Transferring Entanglement Between Different Dimensions

Timothy Forrer
Level 4 Project, MSci Natural Sciences
Supervisor: Dr V. Kendon
Department of Physics, Durham University

Submitted: April 18, 2021

Abstract

Quantum random walks have been an area of significant interest over the past two decades within quantum computation and information. They exhibit many desirable qualities that aid quantum computations and are extremely versatile for use in the design of quantum algorithms. In this report I analyse a recent publication that uses quantum walk dynamics in order to generate higher dimensional entangled states.

Contents

1 Introduction

Quantum computing is a field of research that has gained increasing attention over the course of the 21$^{\text{st}}$ Century. Loosely speaking, it is a field that seeks to examine how quantum phenomena can be exploited to solve computation problems. Quantum computers are of particular interest due to the class of problems they can solve faster than possible classically. For example, Shor's factorising algorithm [1] gives a recipe for a quantum computer to calculate the factors of a number. This is of particular significance as there are cryptographic protocols that secure information by relying on the difficulty of factorising large numbers which are the products of two primes, a difficulty that is bypassed relatively easily with a quantum computer. Furthermore, as classical computers are made more and more computationally powerful, their physical power requirements go up also. Quantum computers offer a potential avenue to completing computations that require this additional classical computing power at a reduced energy cost.

Many schemes for quantum computation utilise qubits, the quantum analogue of the bit which is the unit of information in classical computing. The main difference between the bit and the qubit, is that the bit can only exist in one of two states (0 or 1) at any one time, whereas the qubit, due to its quantum nature, can exist in superposition of both the possible states. Other models utilising qudits, which exist in superposition of d states rather than just two, have been proposed as they can unlock further advantages at the cost of being more complex to implement physically.

In addition to superposition, another well known quantum phenomena that is often taken advantage of in quantum computing is *entanglement*, correlations present in quantum systems that are far stronger than possible to find in classical systems. There are a variety of protocols that require the presence of entangled qubits in order to achieve results not possible with classical computers, for example superdense coding [2], quantum key distribution [3] and quantum teleportation [4]. Higher dimensional entanglement, which in this report is taken to be entanglement between qudits, further enhances the power of quantum algorithms, such as superdense coding [5]. As such the ability to possess and manipulate entangled states in higher dimensions has further benefits but again comes with its own challenges.

Quantum walks (QWs) are powerful tools in the landscape of quantum computing. Much like their classical analogue in the classical random walk, they exhibit many properties that are desirable for computations and are an extremely useful building block for many algorithms designed for quantum computers [6]. Specific research interest into the quantum variant stems from their very significant divergences from the classical random walk, including different spreading speeds and ability to traverse multiple paths at once. Their power is such that QWs can simulate any quantum computation and therefore are a model for universal quantum computing [7]. There is also evidence of robust performance even when the quantum computer is not perfectly isolated from its environment, and in certain situations it has been shown that decoherences due to interactions with the environment is beneficial for a given computation [8]. Quantum walks are divided into two categories, *discrete* and *continuous* time, these labels describing the nature of the evolution of the walker as the quantum walk progresses. Continuous time QWs have been shown to solve a wide range of problems in a number of different settings, in some cases exponentially faster than a classical computer is able to [9], but in this report it will be discrete time QWs that are of interest.

An alternative universal quantum computing model is *ancilla-based quantum computing* (AQC), which is a model that aims to reconcile the conflicting demands in building a quantum computer. Qubits need to be well isolated to prevent decoherence, but doing so also makes them harder to interact with. AQC resolves this by utilising two different kinds of qubits, ones which are well isolated as a main register, and an ancilla register of qubits that are easier to manipulate but whose states decohere faster. By delocalising information across the ancilla and main register, computations can be performed on the ancilla register before the information is relocalised on the long-lived main register qubits and the ancilla qubits can be reset or replaced to be used for further computation.

A potential solution to the demanding task of generating higher dimensional entanglement has been proposed [10] which uses the dynamics of discrete QWs to transfer lower dimensional entanglement between qubits, which is far simpler to generate, into the high dimensional qudits. Whilst this scheme can be used to some moderate degree of success, the underlying priciples of the QW protocol can be adapated to create a scheme based on AQC, which helps overcome some of the issues posed by the QW based protocol.

In this report, a primer on quantum computing, entanglement and two models of quantum computing, quantum walks and ancilla-based quantum computing, is given in section 2. Following this, section 3 focusses on the protocol that uses discrete QW dynamics to facilitate the transfer of entanglement, in particular analysing its efficiency in achieving the aim of entanglement transfer. The AQC scheme for entanglement transfer is presented in section 4. Further uses of the AQC scheme beyond the transfer of entanglement are presented in section 5. Finally, a discussion on the results of the two schemes, potential directions for further work and a concluding remarks are presented in section 6.

