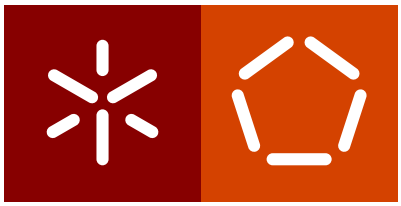


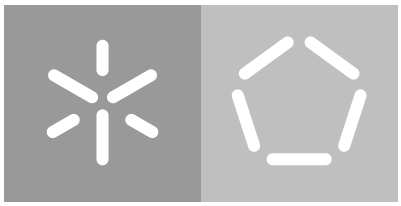
Universidade do Minho
Escola de Engenharia??
Departamento de Informática??

Jaime Santos

Staggered Quantum Walks in Qiskit??
Second Part of Title

First Part of Subtitle
Second part of Subtitle

March 2021



Universidade do Minho
Escola de Engenharia??
Departamento de Informática??

Jaime Santos

Staggered Quantum Walks in Qiskit??
Second Part of Title

First Part of Subtitle
Second part of Subtitle

Master dissertation
Master Degree in Física da Informaçao

Dissertation supervised by
Luís Barbosa
Bruno Chagas

March 2021

ACKNOWLEDGEMENTS

Write acknowledgements here

ABSTRACT

Implementation of staggered quantum walks in qiskit.

RESUMO

Pensar no que escrever aqui. Hello!

CONTENTS

1 INTRODUCTION 1
- 1.1 Brief history of quantum computing 1
- 1.2 Classical and Quantum Random Walks 1
- 1.3 State of the Art quantum walks implementations 1
- 1.4 Text overview and contributions 1

2 QUANTUM COMPUTING 2
- 2.1 Grover's Algorithm 2
  - 2.1.1 One marked element 4
  - 2.1.2 Multiple marked elements 6
  - 2.1.3 Single-Shot Grover 7

3 QUANTUM WALKS 9
- 3.1 Classical Random Walk 9
- 3.2 Coined Quantum Walk 10
- 3.3 Continuous-Time Quantum Walk 15
- 3.4 Staggered Quantum Walk 20
- 3.5 Search problems with Quantum Walks 22
  - 3.5.1 Coined 23
  - 3.5.2 Staggered 24
  - 3.5.3 Continuous 26

4 IMPLEMENTATIONS AND APPLICATIONS 30
- 4.1 Coined 30
- 4.2 Continuous 35
- 4.3 Staggered 39
- 4.4 Search Problems with Qiskit 39
  - 4.4.1 Grover 39
  - 4.4.2 Coined 40
  - 4.4.3 Continuous 40
  - 4.4.4 Staggered 40

5 DISCUSSIONS AND CONCLUSION 41
- 5.1 Conclusions 41
- 5.2 Prospect for future work 41

A SUPPORT MATERIAL 44

LIST OF FIGURES

Figure 1 Grover one marked element temp. 5
Figure 2 Grover Multiple marked temp. 6
Figure 3 Single Shot temp 7
Figure 4 Classical Walk Temp. 9
Figure 5 Probability distribution for the coined quantum walk on a line, after 100 steps, with initial condition $|\Psi(0)\rangle = |0\rangle |x = 0\rangle$ and the Hadamard coin. 13
Figure 6 Probability distribution for the coined quantum walk on a line, after 100 steps, with initial condition $|\Psi(0)\rangle = -|1\rangle |x = 0\rangle$ and the Hadamard coin. 13
Figure 7 Probability distribution for the coined quantum walk on a line, after 100 steps, with initial condition $|\Psi(0)\rangle = \frac{|0\rangle - i|1\rangle}{\sqrt{2}} |x = 0\rangle$ and the Hadamard coin. 14
Figure 8 Probability distribution for the continuous-time quantum walk on a line, at t = 100, with initial condition $|\Psi(0)\rangle = |0\rangle$ and $\gamma = \frac{1}{2\sqrt{2}}$. 17
Figure 9 Probability distribution for the continuous-time quantum walk on a line, after 100 steps, with initial condition $|\Psi(0)\rangle = |0\rangle$ and $\gamma = \frac{1}{6\sqrt{2}}$. 18
Figure 10 Temporary 18
Figure 11 Probability distribution for the continuous-time quantum walk on a line, after 100 steps, with initial condition $|\Psi(0)\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$ and $\gamma = \frac{1}{2\sqrt{2}}$. 19
Figure 12 Temporary 19
Figure 13 Probability distribution for the staggered quantum walk on a line after 50 steps, with initial condition $|\Psi(0)\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$, for multiple angles. 22
Figure 14 $|\Psi(0)\rangle = |0\rangle$ 22
Figure 15 $|\Psi(0)\rangle = |1\rangle$ 22
Figure 16 Discrete-time coined quantum walk search for a complete graph with 16, 32 and 64 nodes. 25
Figure 17 Maximum probability of the marked element as a function of the θ value plotted from 0 to π for number of nodes $N = 64, 128$ and 256. 26

Figure 18 Staggered quantum walk search for a complete graph with 16, 32 and 64 nodes. 27
Figure 19 Value of the difference between the largest eigenvalue and the second largest, plotted as a function of γN, for $N = 512$. 28
Figure 20 Continuous quantum walk search for a complete graph with 16, 32 and 64 vertices. 29
Figure 21 Douglas wang shift operator 31
Figure 22 Douglas wang coined quantum walk circuit 31
Figure 23 Toffoli decomposition 31
Figure 24 General decomposition 32
Figure 25 Temp 33
Figure 26 Temp 33
Figure 27 Temp 34
Figure 28 Temp 34
Figure 29 Temp 37
Figure 30 Temp 37
Figure 31 temp 37
Figure 32 Temp 38
Figure 33 Temp 38
Figure 34 Temp 39
Figure 35 Temp 40
Figure 36 Temp 40
Figure 37 Temp 40

LIST OF TABLES

1

INTRODUCTION

1.1 BRIEF HISTORY OF QUANTUM COMPUTING

1.2 CLASSICAL AND QUANTUM RANDOM WALKS

1.3 STATE OF THE ART QUANTUM WALKS IMPLEMENTATIONS

1.4 TEXT OVERVIEW AND CONTRIBUTIONS

2

QUANTUM COMPUTING

2.1 GROVER'S ALGORITHM

Searching through an unstructured database is a task classically achieved by exhaustively evaluating every element in the database. Assume there exists a black box (oracle) that can be asked to find out if two elements are equal. Since we're looking for a specific element in a database of size N, we'd have to query the oracle on average $\frac{N}{2}$ times, or in the worst case N times.

Grover's algorithm, presented in Grover (1996), comes as a quantum alternative to this type of problems, taking advantage of superposition by increasing desirable states' amplitudes through a process called *amplitude amplification*. This method has a quadratic gain over the classical counterpart Boyer et al. (1998), being able to find a target element in expected time $\mathcal{O}(\sqrt{N})$.

Let us now expand on the inner workings of the black box. We start by focusing on searching indexes instead of directly evaluating the element and we assume $N = 2^n$, n being a positive integer. We can now define a function $f : \{0, 1, ..., N-1\}$ that returns 1 when evaluating the desired (marked) element and 0 otherwise. Since this function is to be applied to a quantum system, we must build a unitary operator $\mathcal{O}$

$$\mathcal{O} |x\rangle |i\rangle = |x\rangle |i \oplus f(x)\rangle . \tag{1}$$

where $|x\rangle$ is the index register, $\oplus$ is the binary sum operation and $|i\rangle$ is a qubit that is flipped if $f(x) = 1$.

The action of the oracle on state $|0\rangle$ will be

$$\mathcal{O} |x\rangle |0\rangle = \begin{cases} |x_0\rangle |1\rangle , & \text{if } x = x_0 \\ |x\rangle |0\rangle , & \text{otherwise.} \end{cases} \tag{2}$$

where x_0 is the marked element. More generically, $\mathcal{O}$ can be written as

$$\mathcal{O} |x\rangle = (-1)^{f(x)} |x\rangle . \tag{3}$$

This offers a bit of insight into the oracle, it *marks* the solutions to the search problem by applying a phase shift to the solutions. The question now is, what is the procedure that determines a solution x_0 using $\mathcal{O}$ the minimum number of times? The answer lies in the amplitude amplification section of Grover's search, starting with the creation of a uniform superposition

$$|\Psi_0\rangle = H^{\otimes n} |x\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \tag{4}$$

where $H^{\otimes n}$ is the *Hadamard* operator applied to an arbitrary number of states.

If we were to measure $|x\rangle$ at this point, the superposition would collapse to any of the base states with the same probability $\frac{1}{N} = \frac{1}{2^n}$, which means that on average, we'd need to try $N = 2^n$ times to guess the correct item. This is where amplitude amplification comes into effect, by means of a second unitary operator

$$\mathcal{D} = (2 |\Psi_0\rangle \langle\Psi_0| - I) = H^{\otimes n}(2 |0\rangle \langle 0| - I)H^{\otimes n} \tag{5}$$

This operator applies a conditional phase shift, with every computational basis state except $|0\rangle$ receiving a phase shift. This can also be described as the *inversion about the mean*, for a state of arbitrary amplitudes

$$|\phi\rangle = \sum_{k=0}^{N-1} \alpha_k |k\rangle \tag{6}$$

the action of $\mathcal{D}$ on state ϕ will be

$$\mathcal{D} |\phi\rangle = \sum_{k=0}^{N-1} (-\alpha_k + 2\langle\alpha\rangle) |k\rangle \tag{7}$$

where $\langle\alpha\rangle$ is the average of α_k

$$\langle\alpha\rangle = \frac{1}{N} \sum_{k=0}^{N-1} \alpha_k |k\rangle \tag{8}$$

The evolution operator that performs one step of the algorithm is then

$$\mathcal{U} = \mathcal{D}\mathcal{O} \tag{9}$$

and after t steps the state of the system is

$$|\Psi(t)\rangle = \mathcal{U}^t |\Psi_0\rangle . \tag{10}$$

2.1.1 *One marked element*

The optimal number of steps is, as aforementioned, proportional to $\sqrt{N}$. More precisely, if there's only one solution, maximum probability can be reached in *approximately* $\frac{\pi}{4}\sqrt{N}$ iterations. In order to show that this is the case, an iteration will be formally defined here as the process that transforms the state

$$|\Psi(k,l)\rangle = k\,|i_0\rangle + \sum_{i\neq i_0} l\,|i\rangle \tag{11}$$

into state $\left|\Psi(\frac{N-2}{N}k + \frac{2(N-1)}{N}l, \frac{N-2}{N}l - \frac{2}{N}k)\right\rangle$. Amplitudes l and k are real numbers that statisfy $k^2 + (N-1)l^2 = 1$. Running m iterations over state $|\Psi_0\rangle$ will eventually lead to state $|\Psi_j\rangle = |\Psi(k_j, l_j)\rangle$ after the $j-th$ iteration, where $k_0 = l_0 = \frac{1}{\sqrt{N}}$ and

$$\begin{cases} k_{j+1} = \frac{N-2}{N}k_j + \frac{2(N-1)}{N}l_j \\ l_{j+1} = \frac{N-2}{N}l_j + \frac{2}{N}k_j. \end{cases} \tag{12}$$

After the last iteration, the system will be in state $|\Psi_m\rangle$ with a certain amplitude. If that amplitude corresponds to the marked element x_0, then it is said that the algorithm was successful.

