The effect of measurement uncertainties on quantum random walk based problems

Ronan den Ouden
(5300444)

Delft University of Technology
Bachelor of Applied Physics and Applied Mathematics
Supervisors: Dr. J.L.A. Dubbeldam and Dr. T.H. Taminiau
June 15, 2023

1 Abstract

Random walks have been used in a number of different fields for a long time. A lot of applications have been made more efficient with the rise of the quantum random walk. Recently there have been a lot of improvements in the realization of these quantum random walks in real world systems. In this report we will look into the effect of uncertainties in the measurement time and frequency on three problems. These problems are the network centrality problem, the graph isomorphism problem and the spatial search problem.

This report will start by looking at the behaviour of both the classical and quantum random walk in discrete time on both the integer number line and more complex networks. After which we will expand this theory to the continuous time case. Then we will look at how measurements are performed on the quantum random walk and look at which type of measurements exist. This theory will then be applied to the detection problem where we look at the probability of finding a specific state in a network. Following this theory we will explain the methods that are used to simulate uncertainty in the measurement time and frequency.

With this information we will look at the three problems, for each of the problems we will study their theoretical behaviour. After that we will simulate the behaviour of the quantum random walk with different levels of uncertainty. Following this we will compare these simulations to the theoretical behaviour to determine the effect of the measurement uncertainty on the outcome of these problems. To determine the effect we study both the pattern between the error in the result for different levels of uncertainty and the severity of the error.

We have found that there is a linear pattern between the uncertainty in the measurement time and frequency and the error of the system for small levels of uncertainty for each of the three problems, however this pattern is not present for larger levels of uncertainty. For the network centrality problem we found that the error in the simulations is still relatively small for high levels of uncertainty, for the spatial search problem we found that the error in the simulations is a bit higher than for the network centrality problem but is still relatively small. However for the graph isomorphism problem we have found that the error in the simulations is very large for high levels of uncertainty and still relatively big for smaller levels of uncertainty.

With these results we are able to conclude that both the network centrality and the spatial search problem are able to be realized with a relatively high level of uncertainty in the measurement time and frequency while the graph isomorphism problem will be almost impossible to solve without very small levels of uncertainty.

Contents

2 Introduction

Random walks are used as a method to solve a number of problems and can be used to create simplified models of certain processes in physics. A random walk is a simple concept which describes the path a particle takes that consists of a number of random steps. Some well known applications of random walks are Google page rank [1], image processing [2] and cryptography [3]. In the search for even better and faster methods to implement the random walk Aharonov et al. [4] came up with the quantum random walk. This quantum random walk has shown to be a significant improvement over the classical random walk in a number of fields such as the hitting time in a graph [5] and search algorithms [6].

In this report we will look at three problems in which classical random walks are used that can be improved and sped up with the use of quantum random walks. The first problem that we will look at is the network centrality problem, in this problem we look at which points in a network have the biggest influence on the network. An application of network centrality is the determination of the busiest point in a transport network like the railway network. The second problem concerns graph isomorphisms, in this problem we compare multiple graphs to determine if they are identical. An application for the graph isomorphism problem for example is the finger scan for your phone. The final problem is the spatial search problem in which we try to find a specific point in a network in the least amount of time. This problem is used for finding search results in a search engine.

The application of quantum random walks in these three problems is currently being tested in practice on different real world systems [7]. In practice it is currently not possible to fully achieve the theoretically calculated results. This is due to the uncertainties in the systems that are used to realize this quantum random walk.

In this report we will look at the effect that uncertainty in the measurement time and frequency has on the results relative to the theoretical results, within these three problems.

This report will start by introducing the theory of the classical random walk in discrete time after which this concept will be expanded to the quantum random walk in discrete time. Following this introduction is the extension of both the classical and the quantum random walk to their continuous time counterparts. With the information of how these walks operate we will then look at how measurements are defined and what type of measurements we can have in a system which we will further expand to the detection problem. Section 4 will then follow by explaining the method that was used to study the effects of uncertainty in general. With the method explained, section 5 will go further into the details on the network centrality problem. Covering the extra theory needed for this problem and the results from simulations with an uncertainty. Following this, section 6 and 7 will follow the same structure as the network centrality problem for both the graph isomorphism problem and the spatial search problem. With all the theory and results covered we will then look at how these systems work in practice in section 8. After which we will discuss the results and possible options for further research. Finally we will give the conclusion is section 10.

3 Theory

In this theory the spin states $|\uparrow\rangle$ and $|\downarrow\rangle$ are eigenstates of the S_z basis and can be written in the following vector notation:

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

3.1 Discrete time

3.1.1 Classical random walk

The classical random walk (CRW) is a random process where a particle walks on a path consisting of random steps on a specified space. The most common random walk is the random walk on the integer number line. We will start by looking at how the CRW behaves on the integer number line. For the CRW on the integer line a particle is placed at position 0 and the particle can either set one step to the left to position -1 or set one step to the right to position $+1$. The particle has a certain probability to step left and a certain probability to set a step to the right, these probabilities are usualy set to be $\frac{1}{2}$. When looking at this specific CRW we can see that there are more paths that a particle can take to stay near the starting position than ending at the edges. This can be seen as a coin toss, where there is only one possible way to end with all tosses as head but there are N possibilities to throw head only once and having tails as the other outcomes when we throw the coin N times. It is also important to note that the particle can only be on even positions on an even number of throws and only on odd positions for an odd number of coin tosses. In the following table this is illustrated with the number of possible paths to get to a certain position for the first 5 time steps.

Table 1: Number of possible paths to end at specific positions on the integer line after the first 5 time steps. For a random walk with equal probability to move left and right. Blank spaces mean that there is no path to that position at that time.

position	-5	-4	-3	-2	-1	0	1	2	3	4	5
0 steps						1					
1 step					1		1				
2 steps				1		2		1			
3 steps			1		3		3		1		
4 steps		1		4		6		4		1	
5 steps	1		5		10		10		5		1

It can be seen that this corresponds with Pascal's triangle, this gives us the ability to calculate the probability of the particle ending at a specific position after N time steps for the CRW on the integer line. The equation for this probability is the following

$$P(n) = \frac{1}{2^N}\binom{N}{\frac{1}{2}(N+n)} \tag{2}$$

Figure 1 shows this probability distribution together with the results of 10000 simulations added together and normalized.

It is clearly visible that the distribution of the CRW on the integer line represents a binomial distribution which is centered around 0 that converges to a normal distribution. This is indeed the case as can be shown with the expectation value and the variance of the classical random walk which are calculated as follows

$$E(S_N) = \sum_{i=1}^{N} E(C_i) = 0 \tag{3}$$

$$E(S_N^2) = \sum_{i=1}^{N} E(C_i^2) + 2 \cdot \sum_{1\leq i<j\leq N} E(C_iC_j) = N + 0 = N \tag{4}$$

Here S_N is the position of the walk after N time steps and C_i are independent identically distributed random variables with probability $\frac{1}{2}$ to be either 1 or -1. With these results it is clear that the CRW

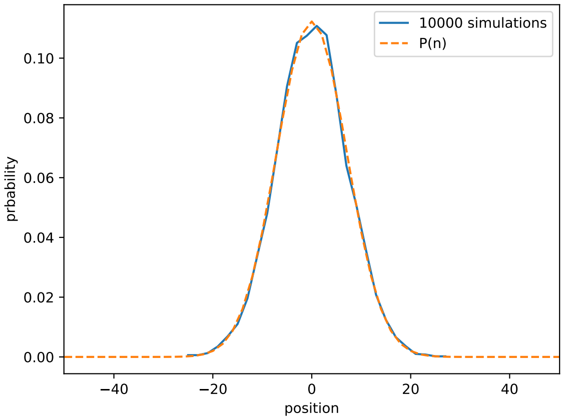


Figure 1: Plot of 10000 simulations of a classical random walk with equal probability to walk left and right and the binomial distribution given by equation 2 after 51 time steps. In this figure it was chosen to not show the probability of the even points because they are always 0 after an odd number of steps.

on the integer line can be described with a normal distribution with standard deviation $\sigma = \sqrt{N}$ where N is the amount of time steps.

With this understanding of the CRW on the integer number line it is possible to extend the CRW to different and more complex spaces. One of these extensions is a ring of finite size. This is highly similar to the CRW on a line; in this extension a periodicity is introduced. An example of this periodicity is that the position 0 on the line can move to either 1 or N - 1 where N is the size of the ring, similarily position N - 1 can move to either N - 2 or 0. An important thing to notice is that the behavior of this CRW depends on if N is even or odd. This is due to the CRW only being on even or only being on odd numbers on the number line. If the ring has an even number of positions then this property of the CRW still holds but if the ring has an odd number of positions then the CRW will eventually be at each position on the ring with a nonzero probability. This is due to the fact that the CRW normally only jumps to a position which has zero probability for a particle to be there and the current position becoming zero after the step. But in the case of the odd ring the particle will jump to a position with nonzero probability of a particle being there and this also happens for the other position resulting in both the positions staying at a nonzero probability of the particle being there. This behaviour will be shown in figure 3.

This CRW can even be extended to walk on any arbitrary graph which also covers the case of the line and the ring. In this case the particle can walk from a vertex to any of the vertices that are connected to the current position, each of these transitions can be given a probability, however the probability is usually set the same for each edge which results in the probability being $\frac{1}{d_i}$ where d_i is the degree of vertex i. For the CRW on a graph the process is described by a Markov chain which is given by the following formula [8]

$$\vec{p}_n = M\vec{p}_{n-1} \tag{5}$$

Here $\vec{p}_n$ is the probability distribution over the vertices after n time steps and M is a matrix that describes the transition of the CRW over the graph which is constructed in the following way

$$M_{i,j} = \begin{cases} \frac{1}{d_i} & i \neq j \text{ and } i \text{ and } j \text{ connected} \\ 0 & \text{else} \end{cases} \tag{6}$$

With these equations it is easy to see that the probability density of a particle on a graph after n time steps is given by $\vec{p}_n = M^n\vec{p}_0$. In section 3.2.1 this concept will be expanded for the continuous-time random walk.

3.1.2 Quantum random walk

This subsection follows the theory of J. Kempe [9] for the construction of the quantum random walk on a line and a graph.

Just as before with the classical random walk we will start by looking at a particle that walks on a line. This particle is described by a wave-packet $|\psi\rangle$, this wave-packet has two degrees of freedom. These degrees of freedom are the spin of this particle and the position. These degrees of freedom are linked by a tensor product which allows us to write the wave-packet in the following form $|\psi\rangle = \left|\psi^{\uparrow\downarrow}\right\rangle \otimes |\psi_x\rangle$. The spin of this particle is a superposition of the eigenstates of the S_z operator and the position is an eigenstate of the position operator. This results in the following equation for the initial state of the particle

$$|\psi\rangle = \sum_{x\in\mathbb{Z}} (a_x^{\uparrow} |\uparrow\rangle + a_x^{\downarrow} |\downarrow\rangle) \otimes |x\rangle \tag{7}$$

Here the spin state is normalized which means that $\sum_{x\in\mathbb{Z}}(|a_x^{\uparrow}|^2 + |a_x^{\downarrow}|^2) = 1$ such that the probability of the particle being anywhere stays 1. Similarly to the classical random walk there is an operation which determines the evolution of the state of the particle. For the quantum random walk (QRW) this operator is built up from two operations, the first one is commonly referred to as the coin of the system and the second is the translation of the position. The coin operator works on the spin state of the system, this operator describes the transition of the spin state. The coins that can be used for the quantum random walk can be constructed in different ways, however the coin does need to be unitary in order to keep the normalization of the spin state. The most common coin is the Hadamard coin H

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{8}$$

The transition of the position of the particle is based on the spin of the particle. If a particle has spin up, then the position will move one unit to the right and if the particle has spin down, it will move one unit to the left. This operator can be written as follows:

$$S = |\uparrow\rangle\langle\uparrow| \otimes \sum_{i\in\mathbb{Z}} |i+1\rangle\langle i| + |\downarrow\rangle\langle\downarrow| \otimes \sum_{i\in\mathbb{Z}} |i-1\rangle\langle i| \tag{9}$$

The transition from one state to the next is described first by applying the coin operator to the spin state of the particle after which the position is changed by the shift operator S. This total transition is described by the operator U

$$U = S \cdot (C \otimes I) \tag{10}$$

Here C is the coin operator, I is the identity operator and S is the shift operator described in Eq 9. Figure 2 shows the probability distribution of a particle after 51 steps of the QRW with the Hadamard coin. It is clear that the particle undergoing the QRW propagates much further than a particle undergoing a CRW which is shown in figure 1.

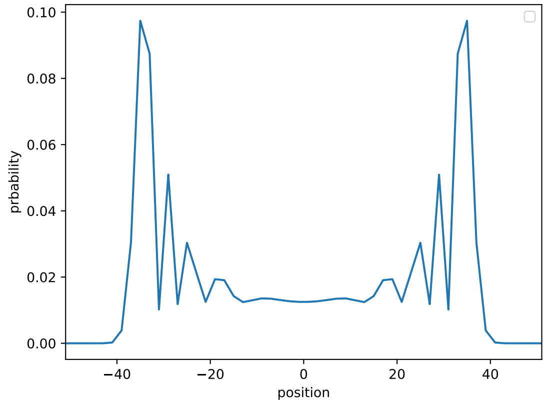
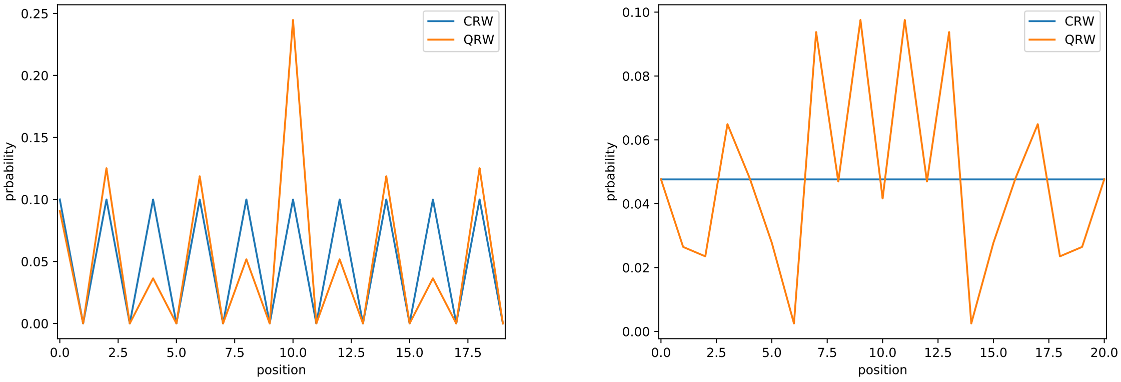


Figure 2: Discrete quantum random walk after 51 time steps with the Hadamard coin described in equation 8. For this figure it was chosen to not show the probabilities of the even positions because these are all 0 on an odd number of time steps.

Similarly to the classical random walk the quantum random walk can also be extended to different networks. One such network is the quantum random walk on a cicle of finite size. For the quantum random walk on a cycle a periodicity is introduced in the position state of a particle which states that $(a^{\uparrow}\left|\uparrow\right\rangle + b^{\downarrow}\left|\downarrow\right\rangle) \otimes \left|0\right\rangle = (a^{\uparrow}\left|\uparrow\right\rangle + b^{\downarrow}\left|\downarrow\right\rangle) \otimes \left|L\right\rangle$ where $a^{\uparrow}$ and $b^{\downarrow}$ are arbitrary normalization constants and L is the size of the cycle. This seems similar to the CRW on a cycle but there is one big difference. Namely the CRW on the cycle will eventually reach an equilibrium position where each position on the cycle has the same probability of a particle ending there, but in the case of the quantum random walk no equilibrium is ever achieved. Just like the case of the classical random walk it is important to note that there is a difference in cycles of even length and cycles of odd length. In the case that the cycle has even length there will always be positions with zero probability of finding a particle at that position and in the case of and odd length all positions will have nonzero probability of particle being there for times larger than L. Figure 3 shows the probability distribution for both a CRW and a QRW on both an even and an odd cycle after a large number of times. The coin used for the quantum walk in this figure is the Hadamard coin and for the classical walk the probability of going left and going right is equal.



(a) Classical and quantum random walk on a cycle of size 20 after 1750 steps. It is clear that the classical random walk has converged to a uniform distribution over the attainable positions and the quantum random walk has not.

(b) Classical and quantum random walk on a cycle of size 21 after 1750 steps. It is clear that the classical random walk has converged to a uniform distribution over the attainable positions and the quantum random walk has not.

Figure 3: Classical and quantum random walks on an even cycle and on an odd cycle for 1750 time steps.

This QRW can be even further generalized by extending it to walk on a graph. For the QRW on a graph we will first look at graphs which are d-regular which means that each vertex has degree d. Which results in a d dimensional coin space. The coin space is the equivalence of the spin space for higher dimensional coins. Each edge out of a vertex will be labeled 1 through d. Index i with edge e_v^i being the i'th edge going out of vertex v. The state of a particle on a graph is given by the following equation

$$\left|\psi\right\rangle = \sum_{v=1}^{N} \left(\sum_{i=1}^{d} a_v^i \left|i\right\rangle\right) \otimes \left|v\right\rangle \tag{11}$$

Here N is the number of vertices, $\left|v\right\rangle$ denotes the vertex and $\left|i\right\rangle$ denotes the coin state of the particle which corresponds with the spin state in equation 7 on a line. Similarly to the quantum random walk on a line or cycle the particle first undergoes an operation in the coin space, in this case the coin-operator is a d-dimensional unitary operator. After the coin-flip, the position of the particle is changed by the shift operator which follows the following equation

$$S\left|i\right\rangle \otimes \left|v\right\rangle = \begin{cases} \left|i\right\rangle \otimes \left|w\right\rangle & \text{if } e_v^i = (v, w) \\ 0 & \text{else} \end{cases} \tag{12}$$

With this shift operator and a coin operator with dimension d it is possible to generate a QRW on each d-regular graph. To extend this QRW to general graphs we take the dimension of the coin space to equal the maximum degree of the vertices and introduce self-loops on each vertex with a degree that is smaller than this maximum. For example if our graph has a maximum degree of 4 we put two extra edges on each vertex with degree 2 and one extra edge on a vertex with degree 3. An alternative for changing the graph to become a d-regular graph is to introduce a coin operator which changes depending on the vertex and is an unitary operator with dimension equal to the degree of the corresponding vertex.