2 Background

2.1 Quantum Computation

2.1.1 Qubits

A qubit can be described by a state belonging to a Hilbert space $\mathcal{H}$ of dimension two. A standard basis for $\mathcal{H}$ is the *computational basis*, labelled by $|0\rangle$ and $|1\rangle$. Naturally we can also use other bases for describing our qubit state. A common alternative is the basis $\{|+\rangle, |-\rangle\}$. This basis is best known as the *Hadamard basis*, as $|+\rangle$ and $|-\rangle$ are equal to $H|0\rangle$ and $H|1\rangle$, where H is the Hadamard transform given by

$$|+\rangle = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{1}$$

$$|-\rangle = H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{2}$$

Qubits can also be written out in vector form, and the natural choice for this is to assign the computational basis states as

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \; |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}. \tag{3}$$

In this basis the Hadamard transform is given by

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}. \tag{4}$$

2.1.2 Qudits

In quantum computing, or indeed classical computing, there is no physical constraint that necessitates the use of two level qubits or bits. In classical computing, the bit can be generalised to a unit that takes on one of d states, known as the *dit*. Similarly, the quantum dit is the *qudit*, which can be in superposition of up to d states. Many aspects of quantum computing specific to qubits generalise nicely to qudits. The computational basis is now extended from $\{|0\rangle, |1\rangle\}$ to $\{|j\rangle\}_{j=0}^{d-1}$. To distinguish kets belonging to Hilbert spaces of different dimension, the notation $|\cdot\rangle_d$ will be used to denote a state belonging to $\mathcal{H}_d$, a Hilbert space of dimension d. For compactness, operators will not have their dimension explicitly labelled, since they belong to the same Hilbert space as the state they are acting on. Similar to qubits, there are many different bases that can be used to describe our qudit states. The d dimensional extension of the Hadamard transform is the *quantum Fourier transform*, given by

$$F|j\rangle_d = \frac{1}{2^{\frac{d}{2}}}\sum_{k=0}^{d-1}\omega^{jk}|k\rangle_d, \tag{5}$$

where $\omega = e^{i\frac{2\pi}{d}}$, i.e. it is the d^{th} root of unity. From this it can be seen that for $d = 2$, $F = H$. Therefore the Hadamard basis $\{|+\rangle, |-\rangle\}$ can be generalised to the set of states $\{|+_j\rangle\}_{j=0}^{d-1}$, where

$$|+_j\rangle_d = F|j\rangle_d. \tag{6}$$

Again, qudits can also be represented by vectors, with the computational basis states again assigned as

$$|0\rangle = \begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix}, \; |1\rangle = \begin{pmatrix}0\\1\\\vdots\\0\end{pmatrix}, \ldots, \; |d-1\rangle = \begin{pmatrix}0\\0\\\vdots\\1\end{pmatrix}. \tag{7}$$

In this basis, the Fourier transform is given by the matrix

$$F = \begin{pmatrix}1 & 1 & \cdots & 1\\1 & \omega & \cdots & \omega^{d-1}\\\vdots & \vdots & \ddots & \vdots\\1 & \omega^{d-1} & \cdots & \omega^{(d-1)^2}\end{pmatrix}, \tag{8}$$

and the associated Fourier basis states are found by applying the matrix representation of F to the computational basis state vectors.

2.1.3 Collections of Qudits

Qudits considered in isolation are not overly useful for quantum computations and in general a collection of qudits is needed for a given algorithm. The Hilbert space of a collection of n qudits, which are not neccessarily all of the same dimension, is found by taking the tensor product, $\otimes$, of the Hilbert spaces of each of the individual qudits

$$\mathcal{H} = \mathcal{H}_{d_0} \otimes \mathcal{H}_{d_1} \otimes \cdots \otimes \mathcal{H}_{d_{n-1}}. \tag{9}$$

The computational basis state for the combined Hilbert space is found by taking all possible tensor product combinations of the basis states for each constituent Hilbert space. The state

$$\left|0\right\rangle_{d_0} \otimes \left|0\right\rangle_{d_1} \otimes \cdots \otimes \left|0\right\rangle_{d_{n-1}} \tag{10}$$

is an example of such a basis state. Notationally, this state can be further simplified by the omission of the tensor product symbol,

$$\left|0\right\rangle_{d_0} \left|0\right\rangle_{d_1} \ldots \left|0\right\rangle_{d_{n-1}}. \tag{11}$$

If the kets are of the same dimension then this can be further compactified by contracting them,

$$\left|0\right\rangle_{d} \left|0\right\rangle_{d} \ldots \left|0\right\rangle_{d} = \left|00\ldots0\right\rangle_{d}. \tag{12}$$