Grover (1996) proves that there exists a value of $m < \sqrt{2N}$ such that the probability of success is at least $\frac{1}{2}$. However the probability of success does not linearly increase with the number of iterations, in fact for $m = \sqrt{2N}$ the system will succeed less that 1 in 10 times. Boyer et al. (1998) argues that an explicit value of m is needed, and it is achieved by finding a closed form formula for k_j and l_j. The first step is to define an angle θ so that $\sin^2\theta = \frac{1}{N}$, and equation 12 will become

$$\begin{cases} k_{j+1} = \sin\left((2j+1)\theta\right) \\ l_{j+1} = \frac{1}{\sqrt{N-1}}\cos\left((2j+1)\theta\right). \end{cases} \tag{13}$$

In order to maximize the probability of success, one must find a value of m so that $k_m = 1$ and l_m is as close to 0 as possible. The value of k after m iterations will be at it's maximum when $\sin\left((2m+1)\theta\right) = \frac{\pi}{2}$, and solving the trigonometric equation leads to a value of $m = \frac{\pi - 2\theta}{4\theta}$. Conversely, $l_{\widetilde{m}} = 0$ when $\widetilde{m} = \frac{\pi-2\theta}{4\theta}$ for an integer number of $\widetilde{m}$. Setting m to the nearest lower integer of $\frac{\pi}{4\theta}$ will lead to

$$|m - \widetilde{m}| \leq \frac{1}{2} \iff |(2m+1)\theta - (2\widetilde{m}-1)\theta| \leq \frac{\pi}{2}. \tag{14}$$

By definition, $(2\widetilde{m}+1)\theta = \frac{\pi}{2}$ which means that $|\cos((2m+1)\theta)| \leq |\sin\theta|$. The probability of failure after m iterations can then be written as

$$(N-1)l_m^2 = \cos^2((2m+1)\theta) \leq \sin^2\theta = \frac{1}{N}. \tag{15}$$

Failure decreases as the number of elements increases. The run time of the algorithm will be

$$m \leq \frac{\pi}{4\theta} \leq \frac{\pi}{4}\sqrt{N} \tag{16}$$

since $\theta \geq \sin\theta = \frac{1}{\sqrt{N}}$. This means that, for a large N, the number of iterations that maximizes the probability of success will be very close to $\frac{\pi}{4}\sqrt{N}$.

Figure 1 was obtained by coding the apropriate operators as to simulate the system presented in equation 10. The unitary evolution operator was applied approximately

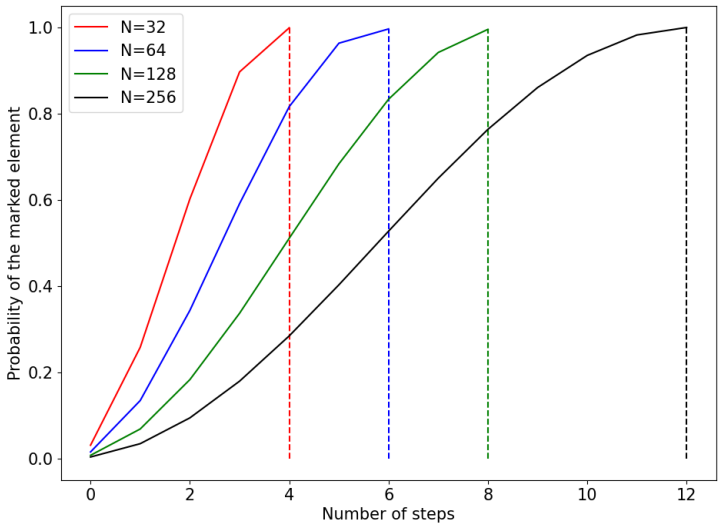


Figure 1: Grover one marked element temp.

$\frac{\pi}{4}\sqrt{N}$ times and the amplitudes associated with those states were stored as a probability distribution. Filtering the probability of the marked element and plotting it against the number of steps, shows that the maximum is indeed reached after the said number of iterations. It also shows that the maximum probability for $N = 16$ is lower than for $N = 128$, which makes sense since the the probability of success is maximized for larger values of N.

2.1.2 *Multiple marked elements*

When there's more than one element marked by the oracle, the number of iterations to achieve maximum probability changes. In fact, the latter part of this section will be used to discuss the case where one single iteration of this algorithm is enough to achieve maximum probability.

Firstly, one must define a set A that is composed of all the marked elements and set B of the remaining. The state from equation 11 will become

$$|\Psi(k,l)\rangle = \sum_{i\in A} k\,|i\rangle + \sum_{x\in B} l\,|x\rangle\,. \tag{17}$$

Assuming t marked elements, iterating over this state will result in

$$\left|\Psi\left(\frac{N-2t}{N}k + \frac{2(N-1)}{N}l, \frac{N-2}{N}l - \frac{2}{N}k\right)\right\rangle. \tag{18}$$

Choosing an angle θ such that $sin^2\theta = \frac{t}{N}$, allows the definition of the amplitudes associated with the states after j iterations

$$\begin{cases} k_j = \frac{1}{\sqrt{t}}\sin\left((2j+1)\theta\right) \\ l_j = \frac{1}{\sqrt{N-t}}\cos\left((2j+1)\theta\right). \end{cases} \tag{19}$$

Similarly to the one solution case, it can be shown that setting the number of iterations

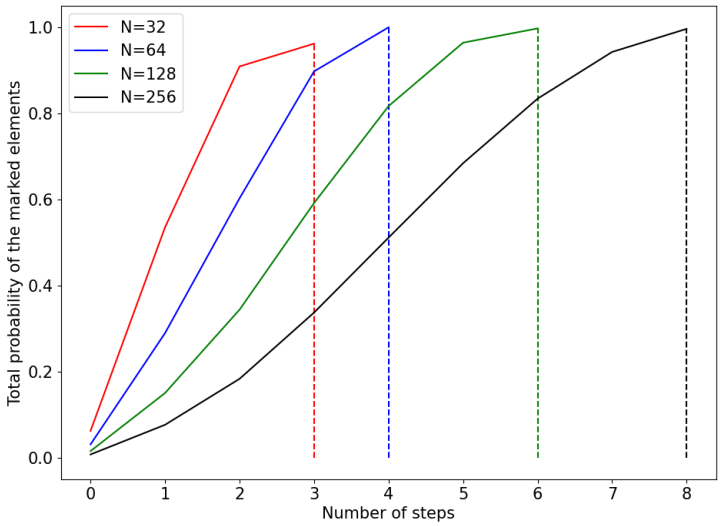


Figure 2: Grover Multiple marked temp.

m, to the nearest lower integer of $\frac{\pi}{4\theta}$ will result in a probability of failure $(N-t)l_m^2 \leq \frac{t}{N}$. Because $\theta \geq \sin\theta = \sqrt{\frac{t}{N}}$ then

$$m \leq \frac{\pi}{4\theta} \leq \frac{\pi}{4}\sqrt{\frac{N}{t}}. \tag{20}$$

From a more practical perspective, if one were to mark two elements of a 64 element set, maximum probability is expected to be reached in approximately 4 steps, since $\lfloor \frac{\pi}{4}\sqrt{\frac{64}{2}} \rfloor = 4$. Likewise, for $N = 256$, the number of iterations is rounded to 8, which is plotted along several other values of N in figure 2. The y-axis is now the sum total probability of the marked elements and the x-axis represents the range of steps that spans from 0 to $\lfloor \frac{\pi}{4}\sqrt{\frac{N}{2}} \rfloor$ for each N. Again, the probability of success approaches 1 as N increases. However, comparing to figure 1, the number of iterations that maximize probability is lower for each N, in agreement with equation 20.

2.1.3 *Single-Shot Grover*

An interesting case arises when the number of marked elements is set to $t = \frac{N}{4}$, because

$$sin^2\theta = \frac{\frac{N}{4}}{N} = \frac{1}{4} \iff \sin\theta = \frac{1}{2} \iff \theta = \frac{\pi}{6}. \tag{21}$$

Note that there are an infinite number of negative and positive solutions, but equation 21

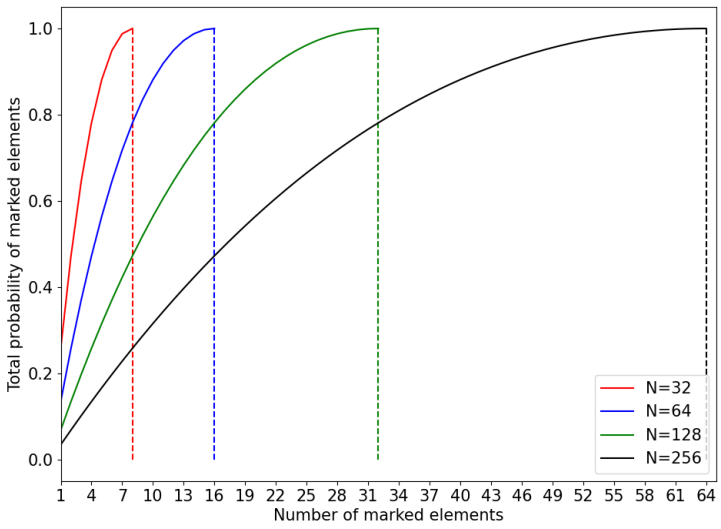


Figure 3: Single Shot temp

reflects the only relevant one in this context. As a consequence, amplitudes associated with state $|\Psi(k_1, l_1)\rangle$ become

$$\begin{cases} k_1 = \frac{1}{\sqrt{t}} \sin\left((2+1)\theta\right) = \frac{1}{\sqrt{\frac{N}{4}}} \sin\left(3\frac{\pi}{6}\right) = \frac{2}{\sqrt{N}} \\ l_1 = \frac{1}{\sqrt{N-t}} \cos\left((2+1)\theta\right) = \frac{1}{\sqrt{N-\frac{N}{4}}} \cos\left(3\frac{\pi}{6}\right) = 0. \end{cases} \tag{22}$$

These results show that the amplitudes associated with the marked states double in relation to $|\Psi_0\rangle$ and the remaining states disappear after only one iteration. This behaviour can be seen in figure 3, where the total probability of marked elements reaches 1 once the number of marked elements is $\frac{1}{4}$ of the total elements.

3

QUANTUM WALKS

3.1 CLASSICAL RANDOM WALK

The term *random walk*, firstly introduced by Pearson (1905), is classically defined as a stochastic process that models the path a walker would take through a mathematical space, where each step made by the walker is random. This can be used to model systems such as a molecule displaying Brownian motion in a fluid, or even fluctuating stock prices as can be seen in Sottinen (2001).

The simplest instance of this walk is on a discretely numbered line, whose mathmatical space is composed of integer numbers. Here, the walker can only advance with equal probability in one of two directions, depending on the outcome of a random event such as tossing a coin. This was coded in Python, and the result of iterating the walk several times is a binomial distribution centered around the starting position.

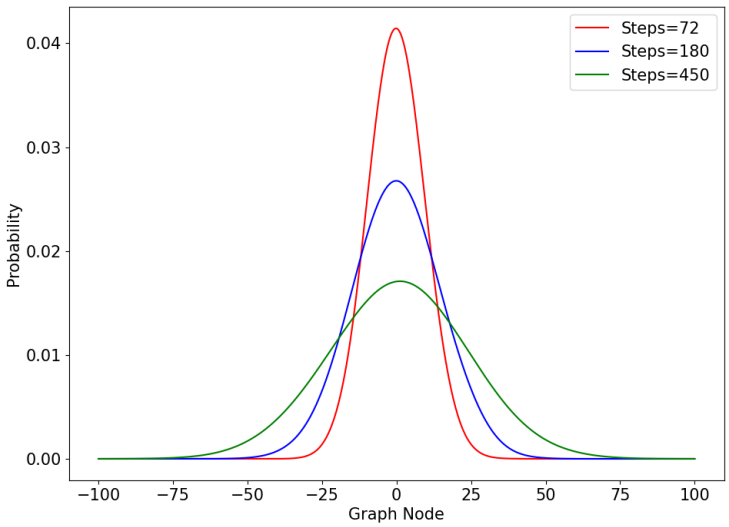


Figure 4: Classical Walk Temp.

The number of steps (iterations) directly affects how far the walker can reach, as can be seen in figure 4. As the number of steps increases, the height of each curve at the starting position decreases and the width of the curves increases. This relationship can be captured by the *position standard deviation*, and Portugal (2018) shows that the standard deviation is

$$\sigma(t) = \sqrt{t}. \tag{23}$$

In other words, equation 23 represents the rate at which a walker moves away from the origin.

Note that this algorithm can be abstracted to graphs of higher dimensions. For example, in a two dimensional lattice, a walker would be transversing a plane with integer coordinatesi, choosing one of four directions in every intersection. Notably, Pólya (1921) proved that a walker in a two dimensional lattice will almost surely return to the origin at some point. However, the probability of returning to the origin decreases as the number of dimensions increases, as shown by Montroll (1956) and Finch (2003).

It is worth noting that a random walk, over a graph whose nodes are weighed and directed, is analagous to a *discrete-time Markov chain*[1].

The following sections will be used to describe various models of a quantum counterpart of the classical random walk.