3.2 Continuous time

3.2.1 Classical random walk

The classical continuous-time random walk (CTRW) can be seen as a generalization of the discrete random walk discussed in section 3.1.1. In the CTRW the particles can jump on random time intervals instead of jumping on a set time interval. The CTRW can be seen as a special case of a continuous time classical Markov chain. In this section we will only focus on CTRW that work on graphs. Just like the CRW particles are localized on a vertex and the edges on the graph denote the possible transitions from one position to another. When looking at a graph with a vertex set V the possible positions are given by the probability distribution $\vec{p} = (p_1, p_2, \ldots, p_{|V|})^T$. This walk transforms due to the matrix H. This matrix is given by the Laplacian matrix multiplied with a factor γ which is given by the following equation [9]

$$H_{i,j} = \begin{cases} -\gamma & i \neq j \text{ and } i \text{ and } j \text{ connected} \\ 0 & i \neq j \text{ and } i \text{ and } j \text{ not connected} \\ d_i\gamma & i = j \end{cases} \tag{13}$$

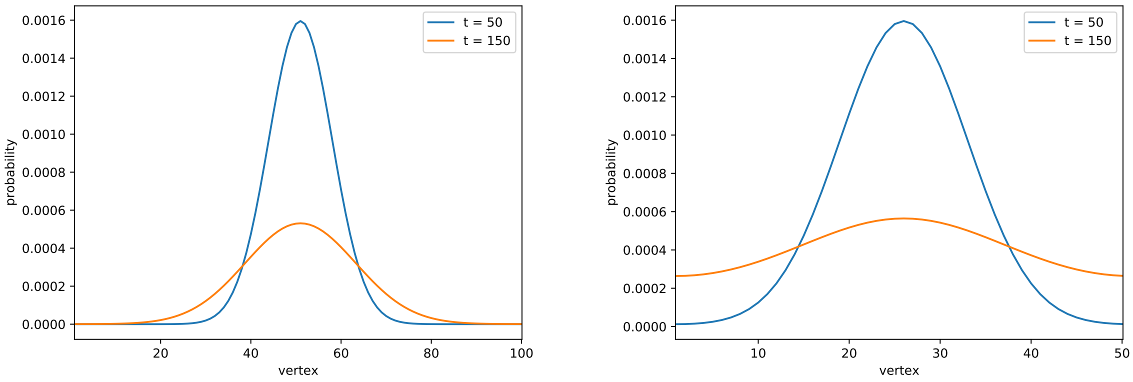
The γ in this fixed constant which changes the speed of evolution, this constant is the probability for a transition to occur per unit time. the d_i in this equation is the degree of vertex i. Then the evolution process of the probability of a particle being in a single vertex p_i is given by the following equation

$$\frac{dp_i(t)}{dt} = -\sum_j H_{i.j} p_j(t) \tag{14}$$

This equation can be written in the following form

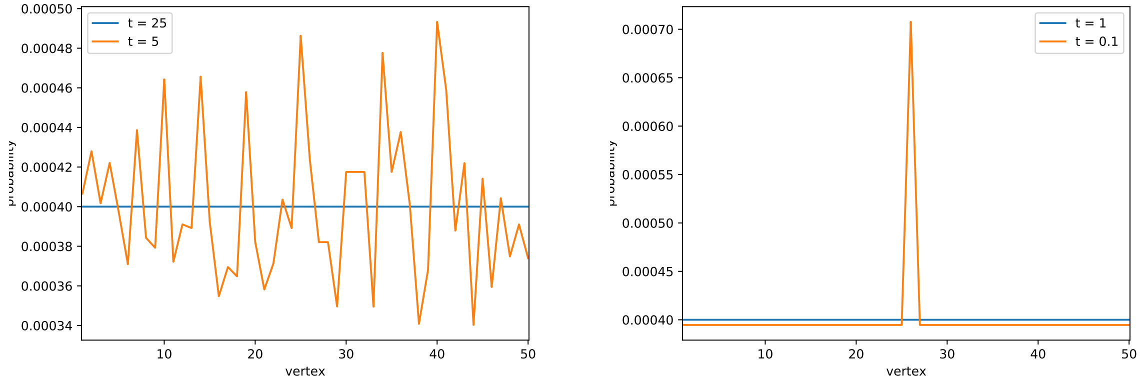
$$\frac{d\vec{p}(t)}{dt} = -H\vec{p}(t) \tag{15}$$

It can be seen that the solution of this differential equation is given by $\vec{p}(t) = e^{-Ht}\vec{p}(0)$. Figure 4 shows a few of these distributions for different types of networks on different times. In these figures it is clear that the CTRW converges to a uniform distribution after some time.



(a) Continuous-time classical random walk on a line graph with 100 vertices starting in vertex 50. The γ in equation 13 is set to 1 for this figure. The graph is shown in figure 23 in appendix A.3.

(b) Continuous-time classical random walk on a cycle graph with 50 vertices starting in vertex 25. The γ in equation 13 is set to 1 for this figure. The graph is shown in figure 24 in appendix A.3.



(c) Continuous-time classical random walk on a Barabasi-Albert(50,2) graph starting in vertex 1. The γ in equation 13 is set to 1 for this figure. The graph is shown in figure 25 in appendix A.3.

(d) Continuous-time classical random walk on a complete graph with 50 vertices starting in vertex 25. The γ in equation 13 is set to 1 for this figure. The graph is shown in figure 26 in appendix A.3.

Figure 4: Continuous-time classical random walks for 4 different types of network for two different evolution times each. The graphs that were used are shown in appendix A.3.

3.2.2 Quantum random walk

The continuous-time quantum walk (CTQW) is highly similar to the CTRW in the way it is constructed. The CTQW uses the Schrödinger equation instead of the differential equation given by equation 15. This idea was introduced by Farhi and Gutmann [10]. In the Schrödinger equation the Hamiltonian is the same matrix that was used for the CTRW given by equation 13. The Schrödinger equation is given by the following equation

$$i\hbar\frac{d}{dt}\left|\psi(t)\right\rangle = H\left|\psi(t)\right\rangle \tag{16}$$

For simplicity $\hbar$ is set to equal 1, this leads to the following solution of the Schrödinger equation

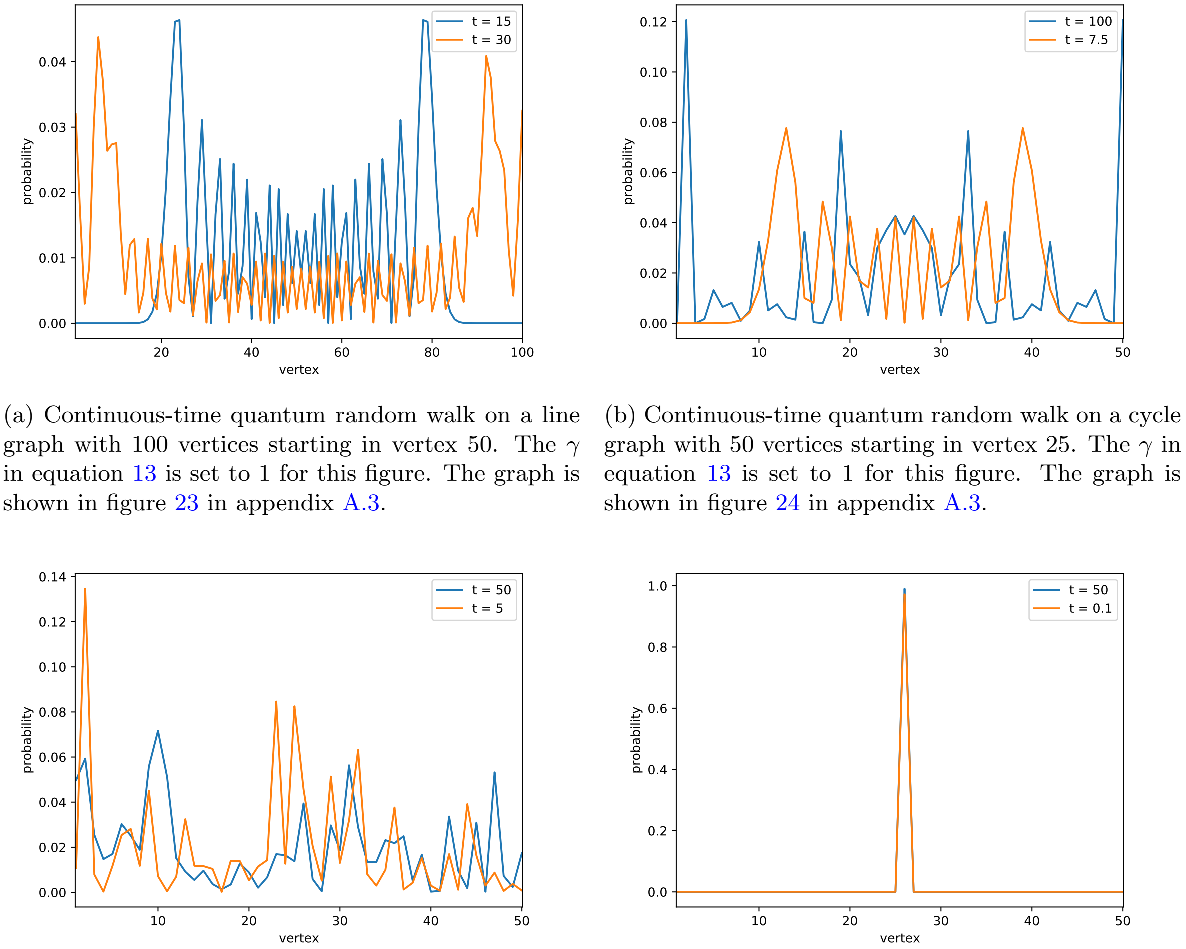
$$\left|\psi(t)\right\rangle = e^{-iHt}\left|\psi(0)\right\rangle \tag{17}$$

Here e^{-iHt} is the unitary evolution operator which will be denoted as $U(t)$ from now on. With this evolution operator one can determine the probability of the particle being at a specific position in the system after an arbitrary amount of time. The probability of the particle being at a specific position, v, after a time t is given by the following formula

$$P_v^{|\psi_0\rangle}(t) = |\left\langle v\right| U(t)\left|\psi_0\right\rangle|^2 \tag{18}$$

In this equation $|\psi_0\rangle$ is the starting position we calculate the overlap between $|v\rangle$ and $|\psi_0\rangle$. This equation can also be used to calculate the probability to be in a state that is a superposition of multiple vertices, in this case v will be changed for the superposition of vertices. Figure 5 shows the probability distribution of the particle for different graphs at different times.

When comparing the CTQW in figure 5 with the CTRW in figure 4 it is clear that the CTQW reaches vertices that lie further away from the starting position in a shorter time than the CTRW and that no equilibrium position is reached for the CTQW in contrast to the CTRW. However for the Complete graph there is almost no evolution, there is only a slight fluctuation in the height of the peak at the starting position in time.



(a) Continuous-time quantum random walk on a line graph with 100 vertices starting in vertex 50. The γ in equation 13 is set to 1 for this figure. The graph is shown in figure 23 in appendix A.3.

(b) Continuous-time quantum random walk on a cycle graph with 50 vertices starting in vertex 25. The γ in equation 13 is set to 1 for this figure. The graph is shown in figure 24 in appendix A.3.

(c) Continuous-time quantum random walk on a Barabasi-Albert(50,2) graph starting in vertex 1. The γ in equation 13 is set to 1 for this figure. The graph is shown in figure 25 in appendix A.3.

(d) Continuous-time quantum random walk on a complete graph with 50 vertices starting in vertex 25. The γ in equation 13 is set to 1 for this figure. The graph is shown in figure 26 in appendix A.3.

Figure 5: Continuous-time quantum random walks for 4 different types of network for two different evolution times each. The graphs that were used are shown in appendix A.3.

3.3 Measurements

In quantum mechanics the topic of measurements is very important. A measurement causes the wavefunction of a particle to collapse to a different wavefunction. Depending on the type of measurement the wave function either fully collapses to a single state or collapses to a state that is still a superposition. For measurements on a QRW we can either measure the position state or the spin state of a particle, in the case of a CTQW particles only have a position state. For this report only measurements on the position state will be discussed. For measurements of the position state of a particle there are two different types of measurements, these two types will be discussed in this subsection for both the discrete- and continuous-time quantum walks. These two types of measurements are called local and global measurements.

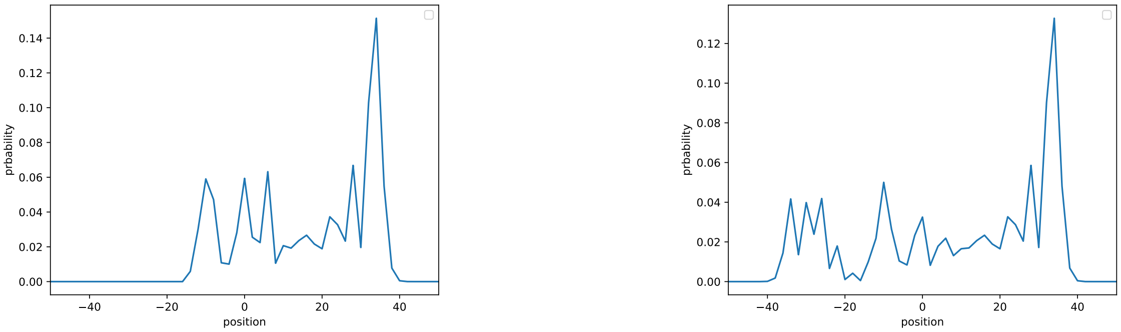
3.3.1 Local measurements

Local measurements are measurements that only measure in a specific position on the walk or measure a superposition of a few position, however this superposition is not allowed to cover the entirety of the space on which the quantum walk is operating. A measurement of this type can be seen as a way to check if the particle is actually at the position, so there are two outcomes of these measurements: either the particle is located at the measurement position or it is not there. In the first case the wavefunction collapses to a state that is localized in the position of the measurement and if it was not in this position it will collapse to a wavefunction that can be seen as the complement of the position of the measurement. This results in the following equation that gives the wavefunction of the particle after measurement [9]

$$M_v \left|\psi\right\rangle = \begin{cases} \left|v\right\rangle & p_v = \mid \left\langle v|\psi\right\rangle \mid^2 \\ \frac{\left|\psi\right\rangle - \left\langle v|\psi\right\rangle \left|v\right\rangle}{\sqrt{1-\mid\left\langle v|\psi\right\rangle\mid^2}} & p_{v\perp} = 1 - \mid \left\langle v|\psi\right\rangle \mid^2 \end{cases} \tag{19}$$

Here M_v is the measurement operator that performs a local measurement at position v, $\left|\psi\right\rangle$ is the wavefunction of the particle on which the measurement is performed, p_v is the probability of the particle being at position v and $p_{v\perp}$ is the probability of the particle not being at position v. A similar aproach is taken when the measurement is performed on a superposition instead of a single position state, the main difference is that the normalization is more difficult in this case since the probabilities of the individual states of the positions which are in the superposition have to be taken into account. Now that the collapsing of the wavefunction is explained it is also important to decide when to measure, this is different for the discrete and continuous walks. The discrete case will be discussed first.

For the QRW the particles always take a whole step each time the evolution operator is applied to the wavefunction. The measurement operator can either operate before or after the evolution operator, but this does not give different results due to the fact that these operations will alternate each other which means that there will only be a difference in the first and last step. It is also important to note that the frequency of the measurements is also important. An example of this is a QRW on the integer line which has a measurement at position v. If the measurement is performed after each time step it will be impossible to go beyond this position and still be centered around the starting point. This is due to the fact that either the particle is not found at position v which results in the wavefunction collapsing to a wavefunction which is still only on one side of v, this process will repeat until there the particle will be found in v. Eventually the particle can be in v which means that the walk will now be centered at v and starts its quantum walk at this position and this will restart each time the particle is at v. However when the measurement is not performed each step there is a chance that it will be at v when there is no measurement which means that next evolution step it will be possible to be on the other side of v compared to the starting position. When the particle is eventually found in v in this case the walk will be similar to the walk in the case where we measure each step. Figure 6 shows the QRW on the integer line with a measurement at position -15 after 50 time steps for two different measurement frequencies. The behaviour explained above is clearly visible in both of the figures, for both figures it was chosen to only show the cases for which no detection has taken place since the other case would be a normal quantum walk as in figure 2 centered around -15.



(a) Discrete time quantum random walk that performs via the Hadamard coin and has a measurement at position -15 after each step for 50 steps.

(b) Discrete time quantum random walk that performs via the Hadamard coin and has a measurement at position -15 after each third step for 50 steps.

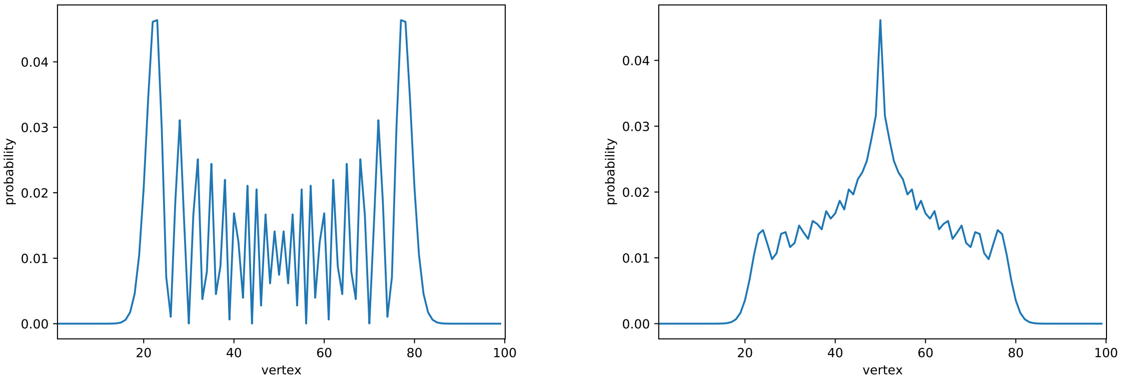
Figure 6: Discrete time quantum random walk that performs via the Hadamard coin and has a measurement at position -15 at different frequencies for 50 steps.