Since tensor products distribute over sums, when combining states in superposition they must also be distributed over. For example with two qudits,

$$\begin{aligned} (\left|0\right\rangle_d + \left|1\right\rangle_d) \otimes \left|2\right\rangle_d &= (\left|0\right\rangle_d + \left|1\right\rangle_d)\left|2\right\rangle_d & (13)\\ &= \left|02\right\rangle_d + \left|12\right\rangle_d & (14)\end{aligned}$$

Recall also from the properties of the tensor product that it is not commutative, therefore care must be taken in ensuring that the order of qudits is preserved throughout,

$$\left|01\right\rangle_2 \neq \left|10\right\rangle_2. \tag{15}$$

In this report all three notational methods are used, with clarity given due preference over compactness. The subscripts are often dropped when the meaning is clear.

2.1.4 Operators

There are a wide variety of other operators beyond those introduced above that equate to a change of bases. One of the most common is the (Pauli) X, or NOT, operator. As the second name implies, in the qubit setting this acts in the same way as the classical NOT operator,

$$\begin{aligned} \left|0\right\rangle &\mapsto \left|1\right\rangle & (16)\\ \left|1\right\rangle &\mapsto \left|0\right\rangle. & (17)\end{aligned}$$

More generally in the qudit setting,

$$X\left|j\right\rangle_d = \left|j+1 \text{ mod } d\right\rangle_d . \tag{18}$$

However, there are other gates that do not have a classical analogue such as the the (Pauli) Z gate. In the qubit setting Z acts as the identity on $|0\rangle$ and applies a relative phase to $|1\rangle \mapsto -|1\rangle$. Again this can be generalised to the qudit setting

$$Z\left|j\right\rangle_d = \omega^j \left|j\right\rangle_d , \tag{19}$$

where ω is again the d^{th} root of unity. Both of these operators are sometimes referred to as Pauli operators as their matrix representations in the computational basis are given by the Pauli matrices σ_x and σ_z. There is also the Y operator corresponding to the Pauli σ_y matrix but this is often omitted from consideration as

$$ZX = iY. \tag{20}$$

Hence when designing a set of operators for a quantum computer, there is no theoretical need for Y as it already exists up to a global phase, which is essentially irrelevant. The final operation to be introduced in the section is the C-U, or controlled-U operator, where U is any unitary operation. This operator is distinct from the others previously introduced in that it acts on two qudits rather than just one, although it only directly alters the state of one of the two qudits. In the qubit setting, one qubit is designated as the control qubit and the other the target. If the control qubit is in the state $|0\rangle$, then nothing happens to the target qubit. If the control qubit is in the state $|1\rangle$, then the operator U is applied to the target qubit. Written mathematically, where the left hand qubit is the control, the right hand qubit is the target in some arbitrary qubit state $|\psi\rangle$,

$$C\text{-}U\left(|0\rangle \otimes |\psi\rangle\right) = |0\rangle \otimes |\psi\rangle \tag{21}$$

$$C\text{-}U\left(|1\rangle \otimes |\psi\rangle\right) = |1\rangle \otimes U\left|\psi\right\rangle . \tag{22}$$

To extend this to the qudit setting, first note that equations 21-22 can be compactly written as

$$C\text{-}U\left(|j\rangle \otimes |\psi\rangle\right) = |j\rangle \otimes U^j\left|\psi\right\rangle . \tag{23}$$

This equation now is well-defined for qudits, where instead of just taking values 0 or 1, j can now take values from $\{0, 1, \ldots, d-1\}$. Note that this definition applies also when the dimensions of the control and target are not equal, provided that the action of U on the target is well-defined

$$C\text{-}U\left(|j\rangle_d \otimes |\psi\rangle_{d'}\right) = |j\rangle_d \otimes U^j\left|\psi\right\rangle_{d'} . \tag{24}$$

In this report, unless otherwise stated, it is assumed that control qudits are always on the left and target qubits on the right.

It is sometimes neccessary to have operators that act on single qudits in a collection of n qudits. These can be expressed by taking the tensor product with the identity acting on the other qudits in the collection. For example, the operator that acts with an X on the qudit of index 1 in the state $|\psi_0\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle$ is given by

$$I \otimes X \otimes I. \tag{25}$$

2.1.5 Circuits and Gates

When describing a circuit that implements a series of operators on our qudits, it is often much clearer and simpler to use a circuit diagram. Straight lines in the circuits represent qudits. Operations on qudits are represented by gates placed on the straight lines corresponding to the qudits we wish to act on. A table of common gates relevant for this report is given in table 1.

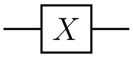
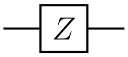
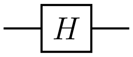
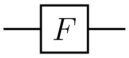
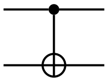
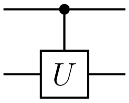
Operator	Gate
(Pauli) X	
(Pauli) Z	
Hadamard	
QFT	
C-X	
C-U	

Table 1: Table of operators and their gate representations in quantum circuits. The C-X gate can also be written in general C-U form.