3.2 COINED QUANTUM WALK

In the quantum case, the walker is a quantum system whose position on a discretely numbered line, is described by a vector $|x\rangle$ in Hilbert Space. The next position of the system will depend, in part, of a unitary operator, which can be viewed as a quantum coin. The analogy is, if the coin is tossed and rolls "heads", for example, the system transitions to position $|x+1\rangle$, otherwise it advances to $|x-1\rangle$. From a physical perspective, this coin can be the spin of an electron or the chirality of a particle, for example, and the outcome of measuring these properties decides whether the walker moves left or right. The coin is a unitary operator defined as

$$\begin{cases} C\,|0\rangle\,|x\rangle = a\,|0\rangle\,|x\rangle + b\,|1\rangle\,|x\rangle \\ C\,|1\rangle\,|x\rangle = c\,|0\rangle\,|x\rangle + d\,|1\rangle\,|x\rangle\,, \end{cases} \tag{24}$$

1 A Markov chain can be described as a sequence of stochastic events where the the probability of each event depends only on the state of the previous event.

where a, b, c and d are the amplitudes associated with each outcome of the coin toss. One of the most commonly used coins is the unbiased coin, also known as Hadamard operator

$$H = \begin{pmatrix} a & c \\ b & d \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{25}$$

which will be the one used in this example.

The Hilbert space of the system is $\mathscr{H} = \mathscr{H}_C \otimes \mathscr{H}_P$, where $\mathscr{H}_C$ is the two-dimensional Hilbert space associated with the coin and $\mathscr{H}_P$ is the Hilbert space of the walker.

The transition from $|x\rangle$ to either $|x+1\rangle$ or $|x-1\rangle$ must be described by a unitary operator, the *shift operator*

$$\begin{cases} \mathcal{S} |0\rangle |x\rangle = |0\rangle |x+1\rangle \\ \mathcal{S} |1\rangle |x\rangle = |1\rangle |x-1\rangle , \end{cases} \tag{26}$$

that can also be described by

$$S = |0\rangle \langle 0| \otimes \sum_{x=-\infty}^{x=\infty} |x+1\rangle \langle x| + |1\rangle \langle 1| \otimes \sum_{x=-\infty}^{x=\infty} |x-1\rangle \langle x| . \tag{27}$$

It follows that the operator that describes the dynamics of the quantum walk will be given by

$$U = S(C \otimes I) = S(H \otimes I). \tag{28}$$

Consider a quantum system located at $|x=0\rangle$ with coin state $|0\rangle$, for $t=0$. It's state will be described by

$$|\Psi(0)\rangle = |0\rangle |x=0\rangle . \tag{29}$$

After t steps

$$|\Psi(t)\rangle = U^t |\Psi(0)\rangle , \tag{30}$$

more explicitly

$$|\Psi(0)\rangle \xrightarrow{U} |\Psi(1)\rangle \xrightarrow{U} |\Psi(2)\rangle \xrightarrow{U} (...) \xrightarrow{U} |\Psi(t)\rangle . \tag{31}$$

In other words, the coined quantum walk algorithm consists on applying the coin operator followed by the shift operator a certain number of times. Iterating this twice, evolves the system to the following respective states

$$|\Psi(1)\rangle = \frac{|0\rangle |x=-1\rangle + |1\rangle |x=1\rangle}{\sqrt{2}} \tag{32}$$

$$|\Psi(2)\rangle = \frac{|0\rangle |x=-2\rangle + |1\rangle |x=0\rangle + |0\rangle |x=0\rangle - |1\rangle |x=2\rangle}{2} \tag{33}$$

$$\tag{34}$$

If one were to measure the system after the first application of $\mathcal{U}$, it would be expected to see the walker at $x = 1$ with probability $P(x) = \frac{1}{2}$, and at $x = -1$ with $P(x) = \frac{1}{2}$ aswell. Measure the system t times, after each application of $\mathcal{U}$, and the result is a binomial probability distribution similar to the one in 4. The conclusion is that repetitive measurement of a coined quantum walk system reduces to the classical case, which means that any desired quantum behaviour is lost.

It is possible, however, to make use of the quantum correlations between different positions to generate constructive or destructive interference, by applying the Hadamard and shift operators successively without intermediary measurements. The consequences of interference between states become very apparent after only 3 iterations

$$|\Psi(3)\rangle = \frac{|1\rangle |x = -3\rangle - |0\rangle |x = -1\rangle + 2(|0\rangle + |1\rangle) |x = 1\rangle + |0\rangle |x = 3\rangle}{2\sqrt{2}}. \tag{35}$$

Even though an unbiased coin was used, this state is not symmetric around the origin and the probability distributions will not be centered in the origin. Moreover, Portugal (2018) shows that the standard deviation will be

$$\sigma(t) \approx 0.54t. \tag{36}$$

This means that the standard deviation for the coined quantum walk grows linearly in time, unlike the classical case which grows with $\sqrt{t}$, as was seen in equation 23. The implication is that the quantum walk displays *ballistic* behaviour, as is reviewed in Venegas-Andraca (2012). This behaviour is usually defined in the context of a moving free particle with unit velocity in a single direction, which is expected to be found at $x = t$ after t steps. The velocity of a walker in a Hadamard quantum walk is approximately half of the free particle example, which is still a quadratic improvement over the classical random walk.

This quadratic gain implies exponentially faster hitting times in certain graphs, as shown by Childs et al. (2002), meaning improvements to problems that require transversing graphs. Ambainis (2007) also shows advantages of the coined quantum walk model in element distinctness problems, and Childs and Goldstone (2004) show advantages in spatial search problems, which will be studied in a later chapter.

In order to study this distribution, a simulation of the coined quantum walk was coded in *Python*. Figure 5 is the result of using the Hadamard coin and the initial condition in equation 29, for varying numbers of steps. Analyzing the plot, it is noticeable that the distributions are asymmetric. The probability of finding the walker on the right-hand side is much larger than on the left, with a peak around $x \approx \frac{t}{\sqrt{2}}$. Regardless of number of steps, this peak is always present (albeit in varying positions), which is to say that the walker can always be found moving in a uniform fashion away from the origin, consistent with ballistic behaviour.

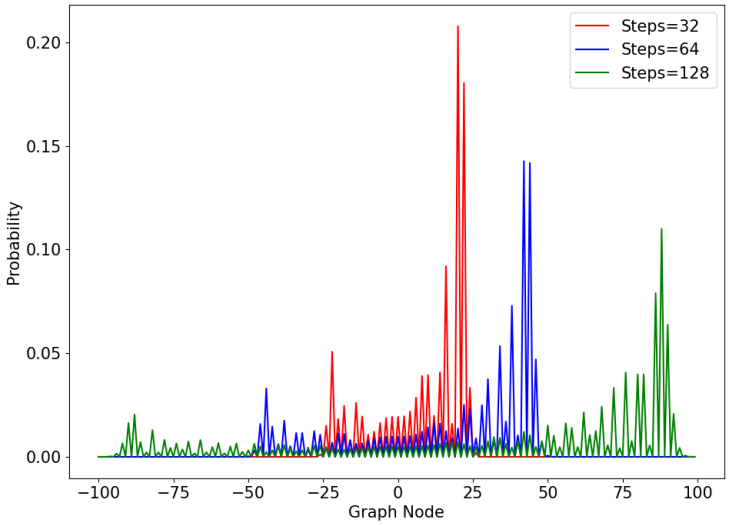


Figure 5: Probability distribution for the coined quantum walk on a line, after 100 steps, with initial condition $|\Psi(0)\rangle = |0\rangle\,|x = 0\rangle$ and the Hadamard coin.

Another interesting case study is to find if this behaviour is preserved for a symmetric distribution around the origin. For this purpose, one must first understand where the asymmetry comes from. The Hadamard operator flips the sign of state $|1\rangle$, hence more

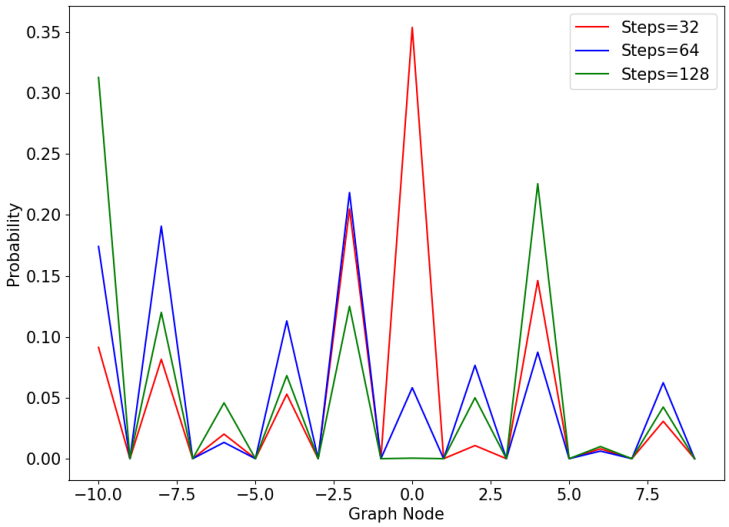


Figure 6: Probability distribution for the coined quantum walk on a line, after 100 steps, with initial condition $|\Psi(0)\rangle = -|1\rangle\,|x = 0\rangle$ and the Hadamard coin.

terms are cancelled when the coin state is $|1\rangle$. Since $|0\rangle$ was defined to induce movement

to the right, the result is as shown in 5. Following this logic, it would be expected that an initial condition

$$|\Psi(0)\rangle = |1\rangle\,|x=0\rangle\,, \tag{37}$$

would result in more cancellations when the coin state is $|0\rangle$, thus the walker would be more likely found in the left-hand side of the graph. This is indeed what happens, as figure 6 is a mirror image of figure 5. The walker still moves away from the origin with ballistic behaviour, but in opposite direction. The peaks behave in a similar fashion, being instead found at $x \approx -\frac{t}{\sqrt{2}}$.

In order to obtain a symmetrical distribution, one must superpose the state in equation 29 with the state in equation 37. However, in order to not cancel terms before the calculation of the probability distribution, one must multiply state $|1\rangle$ with the imaginary unit, i

$$|\Psi(0)\rangle = \frac{|0\rangle + i\,|1\rangle}{\sqrt{2}}\,|x=0\rangle\,. \tag{38}$$

This works because the entries of the Hadamard operator are real numbers. Terms with the imaginary unit will not cancel out with terms without it, thus the walk can proceed to both left and right, as it is shown in figure 7.

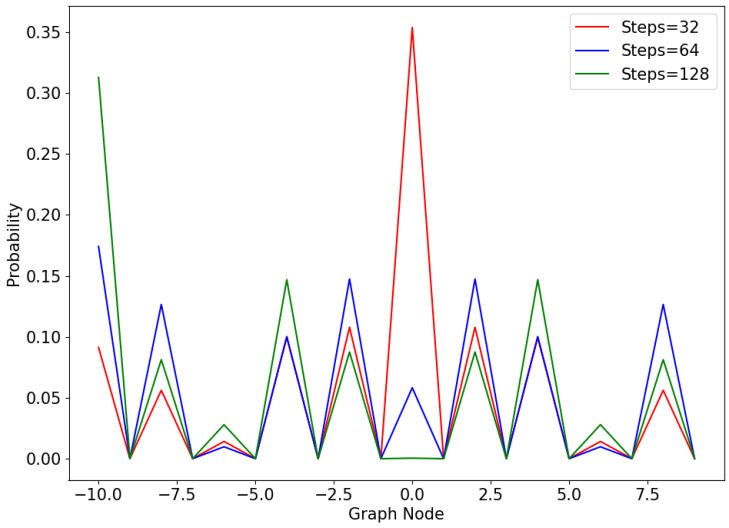


Figure 7: Probability distribution for the coined quantum walk on a line, after 100 steps, with initial condition $|\Psi(0)\rangle = \frac{|0\rangle - i|1\rangle}{\sqrt{2}}\,|x=0\rangle$ and the Hadamard coin.

The probability distribution is now symmetric and it is spread over the range $[-\frac{t}{\sqrt{2}}, -\frac{t}{\sqrt{2}}]$ with peaks around $|x| \approx \pm\frac{t}{\sqrt{2}}$. This means that if the position of the walker was measured

at the end, it would be equally probable to find him either in the left side or the right side of the graph , which is not possible in a classical ballisitic motion.