For the continuous-time quantum walk there are no set times which the particle sets a step and similarly there are no set times at which measurements can take place. For this type of walk it is usually chosen to perform stroboscopic measurements at a specific frequency. Just like the discrete case it is very important to choose the frequency of the measurements especially when one wants to measure a particle at a specific location for example. The next subsection will go into more details about local measurements for the CTQW and the effect of different frequencies and different positions on which we can measure. The collapsing of the wavefunction is also easier in the case of the CTQW, this is due to the particles only having a position state and not having a coin state which also needs to be normalized accordingly. For the rest of this report we will only look at measurements on continuous-time systems.

3.3.2 Global measurements

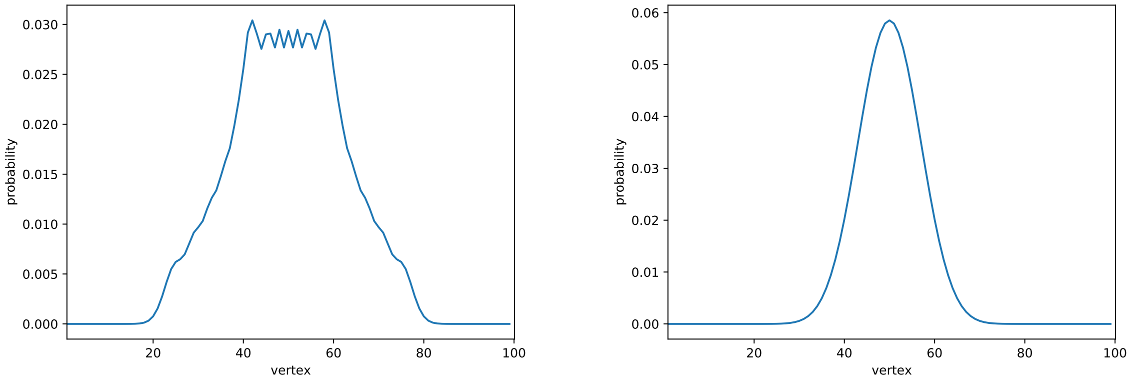
Global measurements are measurements that determine where the particle is located instead of checking if the particle is located in a specific position or not. This type of measurement gives the particle a very different behaviour compared to the local measurements described above. In the case of the global measurements the walk will evolve for a certain amount of time after which the walk starts over but at a different starting position. For the discrete quantum walk the behaviour of the quantum walk differs for different frequencies of global measurements. The walk approaches the classical random walk when the frequency of the measurements is brought to measure after each evolution. This can be seen by the fact that the first step of the QRW and the CRW are the same in the discrete case and since the walk resets after each step one has that the walk constantly has the same probability to move left and the same probability to move right just like the classical case. The other side of the spectrum is not measuring at all which is the quantum random walk described in section 3.1.2. For frequencies between these two the behaviour becomes more classical the lower the frequency is and stays more quantum the higher the frequency is.

For the continuous-time quantum walk we again have that the behaviour of the walk is very dependent on the frequency of the measurements, but now there is no set time on which measurements can be performed which means that the frequency can be lowered even further than the classical case. The limit of the time between measurement goes to 0 is called the Zeno limit. In this limit the CTQW does not evolve at all and will stay centered at the starting position for any number of measurements. The effects of the Zeno limit are further explained in the paper on global measurements in a CTQW from Barkai and Didi [11]. Figure 7 shows the CTQW for different measurement frequencies on $t = 15$. The next subsection will go into further detail on how the probability of the CTQW can be calculated after a number of measurements at a set frequency and the influence of this frequency on the system.



(a) Continuous-time quantum random walk on a line graph with 100 vertices starting in vertex 50 on $t = 15$. The plot has no measurements.

(b) Continuous-time quantum random walk on a line graph with 100 vertices starting in vertex 50 on $t = 15$. The plot has 1 measurement at $t = 7.5$.



(c) Continuous-time quantum random walk on a line graph with 100 vertices starting in vertex 50 on $t = 15$. The plot has 2 measurements at $t = 5$ and $t = 10$.

(d) Continuous-time quantum random walk on a line graph with 100 vertices starting in vertex 50 on $t = 15$. The plot has 10 measurements each $t = 1.5$ apart.

Figure 7: Continuous-time quantum random walks for 4 different number of measurements in the $t = 15$ time that the walk evolved. The graph that was used for these plots is shown in figure 23 in appendix A.3.

3.4 Detection problem

The previous section introduced the two most important types of measurements. Now that we know how these measurements work it is also important to look at what these measurements can be used for. One of the most notable problems in a quantum random walk is the detection problem. The detection problem can be seen as performing measurements at set time intervals to try and find a particle in a specific position. In this subsection we will focus on the probability that we eventually find the particle in the desired location.

3.4.1 Local measurements

This subsection follows the theory used in the research paper on dark states from Thiel et al. [12].

We will now look at what the probability is that a particle is found at a specific location in a network when we take measurements at a set time interval. It is important to note that in the classical case one would always end up with a probability of 1 for finding the particle in any position in the network eventually. Meaning that if we measure an infinite number of times eventually the particle will be found. However when we take local measurements in the quantum case there are certain positions and types of networks which will result in a probability lower than 1 even if an infinite number of measurements are taken.

The procedure for this detection problem is the following. We first start with an initial position $|\psi_0\rangle$ which we evolve for T time with the evolution operator e^{-iHT} where the Hamiltonian is chosen as the Laplacian matrix as in section 3.2.2, after this evolution we perform a measurement and if the measurement is successful that is the particle is found in the target position the procedure ends. If the measurement fails the wave function collapses as discussed earlier, this collapsing can be described by $|\psi_T\rangle = N(I-D)e^{-iHT}|\psi_0\rangle$ where I is the identity matrix, D is the projector matrix of the state we measure given by $D = |\psi_d\rangle\langle\psi_d|$ where $|\psi_d\rangle$ is the target state and N is a normalization constant. With this expression we can calculate the probability that we detect the particle on the n'th measurement. This can be written as $n-1$ unsuccessful measurements followed by a successful measurement resulting in the following equation

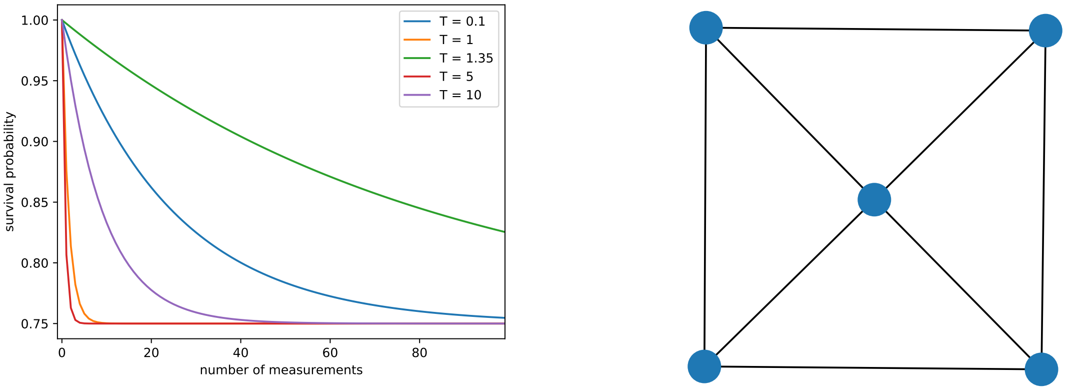
$$F_n = |DU((I-D)U)^{n-1}|\psi_0\rangle|^2 \tag{20}$$

Here F_n is the probability to find the particle for the first time on measurement n. This can also be translated to the probability that we have not found the particle for n times in a row.

This will be called the survival probability S_n and is given by the following equation

$$S_n = 1 - \sum_{i=1}^{n} F_i = |((I-D)U)^n |\psi_0\rangle|^2 \tag{21}$$

From now on we will call $\mathcal{S} = (I-D)U$ the survival operator which results in the following easy expression for the survival probability $S_n = |\mathcal{S}^n |\psi_0\rangle|^2$. Figure 8 shows this survival probability for different measurement intervals T on a square graph with a central vertex which is shown in the same figure.



(a) The survival probability from equation 21 for different measurement intervals for the first 100 measurements. The starting state is the central vertex of figure 8b and the target vertex is the top right vertex

(b) Structure of the network on which the CTQW operates. In this case the starting vertex is the central vertex and the target vertex is in the top right of the figure.

Figure 8: Survival probability for different measurement intervals on a square graph with a central vertex.

In this figure the central vertex was chosen as the initial position and the top right corner vertex was chosen as the detection position, but each of the corners gives the same result because of symmetry. From this figure it is clear that the probability of detection does not go to 1 since the survival probability does not go to 0. It is also clear that the measurement intervals have a substantial impact on the speed of which the probability goes to their limiting value.

We will first explain why the survival probability does not go to 0 as in the classical case, we do this by discussing something called dark and light states of the network. Dark states are states are initial states which have a survival probability of 1 and thus will never be found while light states have a survival probability of 0 and thus will always be found. These dark states do not evolve under the evolution operator and are thus eigenstates of both the Hamiltonian and the evolution operator and they have no overlap with the detection state which results in the following evolution from applying the survival operator

$$\mathcal{S}|\delta_l\rangle = (I-D)U|\delta_l\rangle = (I-D)e^{i\lambda_l T}|\delta_l\rangle = e^{i\lambda_l T}|\delta_l\rangle - 0 = e^{i\lambda_l T}|\delta_l\rangle \tag{22}$$

Here $|\delta_l\rangle$ is a dark state with eigenvalue λ_l of the Hamiltonian. Since the eigenvalue of the dark state with the evolution operator is on the unit circle we know that the survival probability of the dark state S_n is equal to 1 for each value of n. Now we will consider an eigenstate which has overlap with the detection state we can construct the light state $|\beta_l\rangle$ as follows

$$|\beta_l\rangle = \frac{P_l|\psi_d\rangle}{\sqrt{\langle\psi_d| P_l |\psi_d\rangle}} \tag{23}$$

Here P_l is the eigenstate projector of all eigenstates with eigenvalue λ_l with the Hamiltonian. So $P_l = \sum_{i=1}^{m} |E_{l,i}\rangle \langle E_{l,i}|$ where $|E_{l,i}\rangle$ is the i'th eigenstate with eigenvalue λ_l and this eigenvalue is m degenerate. All other $m-1$ eigenstates with this eigenvalue can be made orthogonal to this bright state which will also mean that they are orthogonal to the detection state by using a method like Gram-Schmidt. This will result in $m-1$ dark states.

With this information it is possible to determine the survival and detection probability after an infinite number of measurements at the the detection state for any initial state. The survival probability can be determined by calculating the overlap of the initial state with the dark states and the detection probability by calculating the detection probability with the light states. It is easier to calculate the overlap with the bright states which is what we will do here. The projector of all of the bright states is given by $P_\beta = \sum_l |\beta_l\rangle \langle \beta_l|$ where we sum over all the possible bright states which are given by equation 23. This gives us the following expression for the total detection probability after an infinite number of measurements

$$1 - S_\infty(\psi_0) = \langle \psi_0| P_\beta |\psi_0\rangle = \sum_l \frac{|\langle \psi_d| P_l |\psi_0\rangle|^2}{\langle \psi_d| P_l |\psi_d\rangle} \tag{24}$$

Now that we have found a way to determine the detection and survival probabilities it is also important to look at the measurement intervals which was shown in figure 8 to have a large impact on the convergence rate. However there is no dependence on T in the survival probability which means that each measurement interval eventually goes to the limiting value. However this is not the case, it is true that almost all measurement intervals will go their limiting value calculated by equation 24. But there are certain resonance frequencies which will cause the system to never detect the particle even if the initial state is not completely dark. These resonance frequencies are given by the values of T for which $(E_k - E_l)T = 0 \mod 2\pi$. It can also be seen in figure 9 that there are indeed frequencies for which the detection probability does not reach its limiting value. When n is taken to infinity these peaks become delta peaks centered around the resonance frequencies.

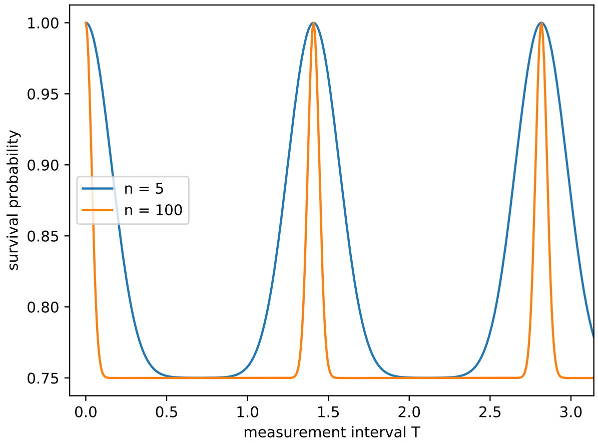


Figure 9: The survival probability after both 5 and 100 measurements plotted against the measurement interval time T. The starting state is the central vertex of figure 8b and the target vertex is the top right vertex. The resonance frequencies of this graph are multiples of $\frac{\pi}{\sqrt{5}}$.

The research paper dark states [12] also shows a simple limiting factor on the survival probability that makes use of the symmetry of a system. If a system has ν equivalent vertices viewed from the initial condition then the survival probability is limited by

$$S_\infty \geq 1 - \frac{1}{\nu} \tag{25}$$

This is also visible in figure 8 where there are 4 identical vertices when we look from the starting vertex which causes the survival probability to be larger than $\frac{3}{4}$ which is also visible in the graph depicting the survival probability.

3.4.2 Global measurements

This subsection follows the theory on the detection probability from the paper on global measurements from Barkai and Didi [11].

Similarly to the case of the local measurements we will periodically measure if the particle is in the desired target state after evolving for a set amount of time, however instead of checking if the particle is located in the target state or not we will now check where the particle is instead. This will result in the wavefunction of the particle fully collapsing after each measurement and in contrast to the case of the local measurements the probability to be detected after an infinite number of measurements will be 1, independent of the target state and initial position. This can be seen by the fact that when we increase the number of measurements in a certain time interval the walk will move more and more to the classical regime which can be seen in figure 7. So for an infinite number of measurements one would also expect behaviour which is similar to classical which is the case for the detection probability always going to 1. Since the wavefunction of the particle fully collapses after each measurement we can see the walk after the first measurement as a normal CTQW with a different starting position and the probability of the starting position is proportional to the probability that the particle was found there after the measurement. Repeating this process we can construct a new evolution operator which gives us the probabilities after each measurement, this operator will be defined as G and is given by the following equation

$$G = \sum_{x,x' \in X} |\langle x'| e^{-iTH} |x\rangle|^2 |x'\rangle \langle x| \tag{26}$$

Here X is the set of all vertices of the network and T is the time between measurements. This operator gives the probabilities after n measurements with this equation

$$|P_n\rangle = G^n |\psi_0\rangle \tag{27}$$

Now that we know how the collapsing walk evolves under the measurements, it is possible to determine the probabilities to detect the particle in the target state for the first time after n measurements. This will go in a similar way as the local measurements. We first have $n-1$ unsuccessful measurements followed by a successful measurement which results in the following probability

$$F_n = \langle \psi_d| (G(I-D))^{n-1} G |\psi_0\rangle \tag{28}$$

Then the probability to be detected eventually is given by the infinite sum over n for these probabilities. Figure 10 shows the cumulative detection probability for different measurement frequencies.

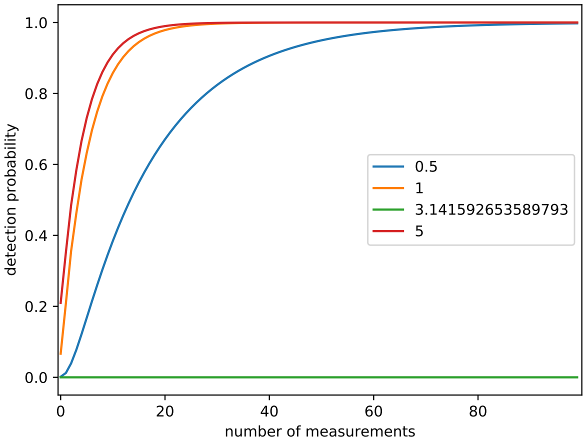


Figure 10: The cumulative detection probability for the first 100 measurements for different measurement intervals. The graph used for this plot is a cycle graph with 6 vertices and the target and starting positions are positioned on the opposite site of the cycle, so vertex 1 and 4.

This probability can also be calculated by the means of Z-transforms of the probabilities in 28, this Z-transform also provides a way to calculate the average number of measurements and the variance in the number of measurements. This report will not focus on these quantities, but they can be found in the paper about the renewal equation for a CTQW by Friedman et al. [13].

As seen in figure 10 the frequency has an important role in the speed of the convergence to the limiting value of 1 for the detection probability, but similarly to the local measurement case, there are some measurement frequencies at which the detection probability remains 0 when the detection state and the initial state differ. This can be seen by figure 11, which shows the probability to detect the particle after a set number of measurements for different frequencies. As in the case of the local measurements these frequencies are the resonance frequencies, which occur, when $(E_k - E_l)T = 0 \mod 2\pi$. Since for these values the evolution operator reduces to the identity.

