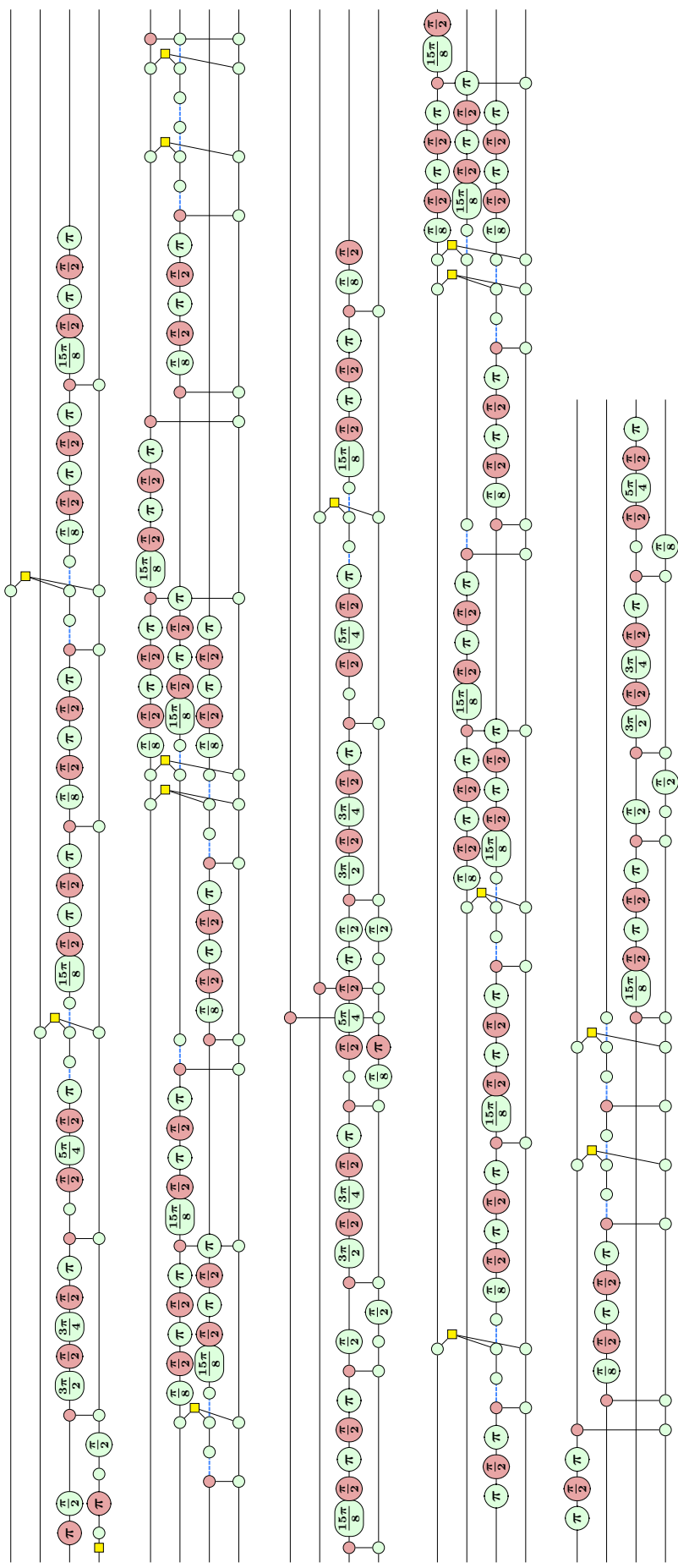
Part I

Appendices

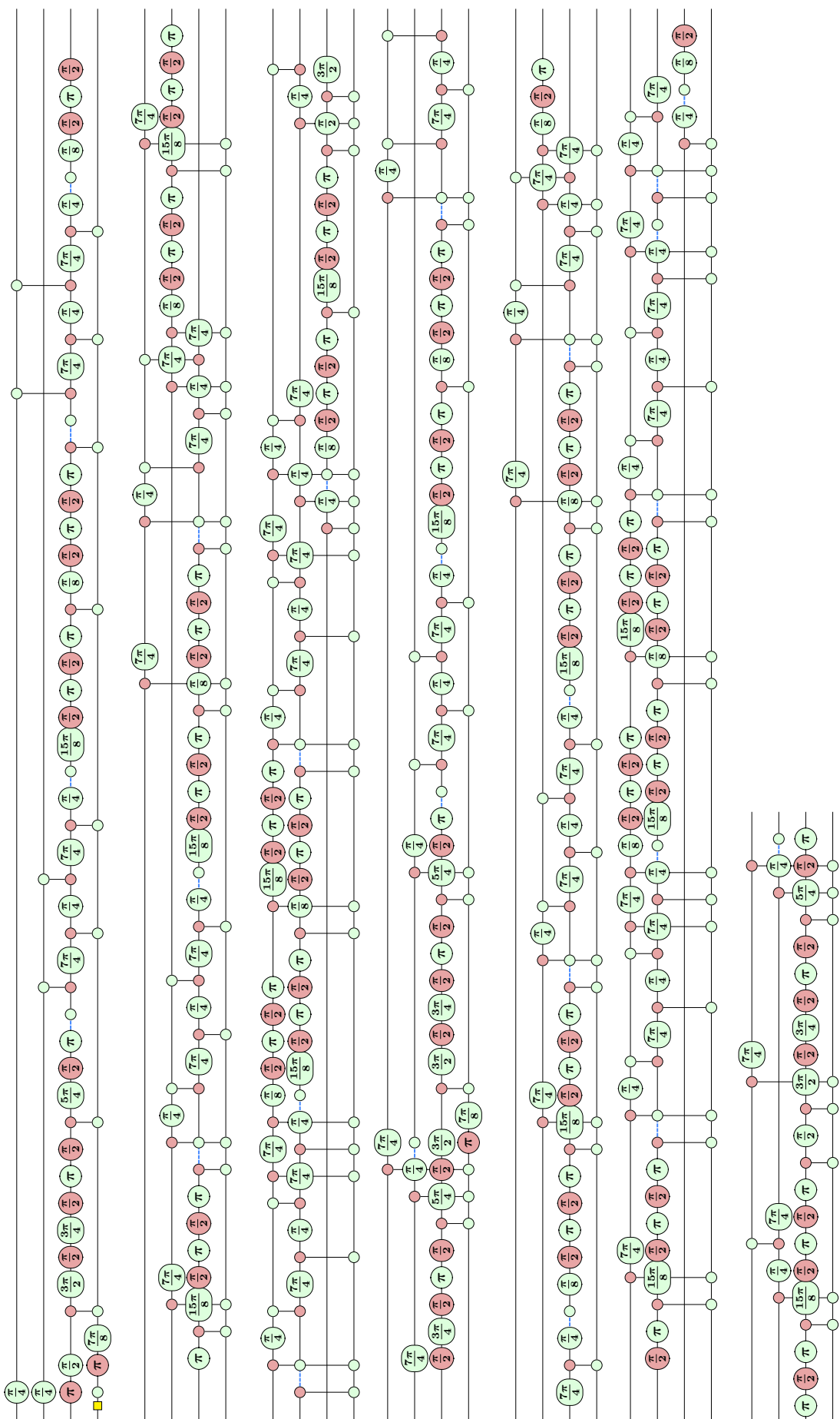
Appendix A

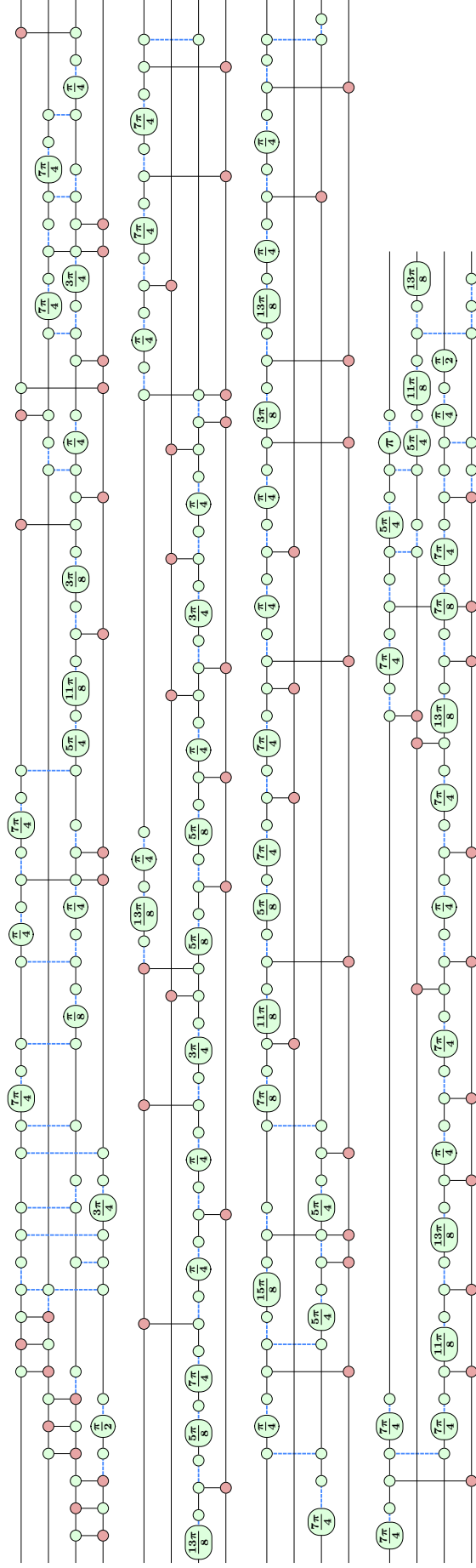
Models

The following pages are dedicated