All of the previous examples are in sharp contrast with the classical random walk distribution in figure 4. There, maximum probability is reached at $x = 0$ since there are approximately equal steps in both directions. Furthermore, the further the vertex is away from the origin, the less likely the walker is to be found there. However, in the quantum case, the walker is more likely to be found away from the origin as the number of steps increases. More specifically, the walk spreads quadratically faster than the classical counterpart.

This is but one model of a quantum random walk. As it will be seen in further sections, there are other approaches to creating both discrete and continuous quantum walk models that do not use a coin.

3.3 CONTINUOUS-TIME QUANTUM WALK

The continuous-time random walk model on a graph is a Markov process where transitions have a fixed probability per unit time γ of moving to adjacent vertices, firstly introduced by Montroll and Weiss (1997). Consider a graph G with N vertices and no self-loops, this walk can be defined by the linear differential equation that describes the probability of jumping to a connected vertex in any given time

$$\frac{dp_i(t)}{dt} = \gamma \sum_j L_{ij} p_j(t), \tag{39}$$

where L is the Laplacian defined as $L = A - D$. A is the adjacency matrix that represents each vertex connection, given by

$$A_{ij} = \begin{cases} 1, & \text{if } (i,j) \in G \\ 0, & \text{otherwise,} \end{cases} \tag{40}$$

and D is the diagonal matrix $D_{jj} = deg(j)$ corresponding to the degree[2] of a vertex j.

In the quantum case, the nodes are quantum states that form the basis for the Hilbert space. The continuous-time quantum walk model will also be described by a differential equation, the Schrödinger equation

$$i\hbar \frac{d\,|\Psi(t)\rangle}{dt} = \hat{H}\,|\Psi(t)\rangle\,, \tag{41}$$

2 The degree of a vertex refers to the number of edges that it is connected to.

where $\hat{H} = -\gamma L$ is the Hamiltonian of the system. More explicitly,

$$\hat{H}_{ij} = \begin{cases} deg(j)\gamma, & \text{if } i = j; \\ -\gamma, & \text{if } i \neq j \text{ and adjacent;} \\ 0, & \text{if } i \neq j \text{ and not adjacent.} \end{cases} . \tag{42}$$

A general state of a system $|\Psi(t)\rangle$ can be written as a function of it's complex amplitudes

$$q_i = \langle i|\Psi(t)|i|\Psi(t)\rangle\,, \tag{43}$$

which means 41 can be rewritten as

$$i\hbar\frac{dq_i(t)}{dt} = \sum_j \hat{H}_{ij}q_j(t). \tag{44}$$

This highlights the similarities between the Schrödinger equation and 39. One of the main differences is the complex phase i, which will result in a very different behaviour. Setting $\hbar = 1$ and solving the differential equation results in the evolution operator of this walk

$$U(t) = e^{-iHt} = e^{i(-\gamma L)t} = e^{-i\gamma(A+D)t} \tag{45}$$

In the regular graph case, where D is simply the degree of the whole graph multiplied by the identity matrix, A and D will commute, meaning that the evolution operator can be written in terms of the adjacency matrix

$$U(t) = e^{-i\gamma At+i\gamma Dt} = e^{-i\gamma At}e^{i\gamma Dt} = \phi(t)e^{-i\gamma At} \tag{46}$$

since the degree matrix becomes a global phase. Applying this operator to an initial condition $\Psi(0)$, will give the state of the system at a time t

$$|\Psi(t)\rangle = U(t)\,|\Psi(0)\rangle\,. \tag{47}$$

Considering a uni-dimensional quantum system, each vertex will have at most 2 other neighboring vertices, reducing equation 42 to

$$\hat{H}_{ij} = \begin{cases} 2\gamma, & \text{if } i = j; \\ -\gamma, & \text{if } i \neq j \text{ and adjacent;} \\ 0, & \text{if } i \neq j \text{ and not adjacent.} \end{cases} \tag{48}$$

For a more detailed visualization, this quantum walk model was coded in python and figure 8 was obtained setting the transition rate to $\gamma = \frac{1}{2\sqrt{2}}$ the initial condition to $|\Psi(0)\rangle = |0\rangle$

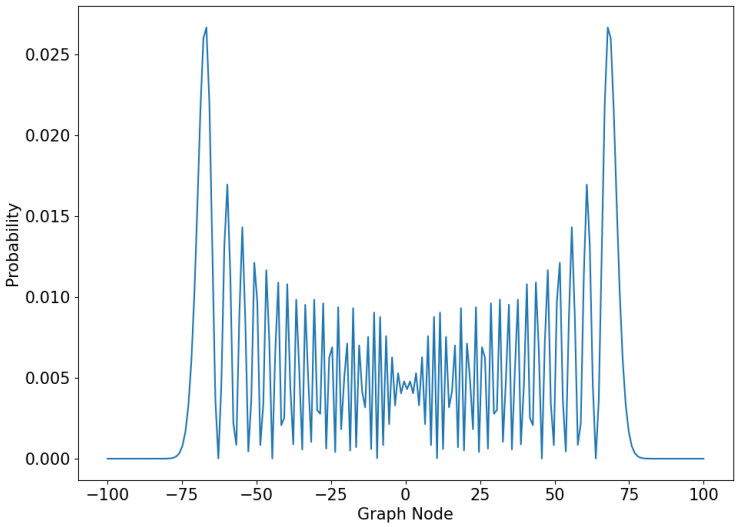


Figure 8: Probability distribution for the continuous-time quantum walk on a line, at t = 100, with initial condition $|\Psi(0)\rangle = |0\rangle$ and $\gamma = \frac{1}{2\sqrt{2}}$.

A brief look at figure 8 reveals several similarities to the coined quantum walk model of figure **??**. Both have two peaks away from the origin and low probability near the origin. However, in the previous quantum walk, these characteristics were altered as a function of the chosen coin and initial condition, whereas in this case different values of γ will influence the probability distribution. For example, a lower value of γ will limit the spread of the probability distribution, as is shown in figure 10.

Moreover, the effects of altering the initial condition will also differ in the continuous-time example. For example, setting the initial condition to the balanced superposition of states $|0\rangle$ and $|1\rangle$ has no effect on the overall pattern of the probability distribution as can be seen in figure 12. Both peaks still are still present and at the same distance from the origin, with intermediate amplitudes being attenuated relative to figure 8. This behaviour is in contrast with the discrete-time case, where a change in the initial condition would dictate the number of peaks and where they would appear.

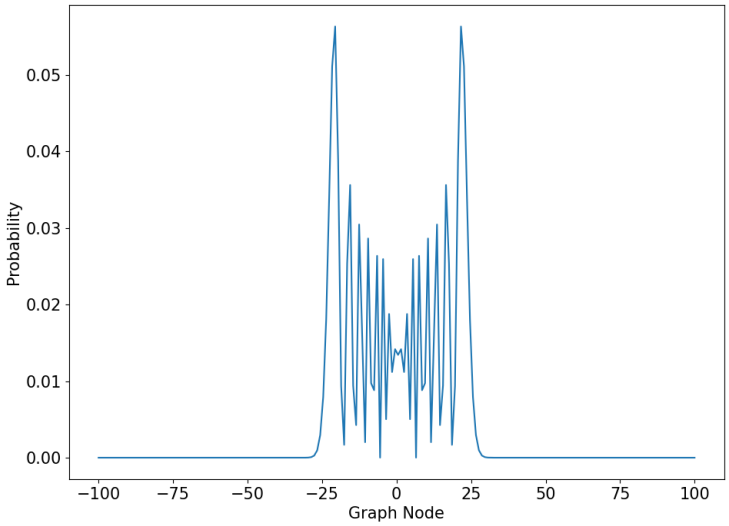


Figure 9: Probability distribution for the continuous-time quantum walk on a line, after 100 steps, with initial condition $|\Psi(0)\rangle = |0\rangle$ and $\gamma = \frac{1}{6\sqrt{2}}$.

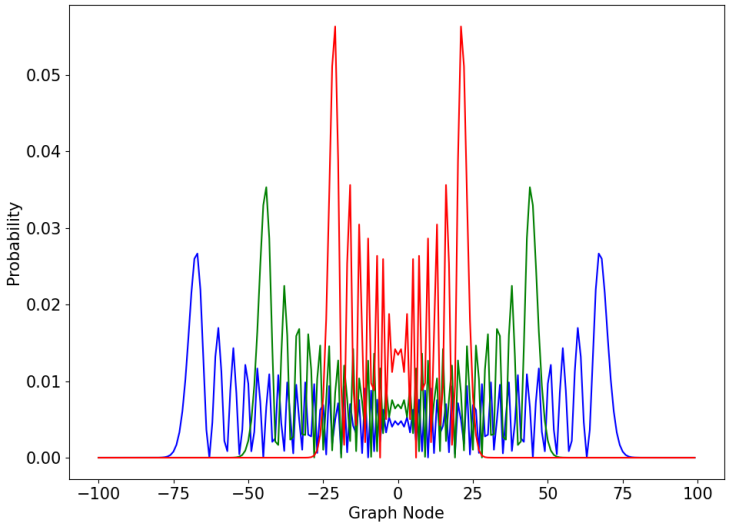


Figure 10: Temporary

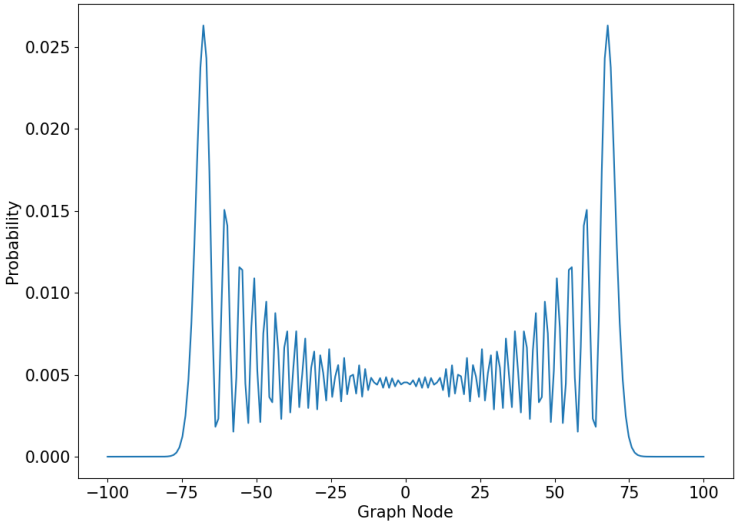


Figure 11: Probability distribution for the continuous-time quantum walk on a line, after 100 steps, with initial condition $|\Psi(0)\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $\gamma = \frac{1}{2\sqrt{2}}$.

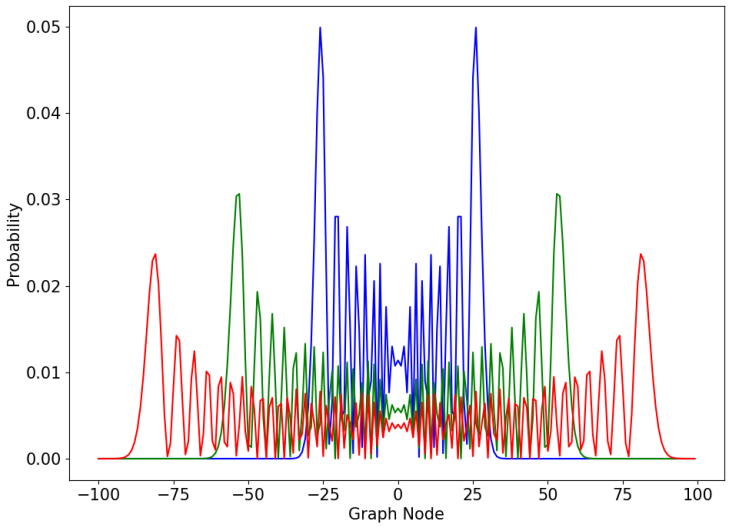


Figure 12: Temporary

3.4 STAGGERED QUANTUM WALK

Similarly to the continuous-time quantum walk, the staggered case aims to spread a transition probability to neighboring vertices but with discrete time steps. The notion of adjacency now comes from cliques[3], and the initial stage of this walk consists in partitioning the graph in several different cliques. This is called tessellation, and it is defined as the division of the set of vertices into disjoint cliques. An element of a tessellation $\mathscr{T}$ is called a polygon, and it's only valid if if all of its vertices belong to the clique in $\mathscr{T}$. The set of polygons of each tessellation must cover all vertices of the graph, and the set of tessellations ($\mathscr{T}_1$,$\mathscr{T}_2$,...,$\mathscr{T}_k$) must cover all edges.