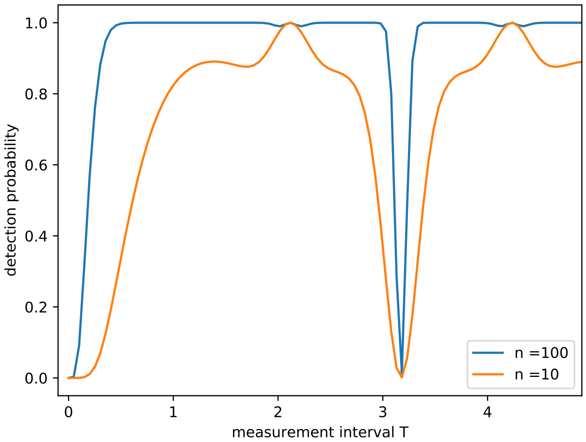


Figure 11: The cumulative detection probability after both 10 and 100 measurements plotted against the measurement time T. The graph used for this plot is a cycle graph with 6 vertices and the target and starting positions are positioned on the opposite site of the cycle, so vertex 1 and 4. The resonance frequencies are multiples of π

4 Measurement uncertainties

Implementations of a CTQW are never perfect, one will always have uncertainty in some aspect of the process. For this report we look at the influence of uncertainty in the time that we perform a measurement. This choice was made because the three main problems which will be discussed in detail in sections 5, 6 and 7 all use measurements at certain times to solve their problems.

When there is no uncertainty in the measurement time we let the CTQW evolve for T time after which a measurement is performed which is then followed by letting the particle evolve for T time once again. When the measurement uncertainty is introduced we do not let this CTQW evolve for T time but instead we let it evolve for $T + \delta t$ where δt is taken from a normal distribution centered around 0, the standard deviation will be changed for different levels of uncertainty. Some conditions need to apply to δt since the uncertainty in the evolution time can never make the total time of evolution, $T + \delta t < 0$ negative, we impose the following condition on δt

$$\delta t = \begin{cases} -T & T + N(0, \sigma^2) < 0 \\ N(0, \sigma^2) & T + N(0, \sigma^2) \geq 0 \end{cases} \tag{29}$$

Here $N(0, \sigma^2)$ is a sample from the normal distribution with mean 0 and variance σ^2. This new evolution time, which changes for each measurement, can be used in every equation that was explained in the theory of this report. Figure 20 in appendix A.2 shows the different probabilities after 3 global measurements, which are performed as explained in section 3.4.2, along with the probabilities when an uncertainty is introduced in the measurement frequency. We can no longer use equation 27 to determine the probability distribution after global measurements with an uncertainty due to the time between measurements no longer being constant. This results in a new G matrix after each measurement. To determine the probability after n measurements we calculate n G matrices with equation 26 with evolution times that have uncertainties. After this calculation we get the following equation for the probability distribution

$$|P_n\rangle = G_1 \cdot G_2 \cdot \ldots \cdot G_n \, |\psi_0\rangle \tag{30}$$

In appendix A.2, we also shows the effect of measurement uncertainty on the detection probability for the case of global measurements and local measurements. These detection probabilities are constructed by taking either n G matrices as above for the global measurements or n different survival operators $\mathcal{S}$ which are described in section 3.4.1.

To determine how severe the effects of the measurement uncertainties are, we calculate the absolute difference between the simulation with no uncertainty and a simulation with uncertainty. In the case of network centrality we do this by taking the difference in the value for each vertex and summing over all vertices in a similar way to equation 35, which will be further explained in section 5. We also determine the error in simulation with uncertainty, which we define as

$$Error = \frac{|Theory - Simulation|}{|Theory|} \cdot 100\% \tag{31}$$

Here $Theory$ represents the result of a calculation with no uncertainty and $Simulation$ represents the result of a simulation with an uncertainty. So $Theory$ and $Simulation$ can either be a probability distribution, ranking of vertices, a difference in graph certificates or a detection probability depending on the problem we study. The exact methods that are used to simulate the uncertainties within the network centrality problem, graph isomorphism problem and spatial search problem will be explained in sections 5.2, 6.2 and 7.2.

5 Network centrality

5.1 Theory

This subsection follows the theory of Izaac et al.[14], Wang et al.[15] and Qiang et al.[7] for the different types of network centrality.

Network centrality has multiple important applications such as finding the most important positions in infrastructure, studying the most influencial people in social networks, pageranking in search engines and in the spreading of deceases in for example an epidemic.

In its most basic form network centrality gives each vertex in a graph a number, the higher this number is the more important or central this vertex is. There are multiple ways one can define a vertex to be important in a graph, which results in there being multiple ways to indicate the centrality of a vertex in a graph. The simplest centrality measure is the degree centrality which ranks the vertices by their degrees. Another centrality measure and the one this section will focus on is eigenvalue centrality, this centrality measure ranks the vertices by the values of their entries in the eigenvector corresponding to the largest eigenvalue of the adjacency matrix of the graph. So $C_i = \mathbf{v}_i$ where C_i denotes the centrality of vertex i and $\mathbf{v}$ is the eigenvector which corresponds to the largest eigenvalue. Bonacich has shown in [16] that these entries of the eigenvector are proportional to the limit of to the sum of the rows of the following matrix S when β approaches $\frac{1}{\lambda}$, where λ is the largest eigenvalue of the adjacency matrix A.

$$S = \sum_{n=1}^{\infty} \beta^{n-1} A^n \tag{32}$$

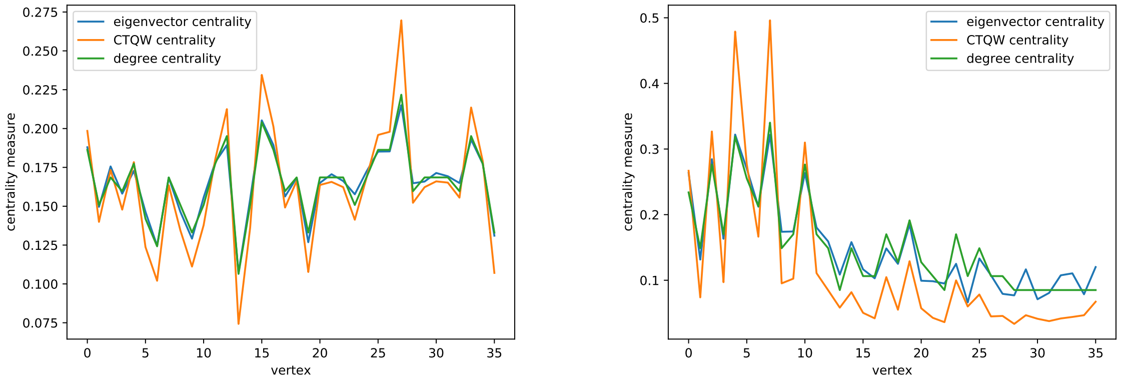
This matrix S can be seen as a matrix which is the sum of walks on the graph of each number of steps. The eigenvector is proportional to this matrix by the following equation $\mathbf{v}_i \propto \sum_j S_{ji}$. This results in vertices that neighbor other vertices with a high degree have a higher centrality measure, due to the random walk reaching these vertices more often.

In the previous sections it was shown that the quantum random walk propagates through graphs much faster than the classical random walk and thus the quantum random walk seems like a viable method to determine the centrality in a network. The method that was chosen for the quantum walk centrality has a similar premise as the eigenvector centrality in the sense that vertices where the particle is most likely to be after each step of a CTQW have a higher centrality measure. As described in section 3.2.2 the probability of a particle being in a specific vertex after time t with initial position $|\psi_0\rangle$ is given by equation 18, it is clear that the initial condition has a large impact in the outcome of this probability. Therefore the initial condition is chosen as an equal superposition over all the vertices in the network. So $|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} |i\rangle$ where N is the number of vertices in the network.

Choosing this initial position does present a problem with the evolution operator which was introduced in section 3.2.2 since this initial position is an eigenvector of the Laplacian matrix which is chosen as the Hamiltonian of the system resulting in the initial position also being an eigenvector of the evolution operator and therefore not undergoing evolution at all. Therefore the Hamiltonian is changed to the adjacency matrix instead of the Laplacian matrix for the centrality measure. It is also clear that one can not take the centrality to be determined by a single probability measure, since the CTQW has no equilibrium position. Therefore the centrality measure is chosen to be taken as a time average over infinite time of the CTQW, this results in the following equation for the centrality measure based on the CTQW

$$C_i^Q = \lim_{T\to\infty} \frac{1}{T} \int_0^T |\langle i| e^{-iAt} |\psi_0\rangle|^2 dt \tag{33}$$

This centrality measure can be performed by the centrality_test() command which can be found on Github which is linked in appendix A.1. We will now look if the centrality measure based on the CTQW is in accordance with the classical centrality measures. Figure 12 shows the value of the centrality measure for two classical centrality measures, namely the degree centrality and eigenvector centrality and the CTQW based centrality measure. These centrality measures where performed on a Barabasi-Albert graph and a Erdös-Rényi graph of 36 vertices.



(a) The normalized centrality measures for degree centrality, eigenvalue centrality and the CTQW based centrality measure from equation 33 for a Erdös-Rényi$(36, 0.5)$ graph.

(b) The normalized centrality measures for degree centrality, eigenvalue centrality and the CTQW based centrality measure from equation 33 for a Barabasi-Albert$(36, 4)$ graph.

Figure 12: Centrality measurements for a Barabasi-Albert and an Erdös-Rényi graph. Examples of the Barabasi-Albert and Erdös-Rényi graphs are shown in figure 25 and 28 in appendix A.3.

From figure 12 it is clear that the centrality measure based on the CTQW is indeed similar to the classical centrality measures and is therefore a viable method to determine the centrality in a network. This can be seen by the fact that the ordering of the vertices from most central to least central in the quantum case matches ordering in the classical case. The ordering can differ slightly for vertices that have similar centrality measures, but this ordering also differs between different classical centrality measures.

5.2 Method

The CTQW based centrality measure works by taking an integral over the probability of a particle being in a specific vertex over an infinte time. To simulate this measure numerically we will take a summation over a large number of measurements which are very close to each other and add these to approximate the behaviour of the integral. In this report it is chosen to perform 10000 measurements over $T = 500$. All of these measurements will be $500/10000 = 0.05$ time apart for the case when there is no measurement uncertainty. When the measurement uncertainty is introduced the time between measurements will be taken as $0.05 + N(0, \sigma^2)$, with the limit from equation 29. The number of measurements will stay at 10000 but the total time might differ slightly from $T = 500$ due to the uncertainty in the measurement times. To determine the effect of the measurement uncertainty we calculate the absolute difference between the theoretical case and the experimental case we also calculate the error via equation 31. This error will be calculated for a set of randomly generated graphs with the same properties, Erdös-Rényi and Barabasi-Albert graphs, For each graph the error will be calculated for a large number of simulations where the standard deviation is kept constant. This process will then repeated for a number of different standard deviations.

5.3 Results

For this report we chose to focus only on Erdös-Rényi and Barabasi-Albert graphs when we study the effects of measurement uncertainty in the network centrality problem. To study this effect we chose to look at 4 different values of σ^2 in equation 29 to obtain different distributions for δt. The values of σ^2 are the following $0.05, 0.01, 0.001$ and 0.0001. The time between each measurement is 0.05 as stated in section 5.2, which means that the larger values of σ^2 correspond to higher levels of uncertainty. The following tables show the absolute difference between the outcome of the centrality measure with and without uncertainties. The average absolute difference is calculated from 100 simulations, the choice of the number of simulations will be further discussed in section 9. The following tables also show the average error between the outcomes of the centrality measure with and without uncertainties.

The Erdös-Rényi graphs have $p = 0.5$ for both tables and the Barabasi-Albert graphs have an average degree 4 for each newly added vertex for both tables. A Barabasi-Albert graph is a scale-free network, this means that the degree distribution follows a power law in the form of $P(k)\ k^{-\gamma}$, here k denotes the degree of a vertex and γ is a constant. For the Barabasi-Albert graph model we have $\gamma = 3$, this means that the degree distribution of the Barabasi-Albert graphs is of the following form $P(k)\ k^{-3}$ [17]. Barabasi-Albert graphs are constructed by growing a network one vertex at a time, each new vertex will form d edges with existing vertices. When there are fewer vertices than the value of d the new vertex will form edges with each existing vertex. The new vertices are more likely to connect to existing vertices with a high degree. The probability of forming an edge with an old vertex scales linearly with the degree of that vertex. Table 2 shows the outcomes for graphs with 50 vertices and table 3 shows the outcomes for graphs with 100 vertices.

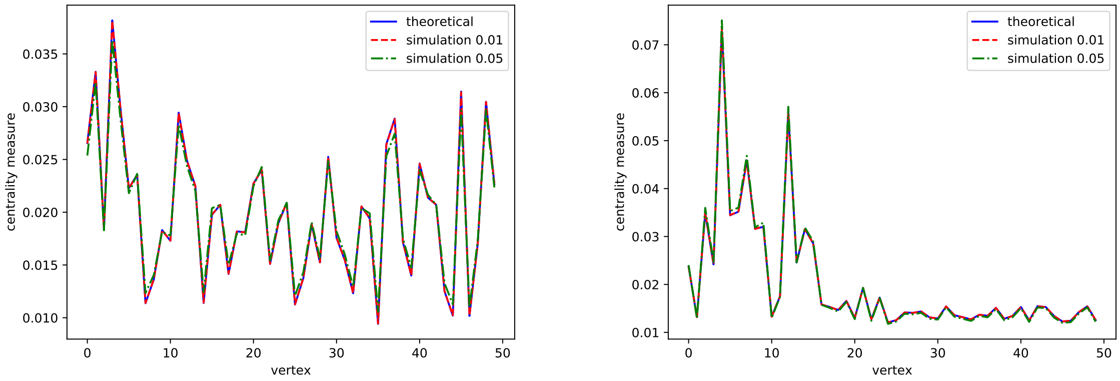
Table 2: Absolute difference between the centrality measure without uncertainties and the centrality measure with different levels of uncertainty. The Erdös-Rényi graphs used for these calculations are all$(N, p) = (50, 0.5)$ and the Barabasi-Albert graphs are all $(N, d) = (50, 4)$. Each value is calculated from 100 simulations. The columns with % show the average error between the theoretical outcome and the outcome with uncertainty.

type	Erdös-Rényi	Erdös-Rényi %	Barabasi-Albert	Barabasi-Albert %
$\sigma^2 = 0.05$	0.027 ± 0.0049	3.2	0.022 ± 0.0071	2.1
$\sigma^2 = 0.01$	0.0016 ± 0.00089	0.17	0.0015 ± 0.00062	0.15
$\sigma^2 = 0.001$	0.00014 ± 0.000070	0.015	0.00018 ± 0.000095	0.017
$\sigma^2 = 0.0001$	0.0000094 ± 0.0000038	0.0010	0.000017 ± 0.000011	0.0016

Table 3: Absolute difference between the centrality measure without uncertainties and the centrality measure with different levels of uncertainty. The Erdös-Rényi graphs used for these calculations are all$(N, p) = (100, 0.5)$ and the Barabasi-Albert graphs are all $(N, d) = (100, 4)$. Each value is calculated from 100 simulations. The columns with % show the average error between the theoretical outcome and the outcome with uncertainty.

type	Erdös-Rényi	Erdös-Rényi %	Barabasi-Albert	Barabasi-Albert %
$\sigma^2 = 0.05$	0.047 ± 0.0043	5.1	0.021 ± 0.0065	2.1
$\sigma^2 = 0.01$	0.0053 ± 0.0010	0.58	0.0027 ± 0.0011	0.26
$\sigma^2 = 0.001$	0.00014 ± 0.000056	0.015	0.00023 ± 0.00013	0.021
$\sigma^2 = 0.0001$	0.000014 ± 0.0000059	0.0015	0.000021 ± 0.000012	0.0022

From these tables it is visible that both the absolute difference and the error between the theoretical and simulated outcomes seem to scale linearly with σ^2 for small values of σ^2 for both the graphs with 50 and 100 vertices. This effect also seems to be evident for the larger values for σ^2 for the Barabasi-Albert graphs, but is less evident for large values for σ^2 in the Erdös-Rényi graphs. These results will be further discussed in section 9. Figure 13 shows a simulation without uncertainty and two simulations with different degrees of uncertainty for both a Barabasi-Albert graph and a Erdös-Rényi graph. From this figure we can see that the values that are assigned to the vertices barely change when an uncertainty is introduced. This results in the overall ordering of most central to least central nodes staying the same. Only vertices that are almost similar in the theoretical case might swap positions in the ordering.



(a) The theoretical centrality measures on an Erdös-Rényi(50,0.5) graph with the centrality measures with both $\sigma^2 = 0.05$ and $\sigma^2 = 0.01$.

(b) The theoretical centrality measures on an Barabasi-Albert(50,4) graph with the centrality measures with both $\sigma^2 = 0.05$ and $\sigma^2 = 0.01$.

Figure 13: The theoretical centrality measures for different types of graphs and the simulations with a certain uncertainty.

6 Graph isomorphism problem

6.1 theory

The study of graph isomorphisms is very important in a number of fields such as image processing, cryptography, security and analyzing social structures. The graph isomorphism (GI) problem is to check if two graphs of finite size are isomorph, that is to say that the graphs are identical up to a reordering of the labels of the vertices. The GI problem is not known to be solvable in polynomial time nor to be NP-complete, the best current algorithm works in quasi-polynomial time [18]. Quantum random walks are thought to be useful in determining if two graphs are isomorph. There are multiple quantum models that are used for the GI problem. In this report a quantum model was chosen that works in continuous-time and makes use of return probabilities of the CRQW.