to the different ZX-diagrams that represent the Coined Quantum Walk, as they were too extensive to be detailed in the main text.

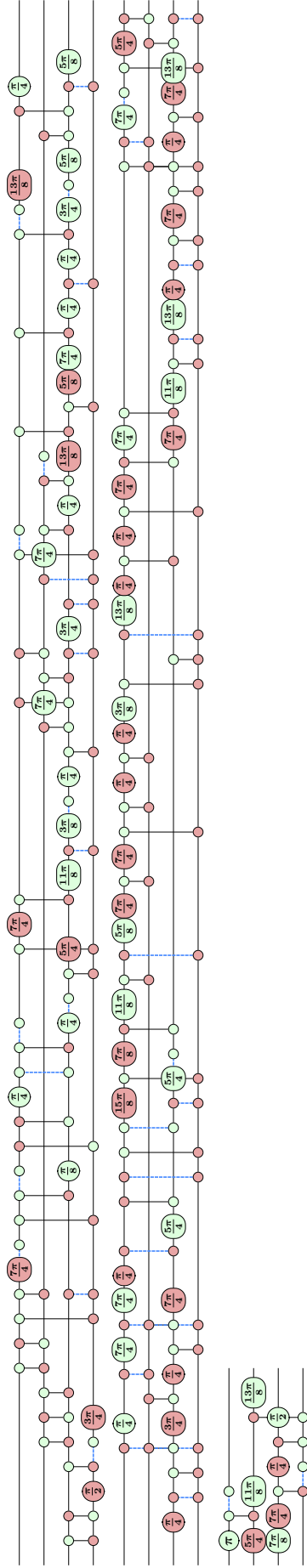


(A.1)





(A.3)



(A.4)