These definitions allow the construction of operators H_1,H_2,...,H_k that will be used to propagate the probability amplitude locally, in each polygon. The state associated to each polygon is

$$\left|u_j^k\right\rangle = \frac{1}{\sqrt{|\alpha_j^k|}} \sum_{l \in \alpha_j^k} |l\rangle \tag{49}$$

where α_j^k is the $j-th$ polygon in the $k-th$ tessellation.

The unitary and Hermitian operator H_k, associated to each tessellation is defined in Portugal et al. (2017) as

$$H_k = 2 \sum_{j=1}^{p} \left|u_j^k\right\rangle \left\langle u_j^k\right| - I \tag{50}$$

Solving the time-independent Schrodinger equation for this Hamiltonian gives the evolution operator

$$U = e^{i\theta_k H_k} ... e^{i\theta_2 H_2} e^{i\theta_1 H_1} \tag{51}$$

where

$$e^{i\theta_k H_k} = \cos(\theta_k) I + i \sin(\theta_k) H_k \tag{52}$$

since $H_k^2 = I$.

The simplest use case of this quantum walk model is the one-dimensional lattice, where the minimum tessalations are

$$\mathscr{T}_\alpha = \{\{2x, 2x+1\} : x \in \mathbb{Z}\} \tag{53}$$

$$\mathscr{T}_\beta = \{\{2x+1, 2x+2\} : x \in \mathbb{Z}\} \tag{54}$$

3 A clique is defined as the subset of vertices of an undirected graph such that every two distinct vertices in each clique are adjacent.

Each element of the tessellation has a corresponding state, and the uniform superposition of these states is

$$|\alpha_x\rangle = \frac{|2x\rangle + |2x+1\rangle}{\sqrt{2}} \tag{55}$$

$$|\beta_x\rangle = \frac{|2x+1\rangle + |2x+2\rangle}{\sqrt{2}} \tag{56}$$

One can now define Hamiltonians H_α and H_β as

$$H_\alpha = 2\sum_{x=-\infty}^{+\infty} |\alpha_x\rangle \langle\alpha_x| - I \tag{57}$$

$$H_\beta = 2\sum_{x=-\infty}^{+\infty} |\beta_x\rangle \langle\beta_x| - I \tag{58}$$

The Hamiltonian evolution operator reduces to

$$U = e^{i\theta H_\beta} e^{i\theta H_\alpha} \tag{59}$$

and applying it to an initial condition $|\Psi(0)\rangle$ results in the time evolution operator

$$U|\Psi(t)\rangle = U^t |\Psi(0)\rangle \tag{60}$$

Having defined the time evolution operator, the walk is ready to be coded with a certain initial condition and θ value, to better understand how the probability distribution spreads through time.

For the first case study, the initial condition will be a uniform superposition of states $|0\rangle$ and $|1\rangle$ and the θ value will be varied in order to understand how this parameter impacts the walk

The overall structure of the probability distribution remains the same, the difference is that the walker is more likely to be found further away from the origin as the angle increases.

Another interesting case study is to see how the initial condition affects the dynamics of the system

Similarly to the coined case, each initial condition results in asymmetric probability distributions, $|\Psi(0)\rangle = |0\rangle$ leads to a peak in the left-hand side while condition $|\Psi(0)\rangle = |1\rangle$ results in a peak in the right-hand side. As was shown in 13, the uniform superposition of both these conditions results in a symmetric probability distribution.

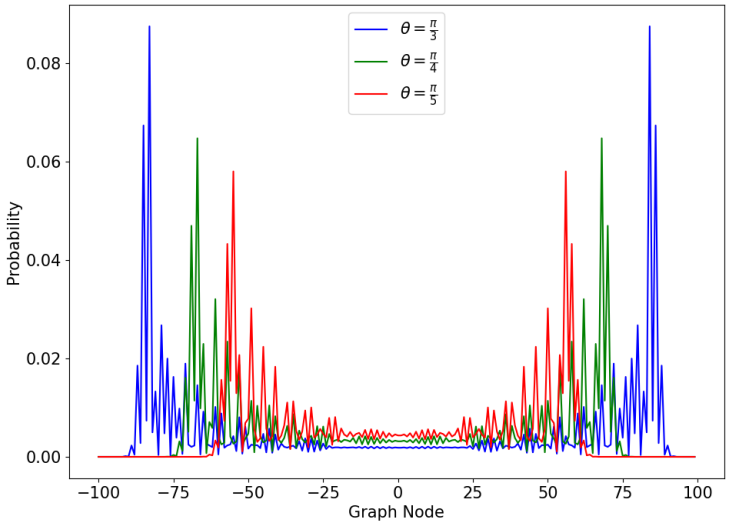


Figure 13: Probability distribution for the staggered quantum walk on a line after 50 steps, with initial condition $|\Psi(0)\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, for multiple angles.

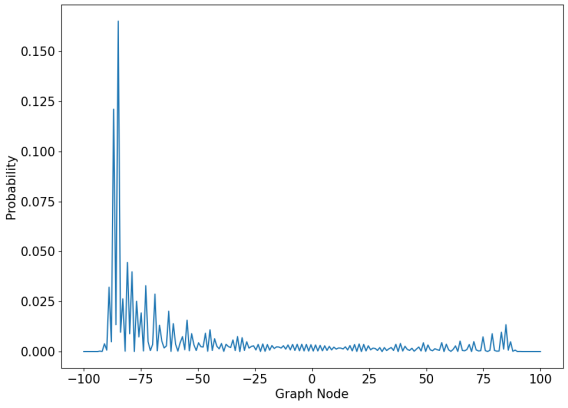


Figure 14: $|\Psi(0)\rangle = |0\rangle$

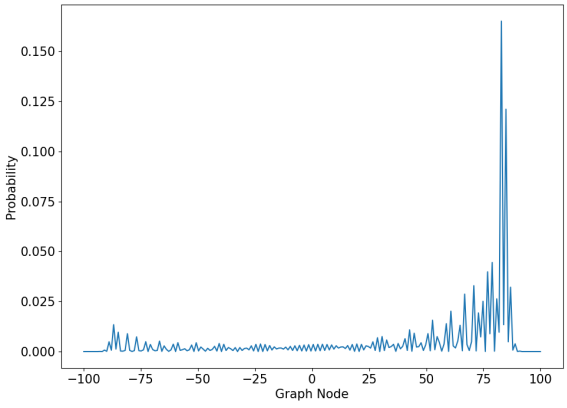


Figure 15: $|\Psi(0)\rangle = |1\rangle$

3.5 Search problems with Quantum Walks

In classical computation, a *spatial search problem* focuses on finding marked points in a finite region of space. Defining this region with graphs is fairly straightforward, the vertices of the graph are the search space, and the edges define what transitions are possible through the search space. As was previously mentioned in **??**, exhaustively searching through an unstructured space, by means of a classical random walk for example, would mean that in the worst case, one would have to take as many steps to find the marked points as there

are vertices in the graph. Quantum computing provides an alternative to this complexity through Grover's algorithm, and applying some of his ideas to the coined quantum walk not only allows a quantum counterpart to the random walk search, but also further insight into the algorithm itself.

3.5.1 *Coined*

Following Portugal (2018)'s definition, a good first step is to borrow the diffusion from Grover's algorithm and invert the sign of the state corresponding to the marked vertex while leaving unmarked vertices unchanged. This is done through the following operator

$$\mathcal{F}' = I - 2 \sum_{x \in M} |x\rangle \langle x| \tag{61}$$

where M is the set of marked vertices and $\mathcal{F}'$ is an analogue to Grover's oracle. For one marked vertex, this oracle can be written as

$$\mathcal{F}' = I - 2 |0\rangle \langle 0| \tag{62}$$

Notice that there is no loss of generality by choosing the marked vertex as 0, since the labelling of the vertices is arbitrary.

The next step is to combine the evolution operator from the coined quantum walk model with the oracle

$$U' = U\mathcal{F}' \tag{63}$$

Similarly to the simple coined case, the walker starts at $|\Psi(0)\rangle$ and evolves following the rules of an unitary operator U followed by the sign inversion of marked vertices. The walker's state after an arbitrary number of steps will be

$$\Psi(t) = (U')^t |\Psi(0)\rangle . \tag{64}$$

For a better understanding of the search problem in the coined quantum walk model, consider a graph where all the vertices are connected and each vertex has a loop that allows transitions to itself, as shown in figure **??**. The next step is to label the arcs using notation $\{(v, v'), v \geqslant 0 \wedge v' \leqslant N - 1\}$ where N is the total number of vertices and (v, v') are the position and coin value, respectively, in the coined model. The shift operator, now called *flip-flop* shift operator, is

$$S |v1\rangle |v2\rangle = |v2\rangle |v1\rangle . \tag{65}$$

The coin operator is defined as

$$C = I_N \otimes G \tag{66}$$

where

$$G = 2\left|D\right\rangle\left\langle D\right| - I \tag{67}$$

is the Grover coin with $|D\rangle$ being the diagonal state of the coin space. Given both of these operators, the evolution is defined for the unmarked case similarly to **??**

$$U = S(I \otimes G). \tag{68}$$

Marking an element in a complete graph is done through the following oracle

$$\mathcal{F}'' = \mathcal{F}' \otimes I = (I_N - 2\left|0\right\rangle\left\langle 0\right|) \otimes I_N = I_{N^2} - 2\sum_v \left|0\right\rangle\left|v\right\rangle\left\langle 0\right|\left\langle v\right|, \tag{69}$$

that can be seen, in the arc notation, as an operator that marks all arcs leaving 0.

Recalling 63, the modified evolution operator can be written as

$$U' = S(I \otimes G)\mathcal{F}'' = S(I \otimes G)\mathcal{F}' \otimes I = S(\mathcal{F}' \otimes G), \tag{70}$$

and the state of the system will evolve according to equation 64.

As was shown in Portugal (2018), maximum probability of the marked vertex is achieved after $\frac{\pi}{2}\sqrt{N}$ steps. Figure 16 is the result of coding and plotting the evolution of this probability distribution, for graphs of varying sizes. It shows that the probability is close to one at *approximately* the predicted ideal steps, because of the discrete nature of the walk. The probability distributions have a stair-like shape, because transitions in this model only occur on even numbered time steps, because of how the unmodified evolution operator was constructed.

3.5.2 *Staggered*

The process for defining the search problem in this model is similar to the coined quantum walk case. The oracle still inverts the sign of a certain state and amplifies it, and the system's state will still be described by equation 64. However,instead of using a coin, the staggered model takes advantage of the notions of cliques and tessellations, as was shown in chapter **??**, which means the unmodified evolution operator has to be defined for an undirected complete graph.

As was shown in figure **??**, the vertices in a complete graph are all neighbors. This is a special case because this is the only connected graph where the tessellation cover can be done by one tessellation, since the graph is it's own clique. The minimum tessellations required to cover this structures are defined by the one clique that encompasses all N nodes of the graph

$$\mathcal{T}_\alpha = \{\{0, 1, 2, ..., N-1\}\}. \tag{71}$$

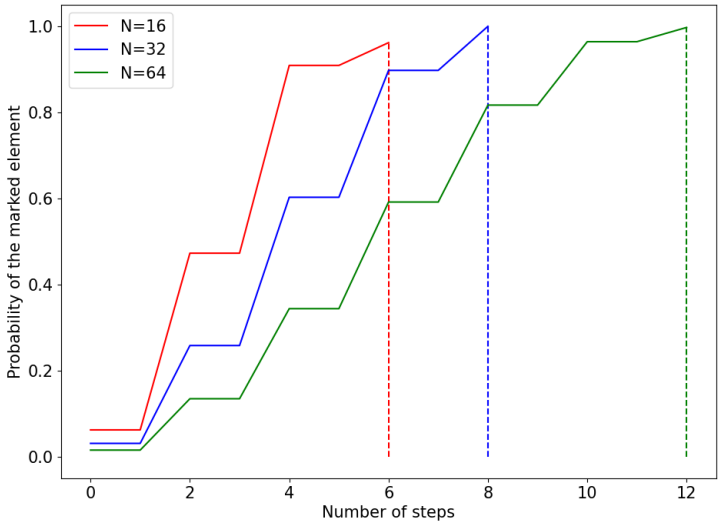


Figure 16: Discrete-time coined quantum walk search for a complete graph with 16, 32 and 64 nodes.