The method to determine if two graphs are isomorph will next be explained; the Python code used for GI detection is presented in the function "isomorph_test()" that can be found on Github, which is linked in appendix A.1. To check if the graphs in question are isomorph, a graph certificate is calculated for each graph. A graph certificate is a vector which we assign to a graph that tells us something about its structure. One could, for example, use the centrality measure which we discussed in section 5 as a graph certificate. After the calculation of the certificates we compare them to each other and if they are identical it means that the graphs are most likely isomorph. To determine if they are actually isomorph a mapping of the vertices of one of the graphs needs to be found to match the labels of the other graph. However, if the certificates differ the two graphs are not isomorph.

To solve the GI problem we first need to know how one calculates the graph certificate. There are multiple different certificates, the one chosen for this report is constructed as follows. First we create an array $A(t)$ of length equal to the number of vertices of the graph, after that each entry of the array is calculated by the following equation [15]

$$A_i(t) = |\langle i| e^{-itH} |i\rangle|^2 \tag{34}$$

Here i is the entry of the array which also corresponds to the label of a vertex of the graph, H is the Hamiltonian of the system which corresponds with the Laplacian matrix of the graph and t is the time on which the certificate is calculated. This means that the certificate array corresponds with an array that gives the return probability at time t of each of the vertices. To check if the certificates are identical they first need to be sorted. After this sorting the difference is taken in the following way

$$\|A - B\| = \frac{1}{2}\sum_i |A_i - B_i| \tag{35}$$

If the outcome of this difference is nonzero, one can conclude that the two graphs are not isomorph. If the outcome is 0, one can check for multiple times and if each check results in the difference being equal to 0 the graphs are most likely isomorph.

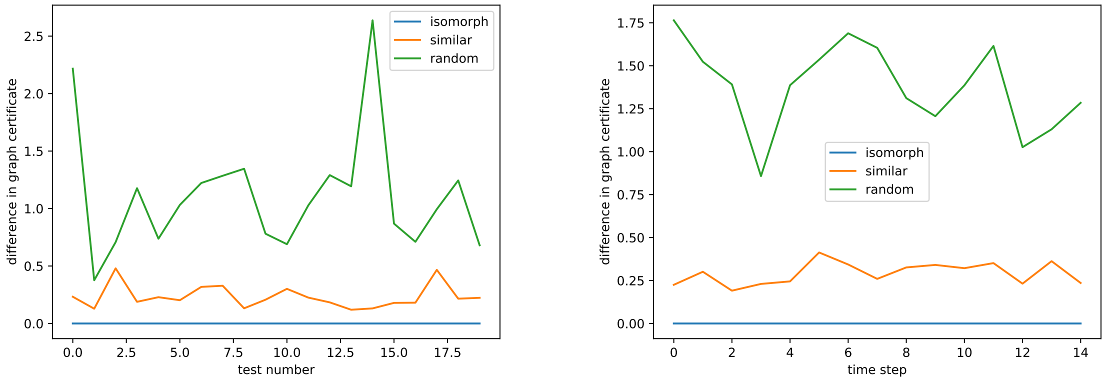
To determine if the two graphs are actually isomorph, a mapping needs to be found between the labels of the vertices of the two graphs. After this mapping has been found, the rows and columns of the adjacency matrix of the graph can be swapped according to this mapping and if the adjacency matrices are the same it can be concluded that the graphs are indeed isomorph.

The mapping of the labels of the graph is also constructed by using using CTQW, however the network is modified slightly. For each of the vertices an extra certificate is calculated where the graph has a weighted self-loop on the vertex in question. This is done for both graphs. After calculating these certificates we calculate the difference between the certificate of a specific vertex of the first graph with all certificates of the second graph. This proces is repeated for each vertex of the first graph. The two certificates which have difference 0 are then mapped to each other. Since this process is done for each vertex it is clear that this algorithm uses $\mathcal{O}(N^2)$ calculations at most since each vertex of the first graph is compared with each vertex of the second graph. This algorithm is shown by the following pseudo code 1[7].

Algorithm 1 Finding the isomorphic mapping from G to H

```
Input: Graph $G$ and graph $H$
$N$ = number_of_vertices($G$)                    ▷ Number of vertices for both graphs
$f := G \to H$                                   ▷ Defining the mapping
for i = 1 : N do
    Create a self-loop for vertex $i$ on graph $G$ to obtain graph $G_i$
    Calculate the graph certificate $C_{G_i}$ of $G_i$
    Create a self-loop for vertex $i$ on graph $H$ to obtain graph $H_i$
    Calculate the graph certificate $C_{H_i}$ of $H_i$
end for
for i = 1 : N do
    for j = 1 : N do
        if $C_{G_i} = C_{H_j}$ then
            Save vertex $i$ of graph $G$ as similar to vertex $j$ of graph $H$
        end if
    end for
end for
Find $f$ from the saved similar vertices
Return: $f$
```

Figure 14 shows the difference in the graph certificates for 20 isomorph graphs, 20 non-isomorph graphs and 20 graphs that are very similar. The isomorph graphs are created by making a copy of the orignal graph and performing a random mapping of the labels of the vertices which guarantees that the graphs are isomorph. The very similar graphs are constructed by adding or changing a small number of edges from the original graph, between 1 and 5 changes for this figure. This figure also shows the difference in graph certificate for 3 graphs which are generated the same as above for different measurement times.



(a) The difference in graph certificate calculated by equation 35 for randomly generated graphs with 100 vertices and on average 400 edges. The generated graphs are either isomorph, similar or completely random from each other. For each type there are 20 generated graphs.

(b) The difference in graph certificate calculated by equation 35 for randomly generated graphs with 100 vertices and on average 400 edges. The generated graphs are either isomorph, similar or completely random from each other. For each type we keep the generated graphs and look at different times.

Figure 14: Difference in the graph certificates for isomorphic graphs, similar graphs and random graphs of 100 vertices and on average 400 edges. An example of such a random graph is shown in figure 27 in appendix A.3.

6.2 Method

In the theoretical case two isomorphic graphs would always have the same graph certificate. However, when we introduce measurement uncertainty these graph certificates are no longer identical. In order to conclude if two graphs are isomorph or not, we will have to introduce an upper bound on the difference in graph certificate for which we say that the graphs are most likely isomorph. In the theoretical case we calculate the return probability of a particle after some time T, which is the same for each vertex, however for simulating the measurement uncertainty we will calculate the return probability for each vertex after some time $T + N(0, \sigma^2)$, which will slightly differ for each vertex. To determine the effect of the measurement uncertainty on the graph isomorphism problem we will calculate the absolute difference and error between graph certificate of the theoretical case and the simulated case. For isomorphic graphs we can not calculate the error because the theoretical result will always be 0 which would result in division by 0 in equation 31. Because of this fact we chose to only calculate the absolute difference in the case of two isomorphic graphs. This absolute difference and error will be calculated a large number of times for a set of specific isomorphic graphs with a constant standard deviation. This process will then be repeated for multiple standard deviations. This process will also be performed for a set of similar graphs and a set of random graphs with the same properties meaning that they are generated in the same way.

6.3 Results

To determine the influence of measurement uncertainty on the graph isomorphism problem we look at three distinct cases, isomorphic graphs, randomly generated graphs with the same properties and graphs which are very similar. We will first discuss the results of the isomorphic graphs. For these graphs we calculate both the difference in graph certificate for a simulation without uncertainty and once with uncertainty, after that we calculate the absolute difference between these two values. Due to the theoretical result always being 0 for graphs that are isomorph, we can not take the error between the theoretical result and the simulated results with an uncertainty as explained in section 6.2. The graph certificates are taken on 4 different times namely $1, 2, 4$ and 8. For each of these times we look at three values of σ^2. Each of the calculated absolute differences is the average of 750 simulations. Table 4 shows these absolute differences for random graphs with 50 vertices and 200 edges which are isomorph.

Table 4: Average absolute difference between the simulation with no uncertainty and the simulations which have a specific uncertainty for isomorphic graphs. The graphs used for this simulation are randomly generated graphs with 50 vertices and 200 edges. Each average is taken from 750 simulations.

time	1	2	4	8
$\sigma^2 = 0.1$	0.27 ± 0.050	0.29 ± 0.062	0.16 ± 0.031	0.17 ± 0.030
$\sigma^2 = 0.01$	0.050 ± 0.0074	0.061 ± 0.0091	0.030 ± 0.0048	0.032 ± 0.0045
$\sigma^2 = 0.001$	0.0055 ± 0.00078	0.0068 ± 0.00099	0.0033 ± 0.00050	0.0035 ± 0.00050

Similarly to the results for the graphs which are isomorph, we calculated the absolute difference between two random graphs which are generated in the same way. That is to say that both are random graphs with 50 vertices and 200 edges. Table 5 shows the average absolute difference. The theoretical results for two random graphs does not yield 0 which allows us to calculate the error between the theoretical results and the simulated results, which are given in table 6.

Table 5: Average absolute difference between the simulation with no uncertainty and the simulations which have a specific uncertainty for random graphs which are generated in the same way. The graphs used for this simulation are randomly generated graphs with 50 vertices and 200 edges. Each average is taken from 750 simulations.

time	1	2	4	8
$\sigma^2 = 0.1$	0.094 ± 0.071	0.077 ± 0.059	0.058 ± 0.045	0.072 ± 0.056
$\sigma^2 = 0.01$	0.011 ± 0.0082	0.0099 ± 0.0071	0.0063 ± 0.0046	0.0070 ± 0.0054
$\sigma^2 = 0.001$	0.0011 ± 0.00088	0.00088 ± 0.00072	0.0062 ± 0.00050	0.00073 ± 0.00055

Table 6: Average error in % between the simulation with no uncertainty and the simulations which have a specific uncertainty for random graphs which are generated in the same way. The graphs used for this simulation are randomly generated graphs with 50 vertices and 200 edges. Each average is taken from 750 simulations.

time	1	2	4	8
$\sigma^2 = 0.1$	7.0	5.0	5.0	7.8
$\sigma^2 = 0.01$	0.83	0.60	0.51	0.72
$\sigma^2 = 0.001$	0.082	0.059	0.050	0.074

We once again calculate these absolute differences, but this time for similar graphs. The results of this are shown in table 7. The average errors are given in table 8. The similar graphs are created by first taking an isomorphic graph after which we add between 1 and 4 new edges to the graph.

Table 7: Average absolute difference between the simulation with no uncertainty and the simulations which have a specific uncertainty for similar graphs which have between 1 and 4 extra edges. The graphs used for this simulation are randomly generated graphs with 50 vertices and 200 edges. Each average is taken from 750 simulations.

time	1	2	4	8
$\sigma^2 = 0.1$	0.12 ± 0.046	0.12 ± 0.063	0.057 ± 0.045	0.049 ± 0.035
$\sigma^2 = 0.01$	0.0093 ± 0.0074	0.013 ± 0.010	0.0073 ± 0.0052	0.0068 ± 0.0048
$\sigma^2 = 0.001$	0.0010 ± 0.00072	0.0013 ± 0.00097	0.00057 ± 0.00039	0.00078 ± 0.00051

Table 8: Average error in % between the simulation with no uncertainty and the simulations which have a specific uncertainty for similar graphs which have between 1 and 4 extra edges. The graphs used for this simulation are randomly generated graphs with 50 vertices and 200 edges. Each average is taken from 750 simulations.

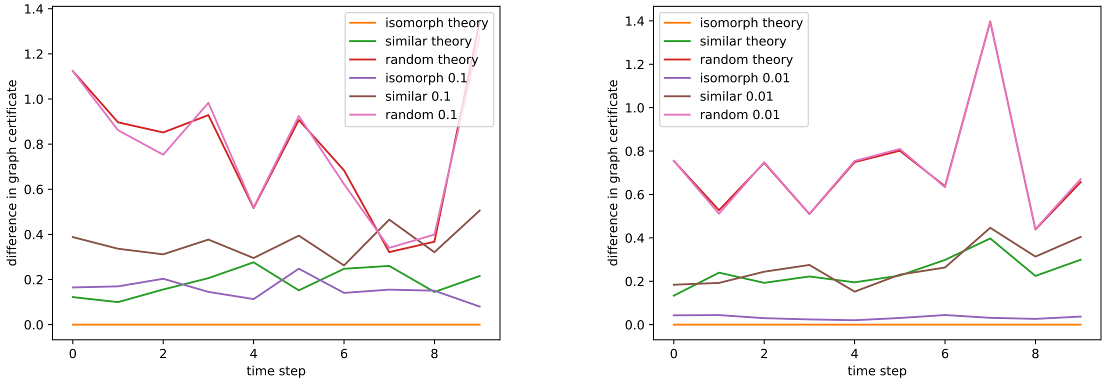
time	1	2	4	8
$\sigma^2 = 0.1$	64	52	19	11
$\sigma^2 = 0.01$	3.6	5.2	1.5	1.2
$\sigma^2 = 0.001$	0.35	0.49	0.14	0.12

All of the mentioned calculations were also done for graphs with 100 vertices and 400 edges for a time of $t = 1$. Table 9 shows these results. For the similar graph we once again add between 1 and 4 edges.

Table 9: Average absolute differene and average error between the simulations with and without an uncertainty for each of the three types of graphs. Each graph has 100 vertices and 400 edges, except for the similar graphs where we have between 401 and 404 edges. Each average is taken from 750 simulations.

type	isomorph	random	similar	random %	similar %
$\sigma^2 = 0.1$	0.31 ± 0.061	0.16 ± 0.069	0.10 ± 0.076	6.5	91
$\sigma^2 = 0.01$	$0.0.064 \pm 0.0074$	0.016 ± 0.0088	0.010 ± 0.0091	0.65	18
$\sigma^2 = 0.001$	0.0062 ± 0.00078	0.0011 ± 0.00081	0.0011 ± 0.00082	0.069	0.8

From these results we once again see the same linear pattern for the smaller values of σ^2. We do however see that the error is much larger for similar graphs than for randomly generated graphs and we see that this effect is even more evident for the larger graph. The reasoning for this will be discussed further in section 9. Figure 15 shows the result of a theoretical simulation and two simulations with an uncertainty to illustrate the effect of this uncertainty. From this figure we can see that it is still possible to determine if two graphs are isomorphic when the uncertainty is small by setting an upper bound on the graph certificate. For large uncertainties as in figure 15a we see that we are no longer able to determine if two graphs are isomorphic due to the difference in graph certificate for isomorphic graphs and similar graphs being of the same magnitude.



(a) The theoretical difference in graph certificate and the difference in graph certificates for a simulation with $\sigma^2 = 0.1$ for 10 different times.

(b) The theoretical difference in graph certificate and the difference in graph certificates for a simulation with $\sigma^2 = 0.01$ for 10 different times.

Figure 15: The theoretical difference in graph certificates for different types of graphs and the simulations with a certain uncertainty for 10 different times. An example of a random graph that was used for this figure is shown in figure 27 in appendix A.3.

7 Spatial search problem

7.1 Theory

This subsection follows the theory from the paper on the spatial search problem on Erdös-Rényi graphs from Chakraborty et al.[19] and Wang et al. [15].

A search problem is computational problem where one tries to find a specific element or structure in an object, such as finding a name in a database. An algorithm solves a search problem if it is able to find the specific structure or it can conclude that the structure is not present in the object. Search problems have a large number of applications such as route-planning, searching for strategies in game theory and retrieving information from a database The search problem can be applied to a number of different systems, but in this section the focus will lie with searches on graphs. For the search problem on a graph one has a specific target vertex which one wants to find and one starts on a specific position. The goal of this problem is to end up in the target vertex and the algorithm will stop once the particle is in the target vertex.

Classical search algorithms can have a time complexity of order $\mathcal{O}(N)$. With the use of quantum random walks the search problem can be sped up quadratically which results in a time complexity of order $\mathcal{O}(\sqrt{N})$, with probability of finding the vertex reaching order $\mathcal{O}(1)$.

The method for the spatial search problem using the CTQW is very similar to a normal CTQW. The goal of this method is to determine the time at which a measurement would result in the optimal chance of the particle being in the desired target state, this can either be a single vertex or one can choose to have multiple target states. The target states are denoted as w_i where the subscript can be omitted when we only have one target state and the starting state is a superposition of the starting positions denoted by r_i. The number of target states is denoted as n_w and the number of starting states is denoted as n_r, it is also chosen to have no common states between the target states and the starting states so $T \bigcap S = \emptyset$ where T is the set of target states and S the set of starting states.

The Hamiltonian of the search problem is similar to the Hamiltonian of the normal CTQW, yet it has some important differences. The Hamiltonian of the search problem is given by the following equation

$$H_{search} = -\sum_{i=1}^{n_r} |r_i\rangle \langle r_i| - \sum_{i=1}^{n_w} |w_i\rangle \langle w_i| - \kappa A \tag{36}$$

Here A is the adjacency matrix of the graph in question and $\kappa = \frac{1}{\lambda_1^A}$, where λ_1^A is the largest eigenvalue of the the adjacency matrix. This search Hamiltonian only differs on the main diagonal compared to Hamiltonian in equation 13. Elements on the main diagonal decrease the probability of moving away from the vertex associated with that element. This results in the search Hamiltonian being a good Hamiltonian for this problem due to the fact that the particles tend to stay in the detection states which is beneficial if we want to find them. As stated earlier the starting position for this search process is a superposition of the starting states, denoted by the following state

$$|\psi_0\rangle = \frac{1}{\sqrt{n_r}} \sum_{i=1}^{n_r} |r_i\rangle \tag{37}$$

The goal of this algorithm is to determine at which time t the probability of being in the target states is at it's maximum, this probability is given by this equation

$$P_w(t) = \sum_{i=1}^{n_w} |\langle w_i| e^{-iH_{search}t} |\psi_0\rangle|^2 \tag{38}$$

This probability is shown for a number of different graphs in figure 16, these probabilities were generated by the spatial_search() command which is shown in appendix A.1.