The associated polygon can then be described as the balanced superposition of all the nodes in the graph

$$|\alpha\rangle = \frac{1}{\sqrt{N}} \sum_{v=0}^{N-1} |v\rangle . \tag{72}$$

The Hamiltonian, as defined in 50, is

$$H_\alpha = 2 \sum_{0}^{1} |\alpha\rangle \langle\alpha| - I = 2 |\alpha_0\rangle \langle\alpha_0| - I \tag{73}$$

The unmodified evolution operator from equation 51

$$U = e^{i\theta_k H_k} ... e^{i\theta_2 H_2} e^{i\theta_1 H_1} \tag{74}$$

reduces to the single Hamiltonian case

$$U = e^{i\theta H_\alpha}. \tag{75}$$

The choice of the θ value is an important one, since maximum probability is achieved at $\theta = \frac{\pi}{2}$, as shown in figure 17. Since $H_\alpha^2 = I$, equation 75 can be rewritten as

$$U = e^{-i\frac{\pi}{2}H_\alpha} = \cos\frac{\pi}{2} I + i \sin\frac{\pi}{2} H_\alpha = iH_\alpha = i(2 |\alpha_0\rangle \langle\alpha_0| - I). \tag{76}$$

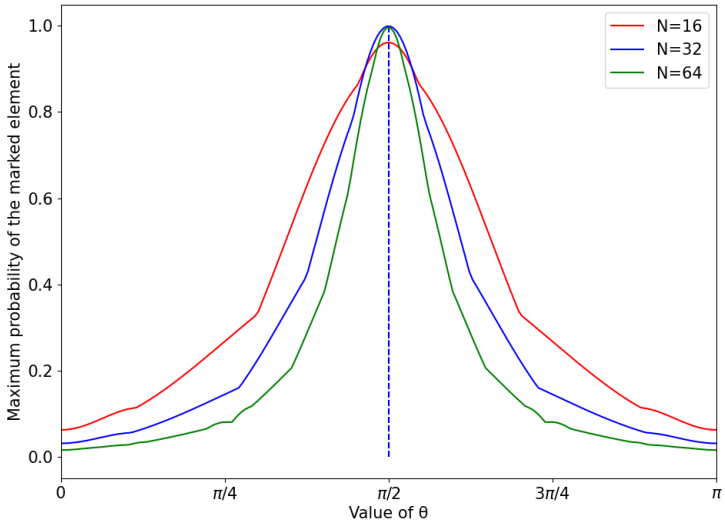


Figure 17: Maximum probability of the marked element as a function of the θ value plotted from 0 to π for number of nodes $N = 64, 128$ and 256.

Having defined the the evolution operator associated with the complete graph, the next step is to use the oracle

$$\mathcal{F}' = I_N - 2\left|0\right\rangle\left\langle 0\right|, \tag{77}$$

to create the modified evolution operator associated with the search

$$U' = U\mathcal{F}'. \tag{78}$$

The walk achieves the same result as Grover's algorithm after $\frac{\pi}{4}\sqrt{N}$ steps, as shown in figure 18. This plot also shows that the probabilities converge to 1 as N increases, this is because time is discretized and deviations to the ideal steps will matter less for bigger values of N.

3.5.3 *Continuous*

As was previously seen, the continuous-time quantum walk model is defined by an evolution operator obtained by solving Schrödinger's equation

$$U(t) = e^{-iHt}. \tag{79}$$

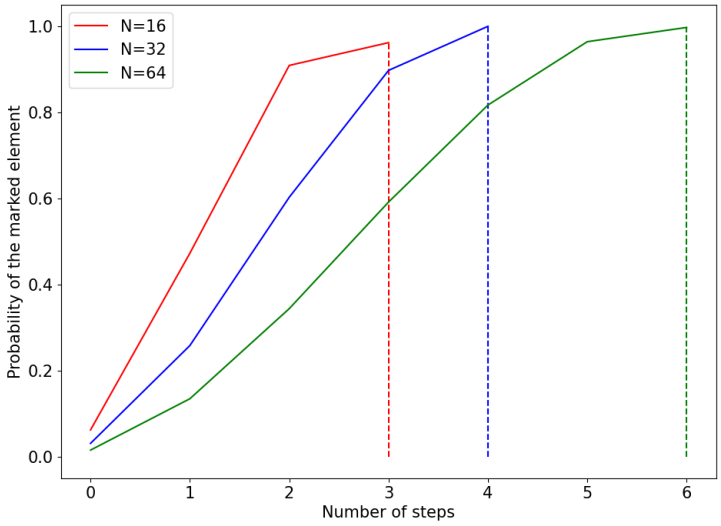


Figure 18: Staggered quantum walk search for a complete graph with 16, 32 and 64 nodes.

The search problem requires introducing an oracle to the Hamiltonian, that will mark an arbitrary vertex m

$$H' = -\gamma L - |m\rangle \langle m| . \tag{80}$$

Since the complete graph is a regular graph, the operator can be rewritten in terms of the adjacency matrix plus the marked element. Considering $|0\rangle$ is marked,

$$U'(t) = e^{iH't} = e^{i(-\gamma L - |0\rangle\langle 0|)t} = e^{i(-\gamma A + \gamma D - |0\rangle\langle 0|)t} = e^{-i\gamma(A+|0\rangle\langle 0|)t + i\gamma Dt}. \tag{81}$$

The degree matrix is again $D = dI$, which means it will commute with $A + |0\rangle \langle 0|$ and become a global phase

$$U'(t) = e^{-i\gamma(A+|0\rangle\langle 0|)t} e^{i\gamma Dt} = \phi(t) e^{-i\gamma(A+|0\rangle\langle 0|)t}. \tag{82}$$

As was show by Zalka (1999), the value of γ is crucial for the success of the search. As γ increases, the contribution of the marked element in the Hamiltonian decreases, and as γ approaches 0 the contribution of the adjacency matrix decreases. To find the optimum value, the Hamiltonian can be rewritten by adding multiples of the identity matrix to the adjacency matrix

$$H' = -\gamma(A + NI) - |0\rangle \langle 0| = -\gamma N |s\rangle \langle s| - |0\rangle \langle 0| \tag{83}$$

where $|s\rangle = \frac{1}{\sqrt{N}} \sum_i |i\rangle$. Now it is obvious that, for $\gamma = \frac{1}{N}$, the Hamiltonian is $H = -|s\rangle\langle s| - |0\rangle\langle 0|$. It's eigenstates are proportional to $|s\rangle \pm |w\rangle$ and eigenvalues are $-1 - \frac{1}{\sqrt{N}}$ and $-1 + \frac{1}{\sqrt{N}}$, respectively. This means that the evolution rotates from the state of balanced superposition to the marked vertex state in time $\frac{\pi}{\Delta E} = \frac{\pi}{2}\sqrt{N}$ which is, as was shown by Zalka (1999), optimal and equivalent to Grover's algorithm. Plotting ΔE as a function of γN, as can be seen in figure 19, has a minimum at $\gamma N = 1$. The difference between the largest eigenvalue and second largest, plotted in the y-axis, is the smallest for a value of $\gamma N = 1 \implies \gamma = \frac{1}{N}$, which will correspond to the maximum probability for the marked vertex, in optimal steps.

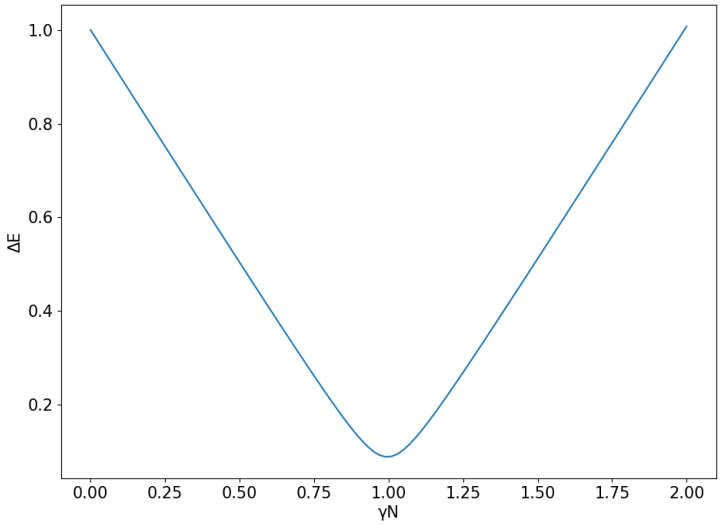


Figure 19: Value of the difference between the largest eigenvalue and the second largest, plotted as a function of γN, for $N = 512$.

Figure 20 shows the evolution of the probability of the marked vertex in time, which is continuous in this model. In contrast with previous models, the distributions are smooth and reach exactly one, since the walk is allowed to evolve to exactly the ideal time steps.

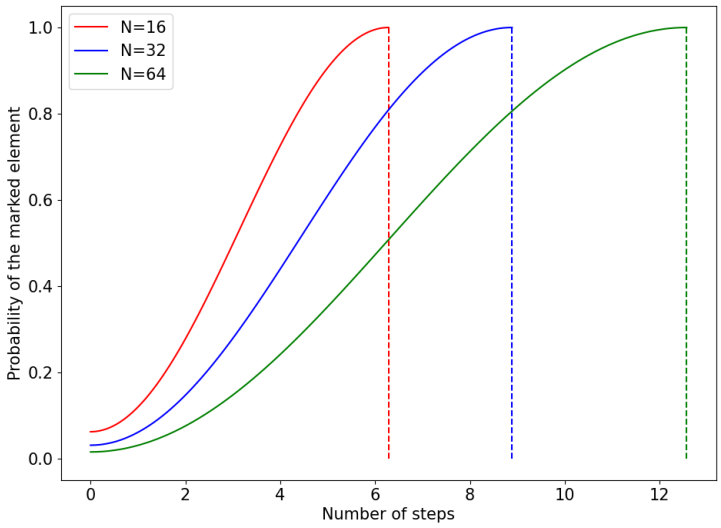


Figure 20: Continuous quantum walk search for a complete graph with 16, 32 and 64 vertices.

4

IMPLEMENTATIONS AND APPLICATIONS

4.1 COINED

Consider the example of a quantum walker on a discretely numbered cycle. It was seen that the evolution operator associated with such a system is, as was defined in equation 28

$$U = S(C \otimes I), \tag{84}$$

where S is a shift operator, defined in equation 27 as

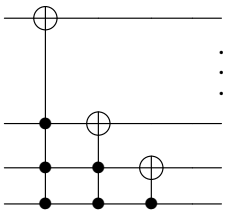
$$S = |0\rangle\langle 0| \otimes \sum_{x=-\infty}^{x=\infty} |x+1\rangle\langle x| + |1\rangle\langle 1| \otimes \sum_{x=-\infty}^{x=\infty} |x-1\rangle\langle x|, \tag{85}$$

that increments or decrements the position of the walker according to the coin operator C.

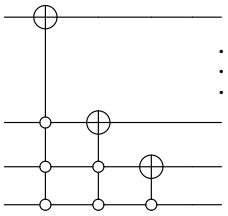
Previously, this system was simulated in Python by implenting it's equations. Now, the focus is to study a quantum circuit based on the work presented by Douglas and Wang (2009). This approach relies on multi-controlled CNOT gates in order to shift the state of the walker by $+1$ or -1, each with a probability associated with the chosen coin, as can be seen in figure 21.

The generalized CNOT gates act on the node states as a cyclic permutator, where each node is mapped to an adjacent state. This can be seen as the walker moving left or right, in the uni-dimensional graph example.

The coin operator will simply be a Hadamard gate acting on a single qubit. For a graph with 16 nodes, for example, 4 qubits are required to encode each node and an extra qubit for the coin. The circuit will then be as shown in figure 22. Note that this circuit limits the number of graph nodes to powers of 2, and an arbitrary implementation of 2^n nodes requires $n+1$ qubits. However, it is possible to have any number of nodes, given that the proper correction is made as can be seen in Douglas and Wang (2009). The method used was *Gray Code Ordering* proposed by Slepoy (2006), whereby a certain arrangement of CNOT gates results in control states only differing by a single bit.