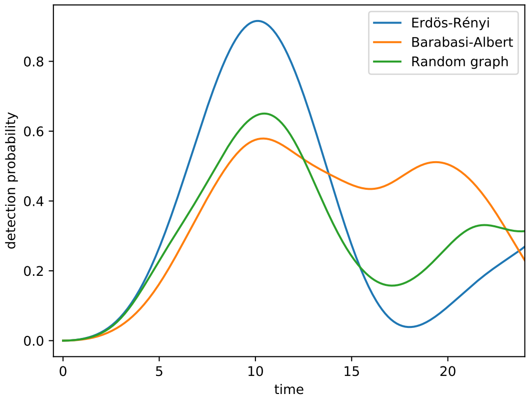


Figure 16: Detection probability in 3 random target states with 3 initial states. Each of the graphs has 64 vertices and the Erdös-Rényi graph has $p = 0.5$, the Barabasi-albert graph has 4 edges on each new vertex and the random graph has 256 edges. Examples of an Erdös-Rényi, Barabasi-Albert and a random graph are shown in figure 28, 25 and 27 in appendix A.3.

It is also very important that the value of κ is chosen optimally as it is when it is chosen as $\frac{1}{\lambda_1^A}$, the proof of why this particular value of κ is optimal is given in the supplementary material of Chakraborty et al.[19]. Other values of κ can result in the probability of the particle being in the target state being far lower and not having a clear optimal time to search. This can be seen in figure 17 where the search probability is shown for different values of κ. All these probabilities are on the same Erdös-Rényi graph with 64 vertices and a probability of 0.5 of forming the edges. For an Erdös-Rényi graph the largest eigenvalue is always around $N \cdot p$, where N are the number of vertices and p is the probability for an edge to form which would result in κ to be around $\frac{1}{64 \cdot 0.5} = 0.03125$ for the optimal case.

It can also be shown that the probability of finding the particle in the target state of an Erdös-Rényi graph can be given by the following equation

$$P_w(t) \approx \frac{4}{\frac{n_r}{n_w} + \frac{n_w}{n_r} + 2} \sin^2\left(\frac{t}{2}\sqrt{\frac{n_r + n_w}{N} + \frac{\delta^2}{4}}\right) \tag{39}$$

Here N is the number of vertices in the graph and $\delta = [-\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}]$. Knowing that $\sin t$ has a maximum at $t = \frac{\pi}{2}$ one can see that the optimal time for a search is indeed of order $\mathcal{O}(\sqrt{N})$. The derivation of equation 39 can be seen in the supplementary materials from Qiang et al. [7].

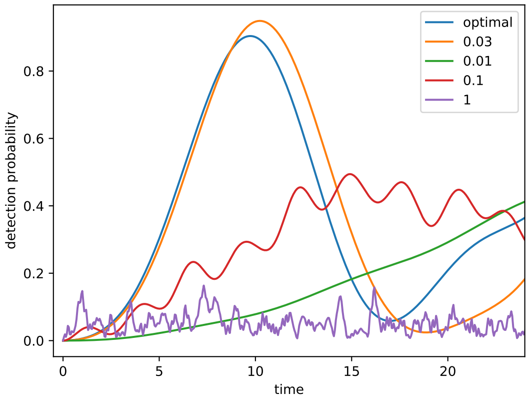


Figure 17: Detection probability in 3 random target states with 3 initial states for different values of kappa from equation 36. The graph used for this problem is an Erdös-Rényi$(64, 0.5)$ graph. The optimal value for kappa is calculated by $\frac{1}{\lambda_A}$ where λ_A is the largest eigenvalue of the adjacency matrix.

7.2 Method

The spatial search problem differs a bit from the other two problems in the sense that we do not calculate anything through the means of measurements on the CTQW, but we instead look at which time it would be best to perform a measurement. This optimal search time is only dependant on the structure of the graph, the value of κ in equation 36 and the starting and target states. However the probability to be in the target state is usually less than 1 even in the optimal search time and the optimal search time differs slightly for each graph. So we need a different approach in order to determine the effect of measurement uncertainty on the spatial search problem. The method that was chosen in this report is the following. The optimal search time is theoretically calculated for a large number of networks with the same properties, after which the average optimal search time is calculated. After this optimal search time is calculated, we perform a large number of measurements to detect if the particle is in the target position at this optimal time. With these measurements it will be possible to calculate the theoretical probability to find the targets at the optimal search time. After this process we once again perform a large number of measurements but this time we perform the measurements at $T + N(0, \sigma^2)$ with T the optimal search time. After having calculated both the theoretical and simulated detection probabilities we will calculate their error. This whole process will then be repeated for different size graphs and different numbers of target and detection states. In this report we will keep the number of target states and starting states the same and will only look at Erdös-Rényi graphs, but the same process can be used for other graphs.

7.3 Results

To determine the effect of uncertainty in the measurement time for the spatial search problem we first need to calculate the optimal search time for some graphs. The optimal search time is found by determining at which time the probability of finding the particle in the target state reaches its first maximum. For this report we repeat this process 1000 times for graphs with similar properties after which we take the average optimal search time. For this report it is chosen to only look at Erdös-Rényi graphs with $p = 0.5$, we look at graphs with $10, 20, 30, 40, 50, 75$ and 100 vertices. For each of these graphs we look at $1, 2$ or 3 target and initial states in the network. Table 10 shows the average optimal search time which is calculated from 1000 simulations for each test.

Table 10: The optimal search time for Erdös-Rényi graphs with $p = 0.5$ with either $1, 2$ or 3 target and initial states for a number of different sized graphs. The optimal search time is calculated by taking the average optimal search time for 1000 simulations with the specific type of graph and number of target and initial states. The number of states refers to both the number of target states and the number of initial states.

vertices	10	20	30	40	50	75	100
1 state	6.71 ± 1.51	9.93 ± 1.50	$12, 24 \pm 1.42$	13.79 ± 1.02	15.79 ± 1.21	18.85 ± 1.01	22.15 ± 1.19
2 states	5.24 ± 1.59	7.07 ± 1.41	8.54 ± 0.69	9.90 ± 0.73	11.08 ± 0.71	13.44 ± 0.70	14.58 ± 0.72
3 states	4.35 ± 1.50	5.69 ± 1.18	6.91 ± 0.53	8.04 ± 0.51	8.88 ± 0.42	10.95 ± 0.43	12.69 ± 0.43

These optimal search times are also plotted against the number of vertices in the network to show the relation between the search time and the number of vertices. In figure 18 it is clearly visible that this optimal search time scales with $\mathcal{O}(\sqrt{N})$ instead of the classical $\mathcal{O}(N)$.

With these optimal search times it is possible to calculate the detection probability on this optimal search time with and without uncertainty. To do this we first calculate the average detection probability on the optimal search time, which can be found in table 10. For each of these calculations we take the average over 5000 simulations. After that we calculate the average detection probability for the following uncertainties in time $1, 0.5, 0.1, 0.01$ and 0.001. For each of the 5000 simulation we also calculate the error between the theoretical result and the uncertain result of which the average is given in table 14. The following three tables show the detection probability on graphs of $30, 40$ and 50 vertices for $1, 2$ or 3 detection and starting states.

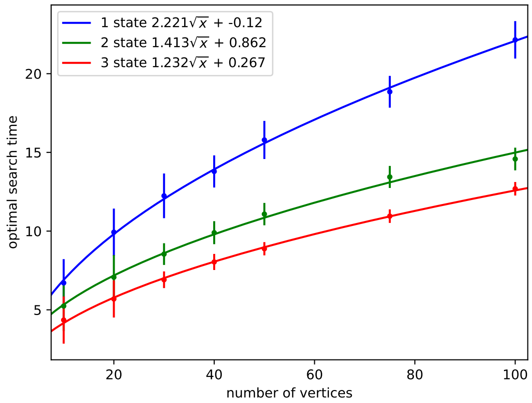


Figure 18: Optimal search times from table 10 with their uncertainties and a squareroot function fitted through the optimal search times.

Table 11: Average detection probabilities on the optimal search times from table 10 for Erdös-Rényi graphs of different sizes with $p = 0.5$ for 3 target and initial states. Each average is taken from 5000 simulations.

vertices	30	40	50
theoretical	82.6 ± 5.6	86.5 ± 4.5	88.9 ± 3.7
$\sigma^2 = 1$	76.1 ± 11.3	80.8 ± 9.7	84.1 ± 8.3
$\sigma^2 = 0.5$	80.1 ± 6.8	85.5 ± 5.5	87.7 ± 4.7
$\sigma^2 = 0.1$	82.6 ± 5.8	86.3 ± 4.6	88.8 ± 3.8
$\sigma^2 = 0.01$	82.6 ± 5.8	86.3 ± 4.6	88.8 ± 3.7
$\sigma^2 = 0.001$	82.6 ± 5.7	86.5 ± 4.5	88.9 ± 3.7

Table 12: Average detection probabilities on the optimal search times from table 10 for Erdös-Rényi graphs of different sizes with $p = 0.5$ for 2 target and initial states. Each average is taken from 5000 simulations.

vertices	30	40	50
theoretical	81.1 ± 7.0	85.2 ± 5.7	87.7 ± 4.7
$\sigma^2 = 1$	77.1 ± 10.2	81.3 ± 8.9	84.6 ± 7.4
$\sigma^2 = 0.5$	79.9 ± 7.8	84.2 ± 6.6	86.8 ± 5.5
$\sigma^2 = 0.1$	81.0 ± 7.3	85.0 ± 5.9	87.6 ± 4.8
$\sigma^2 = 0.01$	81.0 ± 7.2	85.1 ± 5.9	87.7 ± 4.7
$\sigma^2 = 0.001$	81.0 ± 7.1	85.1 ± 5.7	87.7 ± 4.7

Table 13: Average detection probabilities on the optimal search times from table 10 for Erdös-Rényi graphs of different sizes with $p = 0.5$ for 1 target and initial state. Each average is taken from 5000 simulations.

vertices	30	40	50
theoretical	75.2 ± 9.6	80.6 ± 8.2	83.9 ± 6.7
$\sigma^2 = 1$	72.8 ± 11.6	78.8 ± 9.7	82.3 ± 8.5
$\sigma^2 = 0.5$	74.7 ± 9.9	79.9 ± 8.6	83.4 ± 7.2
$\sigma^2 = 0.1$	75.2 ± 9.8	80.4 ± 8.4	83.8 ± 6.8
$\sigma^2 = 0.01$	75.2 ± 9.7	80.4 ± 8.3	83.8 ± 6.8
$\sigma^2 = 0.001$	75.2 ± 9.7	80.4 ± 8.3	83.9 ± 6.7

These results show that the detection probability does converge to the theoretical results for small values of σ^2. It is also clear that the detection probabilities of the simulations with an uncertainty are always lower than the detection probabilities on the optimal search time. This is also true for the probabilities which are similar in the tables, for these values the uncertain results are still smaller when we look at more decimals.

Table 14: Average error in % between the theoretical probabilities and the simulations for Erdös-Rényi grapsh of different sizes with $p = 0.5$ for $1, 2$ and 3 target and initial states. Each average is taken from 5000 simulations.

states-vertices	3-30	3-40	3-50	2-30	2-40	2-50	1-30	1-40	1-50
$\sigma^2 = 1$	9.0	7.4	6.1	6.9	6.0	4.7	6.5	4.9	4.0
$\sigma^2 = 0.5$	3.2	2.6	2.1	3.1	2.5	2.0	3.2	2.4	1.9
$\sigma^2 = 0.1$	0.55	0.46	0.36	0.61	0.46	0.37	0.61	0.46	0.38
$\sigma^2 = 0.01$	0.054	0.044	0.036	0.061	0.045	0.037	0.061	0.046	0.038
$\sigma^2 = 0.001$	0.0056	0.0045	0.0036	0.0059	0.0047	0.0037	0.0061	0.0047	0.0039

From table 14 it is clearly visible that we once again have the similar linear pattern between the error and the value of σ^2 for the smaller values. It is also visible that larger graphs have smaller errors. These results will be further discussed in section 9.

8 Implementation

8.1 Multiple particle walks and exponentially sized networks

Up until now we have only looked at the theory of the quantum random walk and not how this walk would be implemented in actual systems. In actual systems there are limitations in for example the size of the graph that can be generated, however there is a method to create larger graphs from multiple smaller graphs. We will now discuss how these graphs are generated and what their limitations are.

8.1.1 Exponentially sized networks

The construction of exponentially large networks follows the research of Wang et al. [15].

A quantum random walk on a single network operates in a N sized Hilbert space, in this subsection we will explain how this Hilbert space can be expanded exponentially by performing multiple quantum walks on small networks. Similar to the CTQW on a single graph we choose the Laplacian matrix to be the Hamiltonian of our system, however, one can also choose to use the adjacency matrix. This Laplacian will be the Hamiltonian for a single network, but to create the Hamiltonian for the exponentially sized network we need to take the Kronecker sum of our single Hamiltonians. This results in the following Hamiltonian of our large system

$$H^{(p)} = H_1 \oplus H_2 \oplus \cdots \oplus H_p \tag{40}$$

Where p is the number of smaller networks and H_i is the Laplacian matrix of network i. Where the Kronecker product of two Laplacian matrices is given by

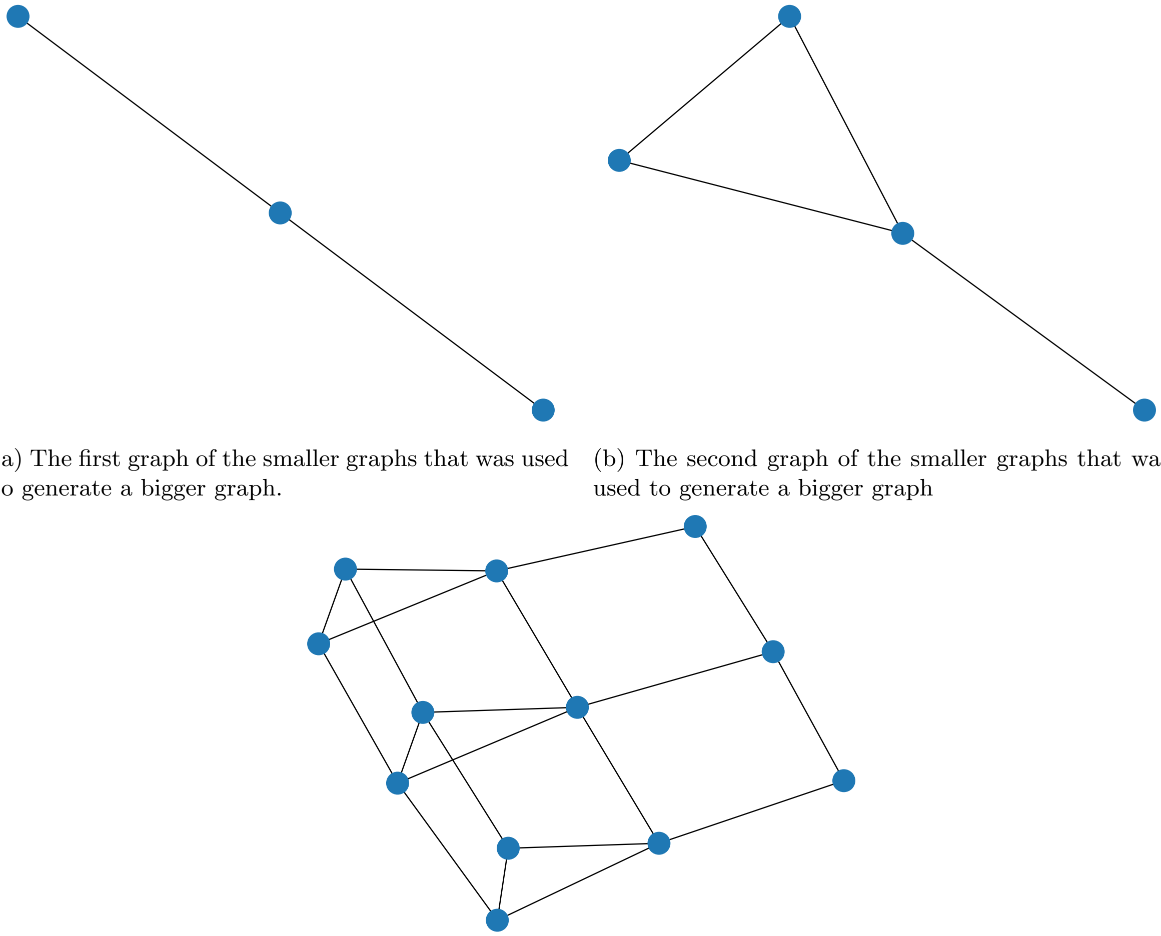
$$H_1 \oplus H_2 = H_1 \otimes I_{n_2} + I_{n_1} \otimes H_2 \tag{41}$$

Here I_n is the identity matrix of size n and n_1 and n_2 are the sizes of their respective matrices. This results in the following evolution operator $e^{-iH^{(p)}t}$. The Kronecker sum has a nice property which enables us to write the evolution operator as a tensor product of p individual evolution operators. This results in the following equation

$$U^{(p)}(t) = e^{-iH^{(p)}t} = e^{-iH_1 t} \otimes e^{-iH_2 t} \otimes \cdots \otimes e^{-iH_p t} \tag{42}$$

It is clear that a tensor product of p wave functions can be evolved by this operator, which results in the tensor product of p single particle quantum random walks on smaller networks. It is also clear that if each of the smaller networks has N vertices that each element of the tensor product has a corresponding Hilbert space of size N which results in the total Hilbert space of this system being N^p. This shows that we can generate exponentially large systems by performing multiple quantum random walks on smaller systems.

From these operations it is also clear that the exponentially sized network can be seen as a network which is constructed by the Laplacian matrix which is equal to Hamiltonian from equation 40. This method of generating a network does come with a limitation, namely not every network can be created with this method. This can be seen by the fact that the bottom left and top right elements of a matrix which is the result of a tensor product with an identity matrix will always be 0. This results in the fact that one can never generate a complete graph. Another way to see this is that the main diagonal which represents the degree of the vertices can never be higher than the sum of the maximal degrees of the single networks. This results in the degree of the exponentially large graph never exceeding half of the number of vertices of the new graph. Figure 19 shows an example of a graph generated from two smaller graphs with the method explained above.



(a) The first graph of the smaller graphs that was used to generate a bigger graph.

(b) The second graph of the smaller graphs that was used to generate a bigger graph

(c) The resulting multiplex graph.

Figure 19: A bigger graph generated from two smaller graphs by the method explained in section 8.1.1.

The construction listed above was for p fully distinguishable particles, but when the particles are fully indistinguishable the dimension of the Hilbert space will be less then N^p. The structure of the new graph is also dependent on the type of particles, to illustrate this we will construct a two-fermion and a two-boson graph.

Fermions are described by anti-symmetric states which results in no two fermions being able to occupy the same state. We will denote the states of the fermions by $|v, w\rangle$ where v works on the first graph and w on the second graph. We know that we can make any state with $v \neq w$ anti-symmetric in the following way

$$A\,|v, w\rangle = \frac{1}{\sqrt{2}}(|v, w\rangle - |w, v\rangle) \tag{43}$$

Since the particles are indistinguishable and can not occupy the same vertex we know that the possible states are given by $A\,|v, w\rangle$ where $v < w$ this results in $\binom{N}{2}$ possible states which forms our complete basis. The single particle Hamiltonian generated by a two-fermion CTQW is then given by

$$(H_f^{(2)})_{ij} = \left\langle F_i^{(2)} \middle| H_1 \oplus H_2 \middle| F_j^{(2)} \right\rangle \tag{44}$$

Here i goes from 0 to $\binom{N}{2}$ and the $\left|F^{(2)}\right\rangle$ states are the elements of the complete basis. This shows that a two-fermion CTQW on graphs of N vertices generate a CTQW on a graph with $\binom{N}{2}$ vertices.

The two-boson CTQW is generated in a similar way however bosons are symmetric which allows us to make the following states

$$B\left|v,w\right\rangle = \begin{cases} \frac{1}{\sqrt{2}}(\left|v,w\right\rangle + \left|w,v\right\rangle) & v \neq w \\ \left|v,v\right\rangle & v = w \end{cases} \tag{45}$$

It is clear that this results in N more possible states than the case of the two-fermion CTQW which results in $\binom{N+1}{2}$ states. The single particle Hamiltonian is then generated by the following equation

$$(H_b^{(2)})_{ij} = \left\langle B_i^{(2)} \middle| H_1 \oplus H_2 \middle| B_j^{(2)} \right\rangle \tag{46}$$

Here i goes from 0 to $\binom{N+1}{2}$ and the $\left|B^{(2)}\right\rangle$ states are the elements of the complete basis for the boson states. This results in the two-boson CTQW generating a graph with $\binom{N+1}{2}$ vertices. A further explanation on multiple particle CTQW for both fermions and bosons can be found in the research of Izaac and Wang [20].

8.1.2 Multiple particle walks

The properties of a multiple particle CTQW follows the theory from the supplementary material of Qiang et al. [7].

Now that we explained how larger networks can be simulated by multiple particle CTQW it is also important to know the basics of how these multiple particle walks work. We consider two indistinguishable particles a and b and start our CTQW in positions j and k. This results in the following initial position

$$\left|\psi_0^\phi\right\rangle = (\alpha a_j^\dagger b_k^\dagger + \beta e^{i\phi} a_k^\dagger b_j^\dagger)\left|0\right\rangle \tag{47}$$

Here $a_i^\dagger$ is the creation operator for position i and particle a, α and β are parameters which determine the entanglement of the two particles and ϕ represents the phase difference of the particles which is 0 for bosons and π for fermions. The evolution operator U can be split into two evolution operators one which works on particle a and one which works on particle b. This results in the following evolution of the initial state

$$U\left|\psi_0^\phi\right\rangle = (U^a \otimes U^b)\left|\psi_0^\phi\right\rangle = \sum_{s,t}(\alpha U_{s,j}^a U_{t,k}^b + \beta e^{i\phi} U_{s,k}^a U_{t,j}^b) a_s^\dagger b_t^\dagger)\left|0\right\rangle \tag{48}$$

This results in the following probability of finding particle a in r and particle b in q

$$P_{r,q}^\phi = |\left\langle 0\right| a_r b_q \sum_{s,t}(\alpha U_{s,j}^a U_{t,k}^b + \beta e^{i\phi} U_{s,k}^a U_{t,j}^b) a_s^\dagger b_t^\dagger)\left|0\right\rangle|^2 = |\alpha U_{r,j}^a U_{q,k}^b + \beta e^{i\phi} U_{r,k}^a U_{q,j}^b|^2 \tag{49}$$

Here a_i is the annihilation operator for position i and particle a. The α and β factors can also be described by a parameter for the indistinguishability γ of the particles this factor takes values between 0 and 1. Where 0 means that the particles are distinguishable and the 1 means that the particles are fully indistinguishable. Using this parameter the probability can be written in the following form

$$P_{r,q}^{\phi,\gamma} = \frac{1}{2}|\gamma(U_{r,j}^a U_{q,k}^b + e^{i\phi} U_{r,k}^a U_{q,j}^b) + \sqrt{1-\gamma^2} U_{r,j}^a U_{q,k}^b|^2 \tag{50}$$

The derivation of this expression can be found in the supplementary material of Qiang et al. [7]. This probability reduces to the single particle detection probability if we take $\gamma = 0$ and $a = b$, in this case we end up with the detection probability in a superposition of q and r with an initial position which is a superposition of k and l.

8.2 Implementation of the quantum random walk

We have now seen how the quantum random walk works and that there is a difference in how the walk behaves if there are multiple particles on the system. We also explained that there are methods to simulate larger networks when we can not build these networks, but it would also be good to know how these walks are implemented in the real world. In this subsection we will look at a succinct explanation on how these walks are constructed outside of the theory. The CTQW has been realized on a number of different systems such as trapped ions [21], photonics [7] and qubits [22].

This subsection will discus two different implementations of the CTQW in the real world. The first implementation is letting a CTQW walk on a two qubit system in the NV center of a diamond. In this NV center the electron spin is given by the spin triplet in its ground state which has 3 levels $m_S = -1, 0, 1$. The implementation in [22] uses the electron spin qubit and the ^{14}N nuclear spin qubit. THe ^{14}N nuclear spin qubit also has three option $m_l = -1, 0, 1$ for the implementation only 4 states are taken into account namely only the combinations with values of 0 and 1. By applying a laser with a wavelength of 532nm and a magnetic field that is aligned with the symmetry of the NV-axis the spin state is polarized to the spin state $|m_S, m_l\rangle = |0, 1\rangle$. After this state is initialized it will undergo the evolution operator which is described in equation 17 in section 3.2.2, this evolution operator is simulated by a set of rotations of the spins and multiple entangeling two-qubit gates. These rotations and entanglements are performed using the universal quantum cirquit described in [23]. The angles of rotation and planes on which the rotation takes place can be found in [22], these parameters determine the constructed network from the 4 states.

Another implementation which was recently constructed made use of silicon photonic quantum processor [7]. In this processor a laser with a wavelength of 1549.3nm is amplified and injected into a photonic chip where it split by grating couplers into two entangled photons where the photons relative phase is shifted. After these coupled photons have been prepared they each go to their own linear optical circuit. This results in the photons having an initial state given by equation

$$\left|\psi_0^{\phi,\theta}\right\rangle = (\cos\theta a_2^\dagger b_1^\dagger + e^{i\phi}\sin\theta a_1^\dagger b_j]2^\dagger)\,|0\rangle \tag{51}$$

This initial condition is generated with probability $\frac{1}{4}$. These photons will then evolve on their respective circuits. By letting the photons walk on two networks they simulate a bigger network with the method explained in the previous section. In [15] they show that they have now succeeded in creating a network with 20 vertices for each particle. These networks are controlled by thermo-optical phase shifters and multimode interferometers. With the setting of these elements on the linear optical circuits it is possible to create specific networks which also allows for certain larger networks to be created. The implementation on the photonic processor also has full control on the evolution speed γ from equation 13.

9 Discussion

From both the figures and the tables in sections 5.3, 6.3 and 7.3 we can see that an uncertainty in the measurement time or frequency does have an effect on the results. We will first look at this effect in further detail after which we will talk more about the choices that were made in order to get these results.

Looking at the error in the measurements with a certain uncertainty for the network centrality problem it is clear that there is no extremely large error not even when σ^2 is chosen to be the same as the measurement frequency. This is also what one would expect since the centrality measure takes the area under the graph of the detection probability. When we approximate this integral by a sum, we take the value of the function for evenly distributed distances and multiply these values by the distance to create thin rectangles. This approximation will be accurate when we take a large number of measurement. If we now introduce an uncertainty we would no longer have that these values are evenly distanced but rather some are closer to each other than others. However we will still take a large number of values over a long period of time which would still approximate the integral. From the results in the tables it is also clear that low values of σ^2 have a very small error and we also see that the error scales close to linearly for small values of σ^2.

For the graph isomorphism problem we will first look at the effects for isomorphic graphs after which we will look at the similar and randomly generated graphs. For the isomorphic graphs we see that large values for σ^2 also cause a large difference between the theoretical result and the result with an uncertainty. Due to the theoretical result always being 0 we can see these differences as the results with an uncertainty. We see that isomorphic graphs can still have a difference in graph certificates that is not 0. We see that for $\sigma^2 = 0.1$ we get values that are even higher than the theoretical difference in graph certificates for similar graphs, which can be seen in figure 15a. This shows us we need to have a small value of σ^2 to still be able to determine if two graphs are isomorph and not just similar. This also shows that the uncertainty does have a big impact in the graph isomorphism problem in contrast to the network centrality problem where large values for σ^2 still gave results with small errors.

If we now look at similar graphs we once again see that large values of σ^2 give very large errors which would make it difficult to determine if graphs are similar or perhaps isomorph. From both the results of the similar graphs and the isomorphic graphs we do however see the same linear depedence for small values of σ^2. This linearity is again visible for the randomly generated graphs, but we do see that the error is much smaller for these simulations. This is however very logical when we also look at the absolute difference, there we see that the difference between the theoretical and simulated calculations are almost the same for the similar and randomly generated graphs but because the difference in graph certificates is much larger for randomly generated graphs we end up getting a smaller error.

From table 9 it is clear that the absolute difference does not change significantly for larger graphs, but we do see that the error increases significantly for the similar graphs. This is logical since these graphs are even more similar to each other than in the case of the graphs with 50 vertices. This is due to the difference in the number of edges staying the same. For both graphs with 50 and 100 vertices being between 1 and 4 but the overall number of edges increases from 200 to 400. This means that there is a smaller percentage increase in the number of edges for the larger graphs.

For the spatial search problem we have two sets of results one being the average optimal search time which has no uncertainties and the difference in detection probability between the theoretical result and the simulated result. We will start by looking at the optimal search times. From table 10 it is clear that the optimal search time becomes much more uncertain for smaller graphs and fewer detection and initial states. This can be explained by the fact that there is a bigger discrepancy between advantageous couples of detection and initial states and disadvantageous couples. When the number of vertices increase we would see more couples of detection and initial states that are similar to each other. This results in the convergence of the optimal search time to a single value. In the supplementary material of Chakraborty et al. [19] it is shown that this optimal search time would go to a single value for all Erdös-Rényi graphs if N goes to infinity.

For the detection probability on these optimal search times it is clear that large values for σ^2 give a significant error, especially for smaller graphs. However, we do see that we converge to the theoretical result for smaller values of σ^2. Once again we see that we get a linear pattern between the value of σ^2 and the error for small values of σ^2. From the results it is also clear that the number of detection and initial states does not have a big influence on the error for small values of σ^2 but fewer target and starting states do lower the detection probability which is logical since it would be harder to find one state than one out of two states.

From all of these results there seems to be a clear linear pattern between the error and the value of σ^2, but for further research we would advise to perform more tests on different types of networks. Especially for the network centrality problem where we only took the average from 100 simulations. This small sample size was chosen because each centrality test consists of 10000 calculations for both the case with and without an uncertainty, performing these calculations for a large sample size would take very long on the computer that was used. We would advise to increase the sample size by performing the simulations on a stronger computer to more accurately determine the effect of measurement uncertainty for the network centrality problem. The simulations are also not performed on large networks due to the limitations of the computer that was used for these simulations. Therefore we would advise to further look into the effect of uncertainties in larger networks.

Another point that might be of interest for further research is looking at what the actual uncertainties are of implementations of the CTQW to determine the effect of these imperfections in the system on the results. The final point that would be interesting to look at for future research is the effect of other forms of uncertainties in the system because the uncertainty in the measurement time is not the only imperfection of the system. An example of this is not being able to determine the initial state with 100% certainty each time.

10 Conclusion

In this report we studied the effect of uncertainty in the measurement time and measurement frequency for three important problems, the network centrality problem, the graph isomorphism problem and the spatial search problem. We looked at the underlying theory of the continuous time quantum walk and the three problems after which we used this theory to simulate the theoretical outcomes. With these theoretical results we were able to simulate the continuous time quantum walk with an uncertainty in its measurement time which we then compared to these theoretical results. These uncertain simulations were performed for different levels of uncertainty in the three problems to study the effect of uncertainty in our system.

From our theoretical simulations we found that the centrality measure based on the CTQW is indeed a viable method to determine the centrality of each of the vertices. We also saw that in the theoretical case the method of calculating the difference between graph certificates is an accurate way to determine that graphs are not isomorphic and indeed always results in identical graph certificates for isomorphic graphs. Finally we saw that the optimal search time for our theoretical simulations indeed returns search times that are of $\mathcal{O}(\sqrt{N})$ instead of the classical $\mathcal{O}(N)$.

To answer our research question of what the effect of uncertainty in the measurement time is we considered two factors, the scaling of the error with uncertainty and the severity of the error based on the uncertainty.

For the network centrality it is clear that the error and absolute difference between the theoretical centrality measure and the simulated centrality measure with an uncertainty scale linearly for small uncertainties. This linear dependency does seem to fail for larger values of uncertainty. It is also visible that uncertainties do not introduce big errors in the network centrality problem, which is evident by the fact that an uncertainty which is equal to the measurement frequency itself still has relatively small errors. In conclusion for small levels of uncertainty we have a linear scaling in the error with the uncertainty and have a small error in the end result meaning that small uncertainties have little effect on the network centrality problem.

The graph isomorphism problem can be split into three categories, the isomorphic graphs, the similar graphs and the randomly generated graphs. For the isomorphic graphs we see that small levels of uncertainty scale linearly with the absolute difference between the theoretical outcome and the simulation. It is also clear that high levels of uncertainty have a severe impact on the error in our system. For the similar graphs we once again see the linear pattern for small levels of uncertainty and we have an even more severe impact on the error for large uncertainties. For the randomly generated graphs we see the linear pattern once again but in this case it is also somewhat visible for the larger uncertainties, in this case we also have a less severe impact in the error of our calculations for larger uncertainties. In conclusion we see a linear scaling between the error and the level of uncertainty for small uncertainties for each of the three cases and the error in our system is very large for large uncertainties resulting in it being almost impossible to determine the difference between similar and isomorphic graphs for high levels of uncertainty.

For the spatial search problem we once again see a similar linear dependence of the error on the level of uncertainty for small uncertainties. For this problem we also see that larger graphs have a smaller error in the final results than smaller graphs with the same uncertainty in the measurement time. It is also clear that we will have a smaller chance to actually detect our target when an uncertainty is introduced, but this error is relatively small for smaller level of uncertainty. So there is a less severe impact on the error from small uncertainties.

Overall we have seen that for each of the three problems we have a linear pattern between the error and the level of uncertainty for smaller uncertainties. However these errors are more severe for some problems than others. The network centrality only has a small impact in the error and the spatial search problem similarly has a relatively small impact on its error, while the graph isomorphism problem suffers from large errors due to the uncertainty. This means that for systems with a considerable uncertainty we will still be able to determine the centrality of vertices and perform spatial searches while it will be close to impossible to determine if two graphs are isomorphic or just similar.

References

[1] Danial A. Spielman. Pagerank and random walks on directed graphs - yale university, Oct 2010.

[2] Zhaobin Wang, Lijie Guo, Shuai Wang, Lina Chen, and Hao Wang. Review of random walk in image processing - archives of computational methods in engineering, Apr 2017.

[3] Wit Forys and Piotr Oprocha. Pseudo-random walks on graphs and cryptography. pages 16–19, 01 2009.

[4] Y. Aharonov, L. Davidovich, and N. Zagury. Quantum random walks. *Phys. Rev. A*, 48:1687–1690, Aug 1993.

[5] Julia Kempe. Quantum random walks hit exponentially faster, 2002.

[6] Xinying Li and Yun Shang. Improvement of quantum walk-based search algorithms in single marked vertex graphs, 2022.

[7] Xiaogang Qiang, Yizhi Wang, Shichuan Xue, Renyou Ge, Lifeng Chen, Yingwen Liu, Anqi Huang, Xiang Fu, Ping Xu, Teng Yi, Fufang Xu, Mingtang Deng, Jingbo B. Wang, Jasmin D. A. Meinecke, Jonathan C. F. Matthews, Xinlun Cai, Xuejun Yang, and Junjie Wu. Implementing graph-theoretic quantum algorithms on a silicon photonic quantum walk processor. *Science Advances*, 7(9):eabb8375, 2021.

[8] Lovász László, L. Lov, and Of Erdos. Random walks on graphs: A survey. pages 1–46, 01 1996.

[9] J Kempe. Quantum random walks: An introductory overview. *Contemporary Physics*, 44(4):307–327, jul 2003.

[10] Edward Farhi and Sam Gutmann. Quantum computation and decision trees. *Physical Review A*, 58(2):915–928, aug 1998.

[11] A. Didi and E. Barkai. Measurement-induced quantum walks. *Physical Review E*, 105(5), may 2022.

[12] Felix Thiel, Itay Mualem, Dror Meidan, Eli Barkai, and David A. Kessler. Dark states of quantum search cause imperfect detection. *Physical Review Research*, 2(4), oct 2020.