(a) Douglas wang increment



(b) Douglas wang Decrement

Figure 21: Douglas wang shift operator

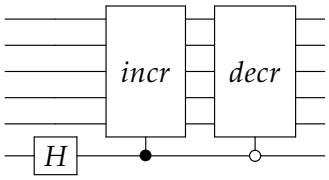


Figure 22: Douglas wang coined quantum walk circuit

In order to run this circuit on a real quantum computer using Qiskit, one must first find a way of creating generalized CNOT gates, since it is not available in the base package. One approach to this problem is to decompose an arbitrarily controlled CNOT gate into elementary gates, as was done by Barenco et al. (1995). In this context, the main idea is that for any unitary operator U, there exists operators such that

$$U = \phi AXBXC, \tag{86}$$

where $ABC = I$, X is the Pauli-X and ϕ is a phase operator described by $\phi = e^{i\delta} \times I$.

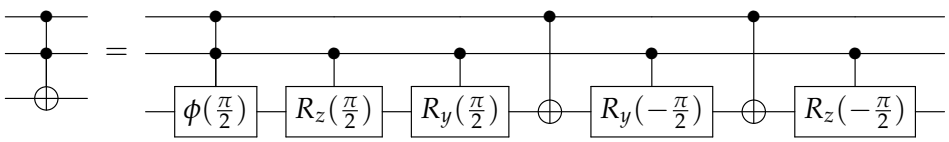


Figure 23: Toffoli decomposition

In order to understand this method, a good first example is the Toffoli gate, as is shown in figure 23. The first rotation in the circuit is defined by the R_z matrix

$$R_z(\theta) = \begin{pmatrix} e^{i\frac{\phi}{2}} & 0 \\ 0 & e^{i\frac{\phi}{2}} \end{pmatrix}, \tag{87}$$

where $\theta = \frac{\pi}{2}$. Secondly, the R_y rotation is

$$R_y(\phi) = \begin{pmatrix} cos(\frac{\theta}{2}) & -sin(\frac{\theta}{2}) \\ sin(\frac{\theta}{2}) & cos(\frac{\theta}{2}) \end{pmatrix}, \tag{88}$$

and $\phi = \frac{\pi}{2}$. The following rotations are simply $R_z^\dagger$ and $R_y^\dagger$. Lastly, the phase operator ϕ is

$$\phi(\delta) = \begin{pmatrix} e^{i\delta} & 0 \\ 0 & e^{i\delta} \end{pmatrix} \tag{89}$$

where $\delta = -\frac{\pi}{2}$. This phase correction is considered because otherwise

$$R_z(\frac{\pi}{2})R_y(\frac{\pi}{2})XR_y(-\frac{\pi}{2})XR_z(-\frac{\pi}{2}) = \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix} \neq X. \tag{90}$$

Introducing the phase correction results in

$$\phi(\frac{\pi}{2}) \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix} = \begin{pmatrix} i & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = X \tag{91}$$

However, since this is a global phase, it won't be included since it has no effect on the result of the measurement.

A more generalized version of this method can be seen in figure 24. Each individual generalized CNOT gate in this circuit can be expanded as was done for the Toffoli gate example, stopping once the generalised inverter gates are simply Toffoli gates.

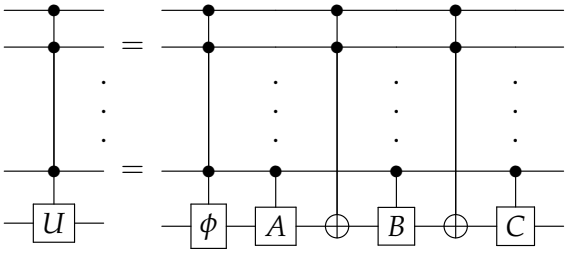


Figure 24: General decomposition

This was the chosen method because it provides a way of implementing arbitrarily controlled CNOT gates without the use of ancillary qubits, which are a scarce resource.

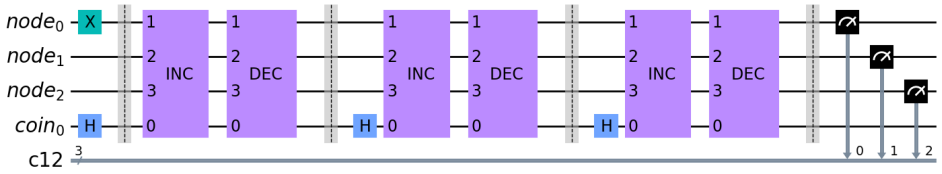


Figure 25: Temp

This circuit was implemented in Qiskit, as can be seen in figure 25. In this example, the increment and decrement sequence was applied three times on a graph of size $2^3 = 8$ nodes. The starting position of the walker was set to $\Psi(0) = |4\rangle$ and the Hadamard coin was used. The first block after the barrier is the sequence of operations that will increment the state of the walker, as is shown in figure 26. The circuit is simply the CNOT decomposition of figure 24 applied to the increment circuit of figure 21a for the $N = 4$, qubit case. The following

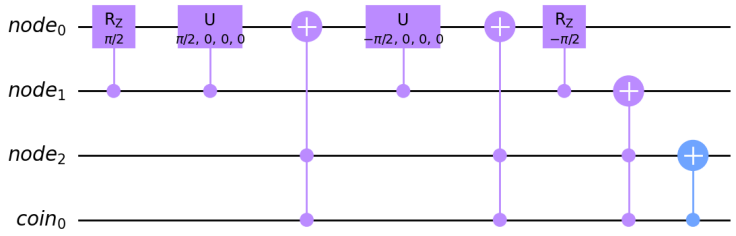


Figure 26: Temp

block represents the decrement of the state of the walker, which is just an increment block with it's controls negated as is shown in figure 27. The rest of the circuit is just the repetition of these operations as a function of the number of time steps required.

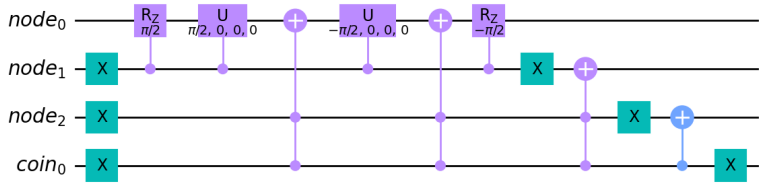


Figure 27: Temp

Lastly, the circuit is measured and the results can be seen in figure 28.These results can be verified by calculating the time evolution of the wave function associated with the system

$$|\Psi(0)\rangle = |4\rangle \tag{92}$$

$$|\Psi(1)\rangle = \frac{|0\rangle\,|x=3\rangle + |1\rangle\,|x=5\rangle}{\sqrt{2}} \tag{93}$$

$$|\Psi(2)\rangle = \frac{|0\rangle\,|x=2\rangle + |1\rangle\,|x=4\rangle + |0\rangle\,|x=4\rangle - |1\rangle\,|x=6\rangle}{2} \tag{94}$$

$$|\Psi(3)\rangle = \frac{|1\rangle\,|x=1\rangle - |0\rangle\,|x=3\rangle + 2(|0\rangle + |1\rangle)\,|x=5\rangle + |0\rangle\,|x=7\rangle}{2\sqrt{2}}. \tag{95}$$

Taking the modulus squared of the amplitudes associated with the states, confirms that the probability distribution presented in figure 28 is correct.

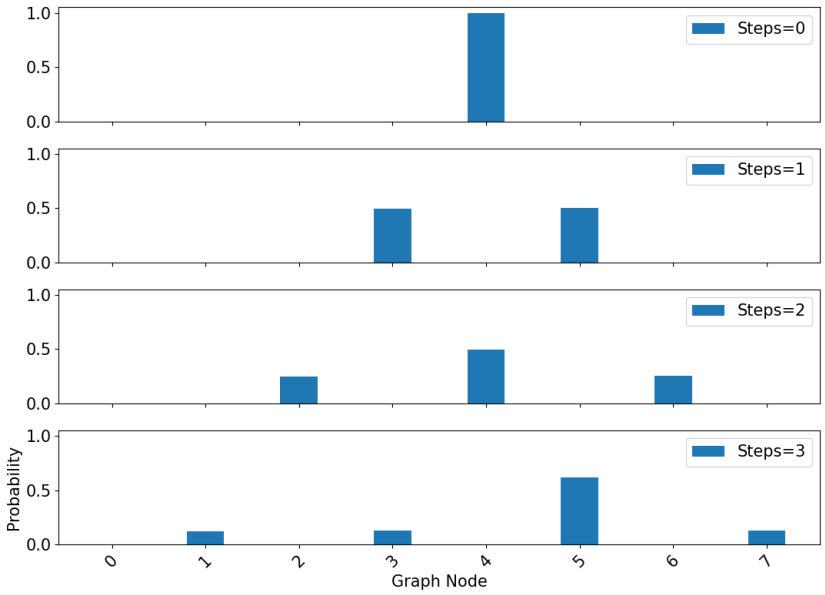


Figure 28: Temp

4.2 CONTINUOUS

As was seen in section 3.3, the unitary evolution operator of this model is defined as

$$U(t) = e^{-iHt} = e^{i(-\gamma L)t} = e^{-i\gamma(A+D)t}. \tag{96}$$

Considering a regular graph, this operator can be rewritten as

$$U(t) = \phi(t)e^{-i\gamma(A)t}, \tag{97}$$

where $\phi(t)$ is a global phase and A is the adjacency matrix associated with the graph.

Here, the study will focus on the circuit implementation of this walk in a cyclic graph, and a different approach was used to define the adjacency matrix. This approach relies on the concept of a circulant graph, which are a class of graphs defined by a circulanti matrix such that

$$A = \begin{pmatrix} c_0 & c_{N-1} & \cdots & c_3 & c_2 \\ c_1 & c_0 & c_{N-1} & & c_3 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{N-2} & & \ddots & \ddots & c_{N-1} \\ c_{N-1} & c_{N-2} & \cdots & c_1 & c_0 \end{pmatrix}. \tag{98}$$

In order to generate the proper circulant graphs, restrictions on this matrix are needed. Firstly, $c_0 = 0$, since self-loops are not part of the structure. Secondly, the matrix must be symmetric, therefore $c_{n-j} = c_j$.

These matrices can be fully described by the first column of the matrix

$$v_1 = [c_0, c_1, \cdots, c_{N-2}, c_{N-1}]^T \tag{99}$$

with a discrete convolution operator performing cyclic permutations of c, on each column. For example,

$$Dv_1 = [c_{N-1}, c_0, \cdots, c_{N-3}, c_{N-2}]^T = v_2. \tag{100}$$

More specifically, for the cycle case

$$Dv_1 = D[0,1,0,\cdots,0,1]^T = [1,0,1,0,\cdots,0,0]^T = v_2. \tag{101}$$

The operator can then be recursively defined as

$$Dv_k = v_{k+1}. \tag{102}$$

and the matrix will be

$$A = \sum_{j=0}^{N-1} Dv_j \tag{103}$$

$$A^k = \sum_{j=0}^{N-1} c_{(j-1) \mod N} \tag{104}$$

The eigenvalues of a circulant matrix can be found by

$$\lambda_p = c_0 + \sum_{q=1} c_{N-q}\omega^{pq}, \tag{105}$$

and the eigenvectors are

$$|\varphi_p\rangle = \frac{1}{\sqrt{n}} \sum_{q=0}^{n-1} \omega^{pq}. \tag{106}$$

This given, it is possible to construct an operator that diagonalizes the circulant matrix through the eigenvectors, which is useful for constructing the circuit. For this purpous, the Quantum Fourier Transform can be used and it is defined as

$$F = \frac{1}{\sqrt{N}} \sum_{p,q} \omega^{pq} |p\rangle \langle q| . \tag{107}$$

The adjacency matrix of a circulant graph can then be diagonalized such that

$$A = F^\dagger \Lambda F, \tag{108}$$

where Λ is a diagonal operator that encodes the eigenvalues

$$\Lambda = \sum_j \lambda_j |j\rangle \langle j| . \tag{109}$$

The unitary operator of the walk can then be rewritten as

$$U = F^\dagger e^{i\gamma\Lambda t} F \tag{110}$$

where

$$e^{i\gamma\Lambda t} = \sum_j e^{i\gamma\lambda_j t} |j\rangle \langle j| . \tag{111}$$

The circuit can now be constructed making use of the *diagonal* function provided by Qiskit, which decomposes diagonal operators based on the method presented in theorem 7 of Shende et al. (2006). The other tool used was the Quantum Fourier Transform, also provided by the Qiskit package. Figure 29 shows the implementation of the circuit for 3 qubits or $2^3 = 8$ graph nodes and $t = 3$.