[13] H Friedman, D A Kessler, and E Barkai. Quantum renewal equation for the first detection time of a quantum walk. *Journal of Physics A: Mathematical and Theoretical*, 50(4):04LT01, dec 2016.

[14] Josh A. Izaac, Xiang Zhan, Zhihao Bian, Kunkun Wang, Jian Li, Jingbo B. Wang, and Peng Xue. Centrality measure based on continuous-time quantum walks and experimental realization. *Physical Review A*, 95(3), mar 2017.

[15] Yizhi Wang, Yingwen Liu, Junwei Zhan, Shichuan Xue, Yuzhen Zheng, Ru Zeng, Zhihao Wu, Zihao Wang, Qilin Zheng, Dongyang Wang, Weixu Shi, Xiang Fu, Ping Xu, Yang Wang, Yong Liu, Jiangfang Ding, Guangyao Huang, Chunlin Yu, Anqi Huang, Xiaogang Qiang, Mingtang Deng, Weixia Xu, Kai Lu, Xuejun Yang, and Junjie Wu. Large-scale full-programmable quantum walk and its applications, 2022.

[16] Phillip Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.

[17] Albert-László Barabási, Sep 2014.

[18] László Babai. Graph isomorphism in quasipolynomial time, 2016.

[19] Shantanav Chakraborty, Leonardo Novo, Andris Ambainis, and Yasser Omar. Spatial search by quantum walk is optimal for almost all graphs. *Physical Review Letters*, 116(10), mar 2016.

[20] J. A. Izaac and J. B. Wang. Systematic dimensionality reduction for continuous-time quantum walks of interacting fermions. *Phys. Rev. E*, 96:032136, Sep 2017.

[21] F. Zähringer, G. Kirchmair, R. Gerritsma, E. Solano, R. Blatt, and C. F. Roos. Realization of a quantum walk with one and two trapped ions. *Phys. Rev. Lett.*, 104:100503, Mar 2010.

[22] Maimaitiyiming Tusun, Yang Wu, Wenquan Liu, Xing Rong, and Jiangfeng Du. Experimental implementation of a continuous-time quantum random walk on a solid-state quantum information processor*. *Chinese Physics B*, 28(11):110302, nov 2019.

[23] Yang Wu, Ya Wang, Xi Qin, Xing Rong, and Jiangfeng Du. A programmable two-qubit solid-state quantum processor under ambient conditions. *npj Quantum Information*, 5(1), jan 2019.

[24] Ronan den Ouden. Quantum-random-walk-bep. https://github.com/Ronandenouden/Quantum-random-walk-BEP/blob/main/continuous%20walk.py, 2023.

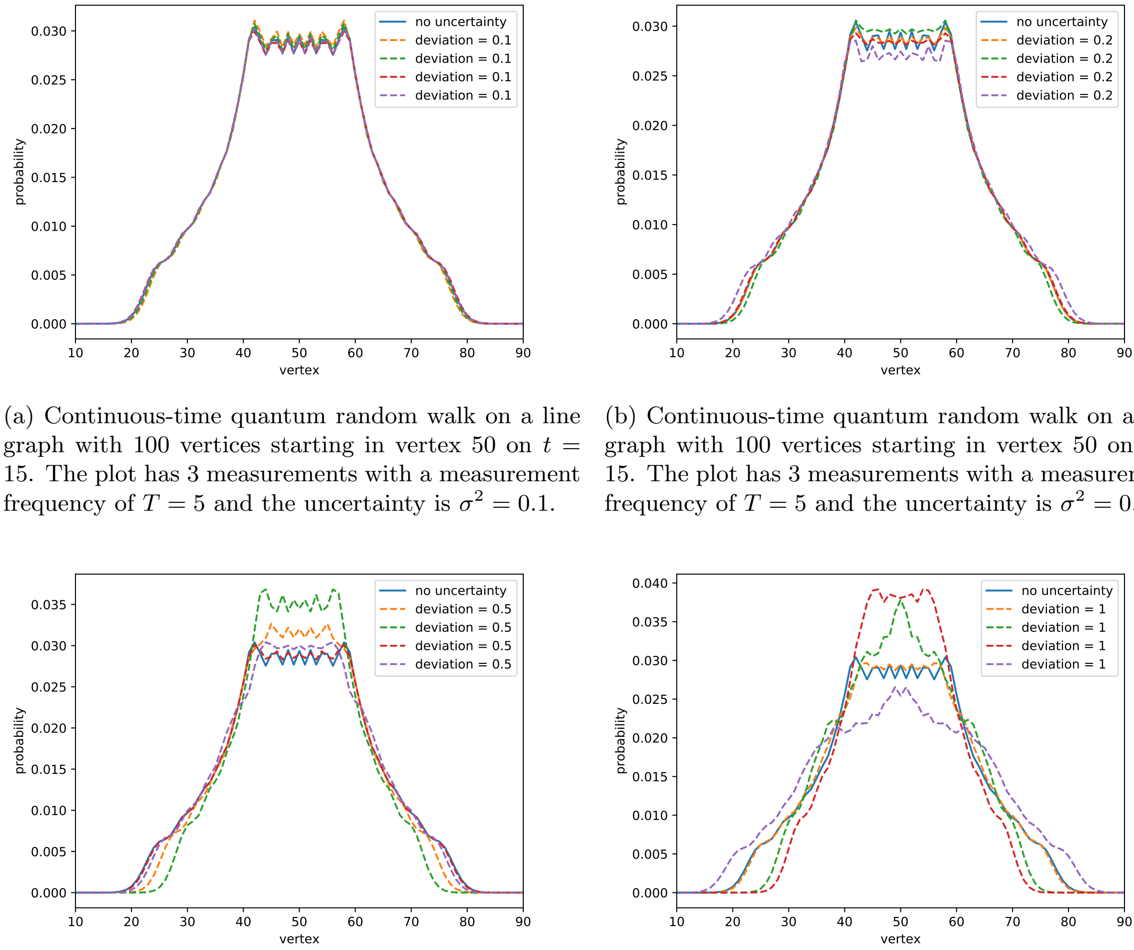
A Appendix

A.1 Code

The code that was used for this report can be found in the following link.

A.2 Extra results

Some extra figures that show the effect of uncertainty in the measurement time and frequency for other problems.



(a) Continuous-time quantum random walk on a line graph with 100 vertices starting in vertex 50 on $t = 15$. The plot has 3 measurements with a measurement frequency of $T = 5$ and the uncertainty is $\sigma^2 = 0.1$.

(b) Continuous-time quantum random walk on a line graph with 100 vertices starting in vertex 50 on $t = 15$. The plot has 3 measurements with a measurement frequency of $T = 5$ and the uncertainty is $\sigma^2 = 0.2$.

(c) Continuous-time quantum random walk on a line graph with 100 vertices starting in vertex 50 on $t = 15$. The plot has 3 measurements with a measurement frequency of $T = 5$ and the uncertainty is $\sigma^2 = 0.5$.

(d) Continuous-time quantum random walk on a line graph with 100 vertices starting in vertex 50 on $t = 15$. The plot has 3 measurements with a measurement frequency of $T = 5$ and the uncertainty is $\sigma^2 = 1$.

Figure 20: Continuous-time quantum random walks for 4 different measurement uncertainties after $t = 15$ with a measurment frequency of $T = 5$.

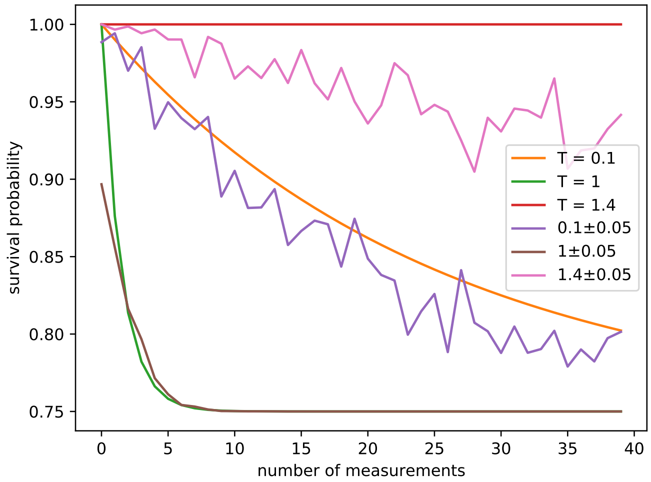


Figure 21: The survival probability from equation 21 for different measurement intervals for the first 100 measurements with and without an uncertainty. The starting state is the central vertex of figure 8b and the target vertex is the top right vertex

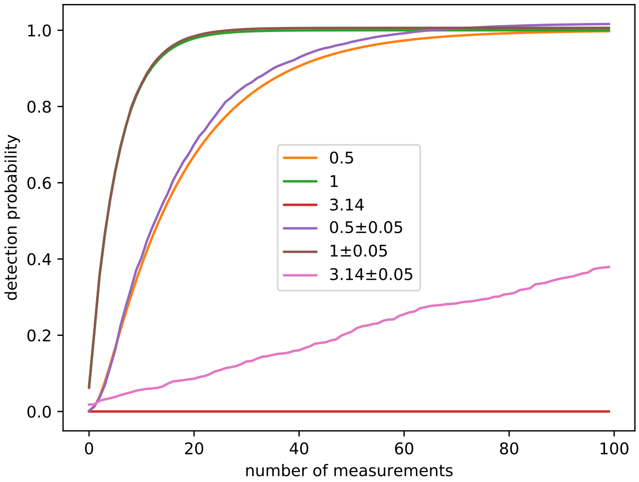


Figure 22: The cumulative detection probability for the first 100 measurements for different measurement intervals with and without an uncertainty. The graph used for this plot is a cycle graph with 6 vertices and the target and starting positions are positioned on the opposite site of the cycle, so vertex 1 and 4.

A.3 Graphs used for calculations

This section shows some examples of the types of graphs that were used for this report.

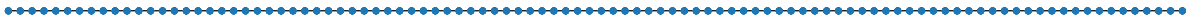


Figure 23: 1D path graph with 100 vertices.

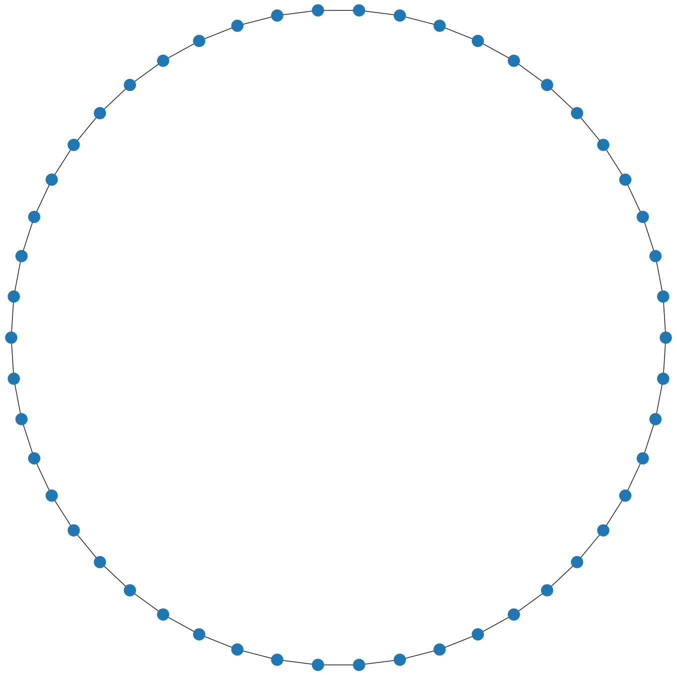


Figure 24: Cycle graph with 50 vertices.

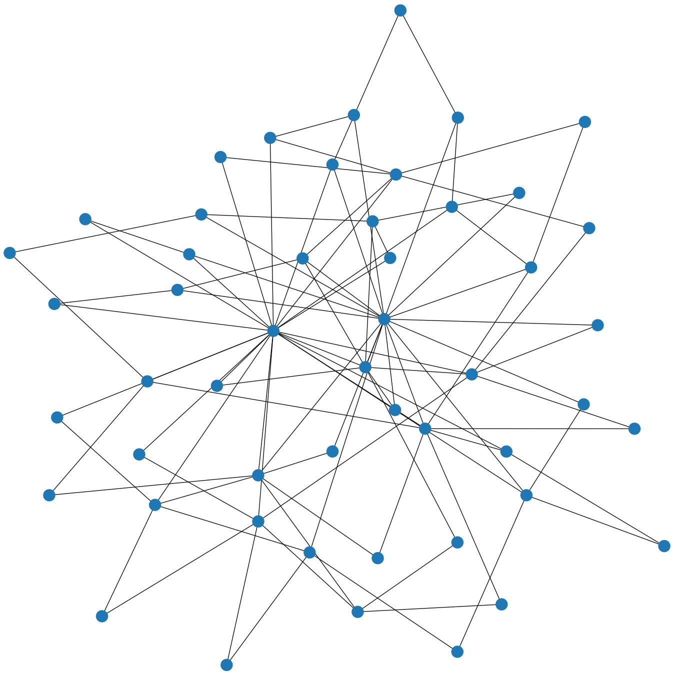


Figure 25: Barabasi-Albert graph with 50 vertices and degree 2 for each newly added vertex.

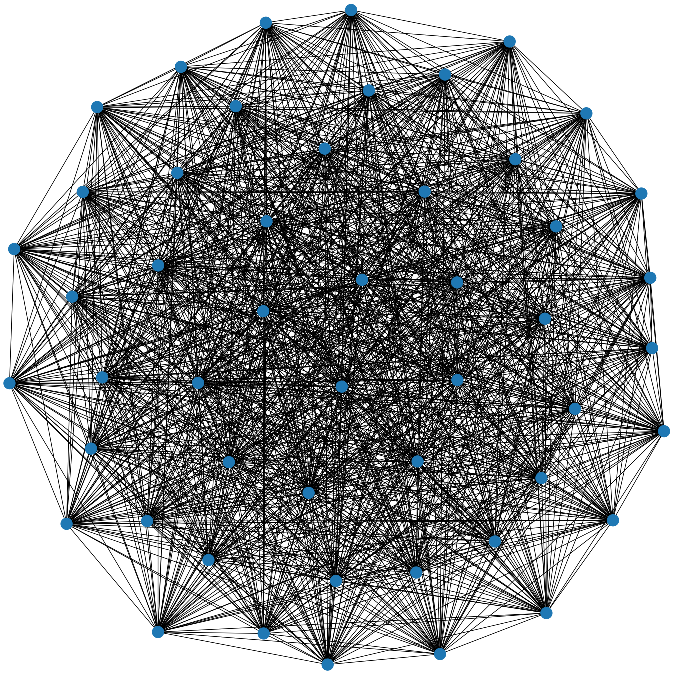


Figure 26: Complete graph with 50 vertices.

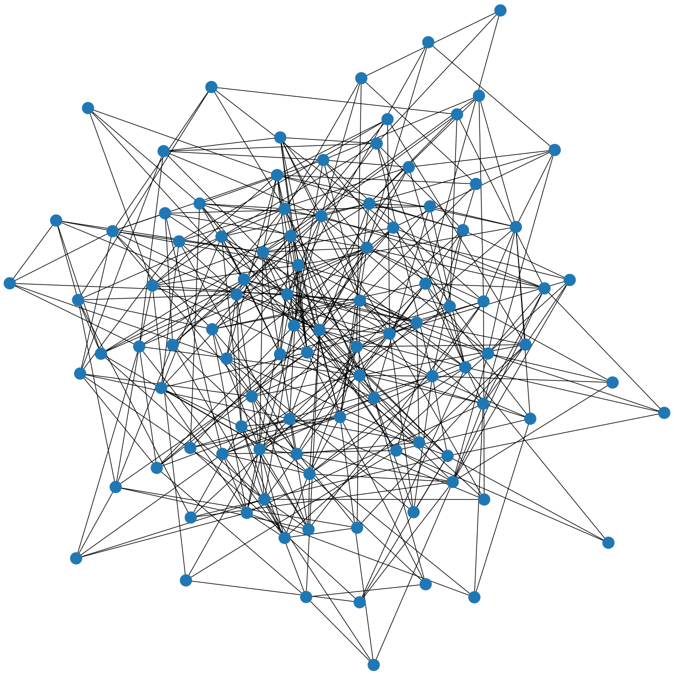


Figure 27: Random graph with 100 vertices and 400 edges.

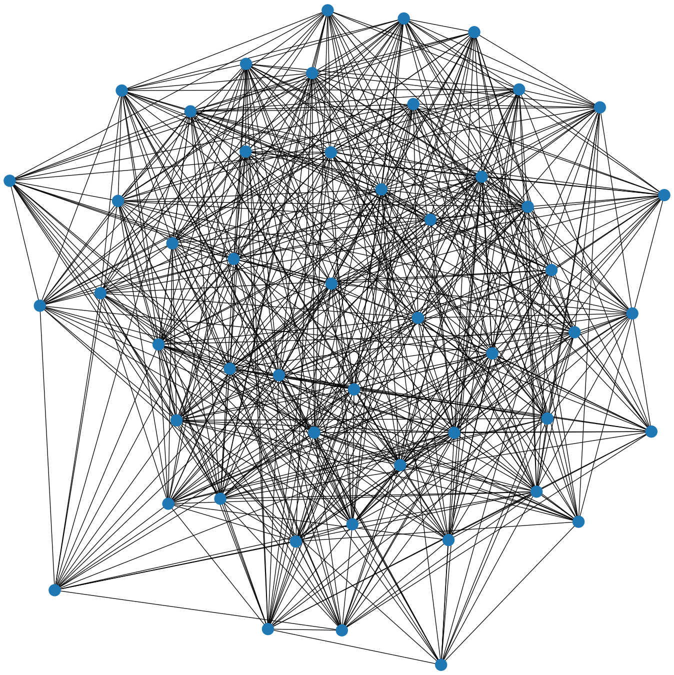


Figure 28: Erdös-Rényi graph with 50 vertices and $p = 0.5$.