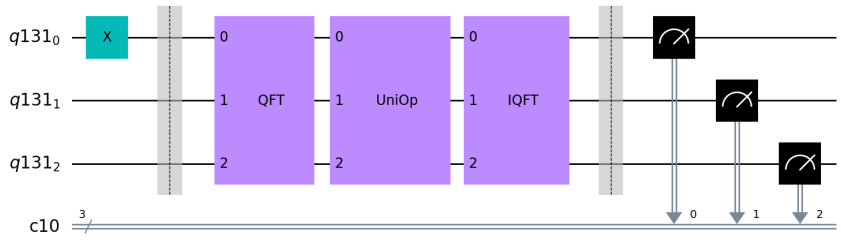


Figure 29: Temp

The circuit for the Quantum Fourier transform is well known and presented in figure 30. The circuit associated with $F^{\dagger}$ is similarly constructed by negating the previous figure's rotations.

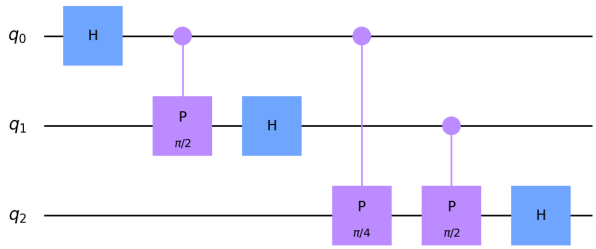


Figure 30: Temp

The circuit associated with the diagonal operator is show in figure 31. Furthermore equation 110 says that time is simply a constant inside the exponential, which means that the diagonal operator's circuit will not need extra gates when increasing time, it will only need different rotations and differ in global phase. This is an advantage when comparing to

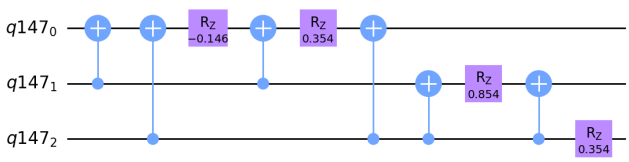


Figure 31: temp

the previous model, seen in figure 25, where each extra step required another increment

and decrement gates. Further evidence of this behaviour can be seen in figure 32, where it was created a circuit for $t = 0$ up to $t = 100$ with increments of 1. It was then counted the number of gates, for each circuit, and plotted against the respective time. This graph

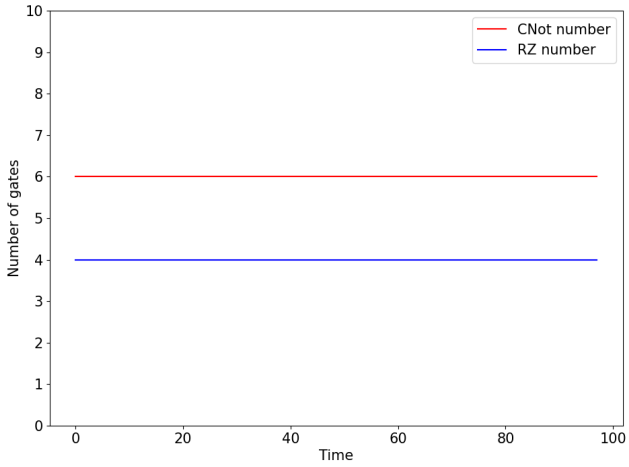


Figure 32: Temp

clearly shows that both the number of R_z rotations and CNot operations remain constant throughout the entire time interval.

Finally, the circuit was measured, and the result can be seen in figure 33

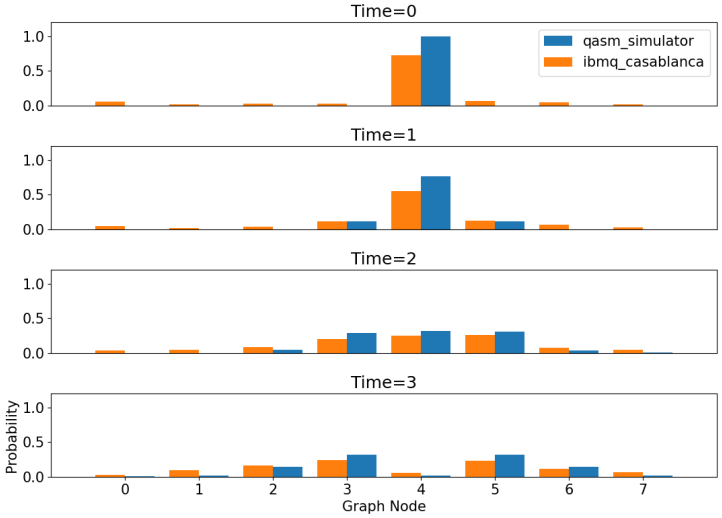


Figure 33: Temp

4.3 STAGGERED

4.4 SEARCH PROBLEMS WITH QISKIT

4.4.1 *Grover*

Like it was seen in section 2.1 Grover's algorithm is a tool that provides quadratic speedups to unstructured search problems. The routine consists on applying an oracle

$$\mathcal{O}\left|x\right\rangle = (-1)^{f(x)}\left|x\right\rangle, \tag{112}$$

that flips the solution states, followed by an amplitude amplification process by means of the diffusion operator

$$\mathcal{D} = (2\left|\Psi_0\right\rangle\left\langle\Psi_0\right| - I) = H^{\otimes n}(2\left|0\right\rangle\left\langle 0\right| - I)H^{\otimes n}, \tag{113}$$

where $\left|\Psi_0\right\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}\left|x\right\rangle$. The unitary operator that describes the algorithm will then be

$$\mathcal{U} = \mathcal{D}\mathcal{O}. \tag{114}$$

As was before mentioned, this evolution will be done several times, depending on the number of elements. Optimal probability of success finding a single solution will be reached after approximately $\sqrt{N}$ steps, and $\sqrt{\frac{N}{K}}$ for K solutions. Consider the 3 qubit case, where $N = 8$, with 3 iterations, as shown in figure 34.

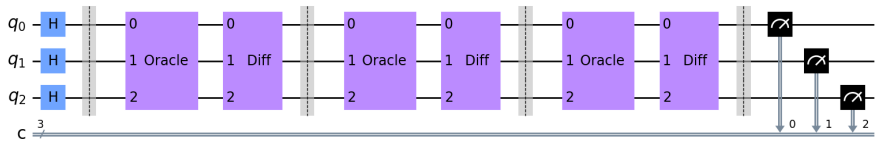


Figure 34: Temp

The oracle circuit is shown in figure 35.
The diffusion circuit is shown in figure 36
The results of measurement are shown in figure 37

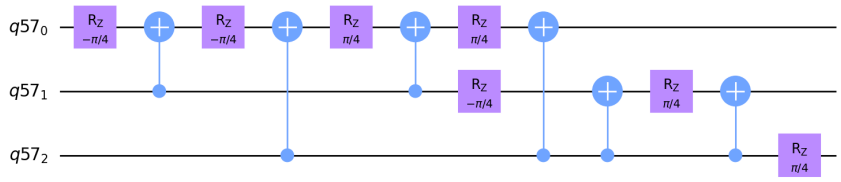


Figure 35: Temp

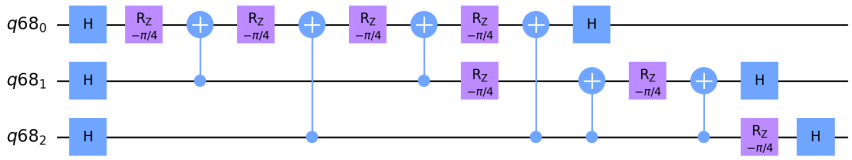


Figure 36: Temp

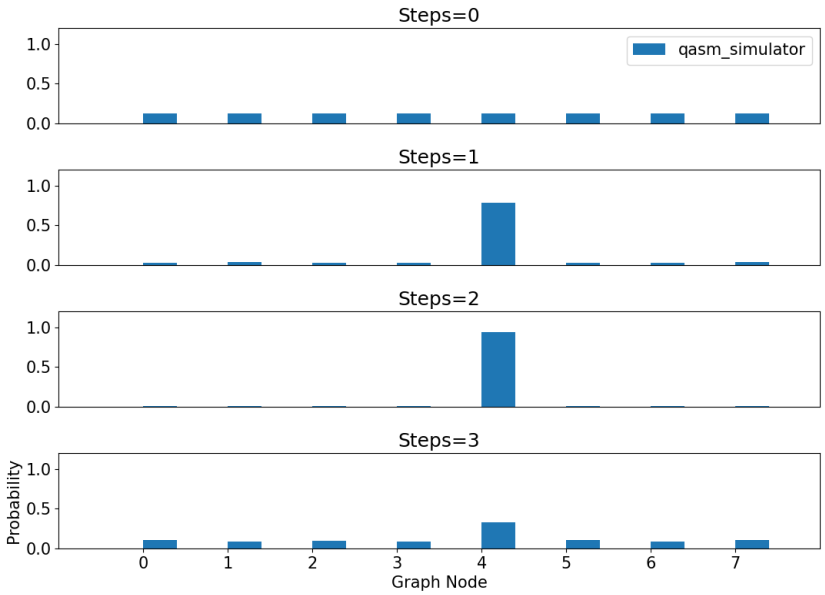


Figure 37: Temp

4.4.2 *Coined*

4.4.3 *Continuous*

4.4.4 *Staggered*

5

DISCUSSIONS AND CONCLUSION

5.1 CONCLUSIONS

5.2 PROSPECT FOR FUTURE WORK

BIBLIOGRAPHY

Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007.

Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, Nov 1995. ISSN 1094-1622. doi: 10.1103/physreva.52.3457. URL http://dx.doi.org/10.1103/PhysRevA.52.3457.

Michael Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4-5):493–505, 1998.

Andrew M. Childs and Jeffrey Goldstone. Spatial search by quantum walk. *Physical Review A*, 70(2):022314, 2004.

Andrew M. Childs, Richard Cleve, Enrico Deott, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by quantum walk. *Proc. 35th ACM Symposium on Theory of Computing (STOC 2003), pp. 59-68*, 2002. doi: 10.1145/780542.780552.

B.L. Douglas and J. B. Wang. Efficient quantum circuit implementation of quantum walks. 2009.

Steven Finch. "pólya's random walk constant." in mathematical constants. *Cambridge University Press*, pages 322–331, 2003.

Lov K. Grover. A fast quantum mechanical algorithm for database search. *STOC '96 Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

Elliott Montroll. Random walks in multidimensional spaces, especially on periodic lattices. *Journal of the Society for Industrial and Applied Mathematics*, 4(4):241–260, 1956. doi: 10.1137/0104014.

Elliott Waters Montroll and George Herbert Weiss. Random walks on lattices. ii. *Journal of Mathematical Physics*, page 167–181, 1997.

Karl Pearson. The problem of the random walk. *Nature*, 72(1865):294, 1905. doi: 10.1038/072294b0.

Renato Portugal. *Quantum Walks and Search Algorithms*. Springer, 2018.

Renato Portugal, M. C. de Oliveira, and J. K. Moqadam. Staggered quantum walks with hamiltonians. *Physical Review A*, 95(1):012328, 2017.

George Pólya. Über eine aufgabe der wahrscheinlichkeitsrechnung betreffend die irrfahrt im straßennetz. *Mathematische Annalen*, 84:149–160, 1921. doi: 10.1007/BF01458701.

V.V. Shende, S.S. Bullock, and I.L. Markov. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010, Jun 2006. ISSN 1937-4151. doi: 10.1109/tcad.2005.855930. URL http://dx.doi.org/10.1109/TCAD.2005.855930.

Alexander Slepoy. Quantum gate decompostition algorithms. *Sandia National Laboratories*, 2006.

Tommi Sottinen. Fractional brownian motion, random walks and binary market models. *Finance and Stochastics*, (5):343–355, 2001. doi: 10.1007/PL00013536.

Salvador Elías Venegas-Andraca. Quantum walks: a comprehensive review. *Quantum Information Processing*, 11(5):1015–1106, 2012.

Christof Zalka. Grover's quantum searching algorithm is optimal. *Physical Review A*, 60(4): 2746–2751, 1999.

A

SUPPORT MATERIAL