When reading a circuit, time flows from left to right and a gate must finish its operation on a qudit before the next gate on the same qudit is enacted. Figure 1 is an example of a two qubit circuit implementing the following series of operations

$$C\text{-}X(Z \otimes I)(X \otimes I)\left(|0\rangle_2 \otimes |\psi\rangle_2\right). \tag{26}$$

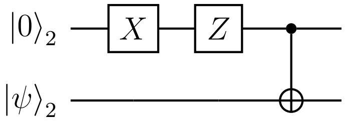


Figure 1: An example of a circuit representing the expression given in equation 26.

2.2 Entanglement

Entanglement is a property of quantum systems that is the source of many of the remarkable results displayed by quantum algorithms. Loosely, it is the presence of strong correlations in a quantum system, much stronger than possible classically. Strictly speaking, an entangled state is one that cannot be expressed as a separable state. A separable state is one that can be written as a tensor product of states in the Hilbert subspaces of a multipartite system. All of the basis states formed in section 2.1.3 are examples of separable states, as they are formed by taking the tensor product of basis states in each of the constituent Hilbert spaces. An example of an entangled state would be this state between two qubits,

$$\frac{1}{\sqrt{2}}(\left|00\right\rangle_2 + \left|11\right\rangle_2). \tag{27}$$

It is impossible to write this state in the form $(a\left|0\right\rangle_2 + b\left|1\right\rangle_2) \otimes (c\left|0\right\rangle_2 + d\left|1\right\rangle_2)$, hence it is not separable. This is one of the four Bell states which are the *maximally entangled* two qubit states.

Definition 2.1 (Maximally Entangled State)**.** *A state $\left|\psi\right\rangle$ describing a bipartite system $\mathcal{H} = \mathcal{H}^{(A)} \otimes \mathcal{H}^{(B)}$ is maximally entangled if the reduced density matrix representing $\left|\psi\right\rangle$ in either subspace is maximally mixed, that is, it is directly proportional to the identity.*

Taking the Bell state given in equation 27, its density matrix is given by

$$\rho = \frac{1}{2}\left(\left|00\right\rangle\left\langle00\right| + \left|00\right\rangle\left\langle11\right| + \left|11\right\rangle\left\langle00\right| + \left|11\right\rangle\left\langle11\right|\right) \tag{28}$$

$$= \frac{1}{2}\left(\left|0\right\rangle\left\langle0\right| \otimes \left|0\right\rangle\left\langle0\right| + \left|0\right\rangle\left\langle1\right| \otimes \left|0\right\rangle\left\langle1\right| + \left|1\right\rangle\left\langle0\right| \otimes \left|1\right\rangle\left\langle0\right| + \left|1\right\rangle\left\langle1\right| \otimes \left|1\right\rangle\left\langle1\right|\right) \tag{29}$$

The reduced density matrix in $\mathcal{H}^{(A)}$ is therefore

$$\rho^{(A)} = \mathrm{Tr}_B(\rho) = \frac{1}{2}\left(\left|0\right\rangle\left\langle0\right| + \left|1\right\rangle\left\langle1\right|\right) = \frac{1}{2}I, \tag{30}$$

where Tr_B denotes the partial trace over subspace $\mathcal{H}^{(B)}$.

2.2.1 Higher Dimensional Entanglement

Generalising entanglement to higher dimensions in the bipartite setting is done by having entanglement between qudits as opposed to qubits. There is no requirement for the dimensions of the entangled qudits to be matching. For example,

$$\left|\psi\right\rangle = \frac{1}{\sqrt{2}}\left(\left|0\right\rangle_2 \otimes \left|0\right\rangle_3 + \left|1\right\rangle_2 \otimes \left|1\right\rangle_3\right), \tag{31}$$

is a entangled state between a qubit and a qutrit ($d = 3$). Whilst higher dimensional entanglement can also refer to entanglement between multipartite systems, such as GHZ states or W states [11], in this report only bipartite entanglement is of concern.

Higher dimensional entanglement is useful as it contains stronger correlations than present in a Bell pair and can be utilised to further enhance the power of quantum computations. For example, the superdense coding protocol can transmit two classical bits of information for every Bell pair available. If higher dimensional entangled states are used instead, then more bits of classical information can be transmitted per entangled state [5], making the coding even denser. However, whilst higher dimensional entanglement unlocks further benefits, it comes with further challenges in its creation. Many schemes for physically generating entangled Bell pairs exist (see for example [12–14]) if it is instead possible to transfer and accumulate entanglement from Bell pairs into qudit pairs, then this does away with the need for different entanglement generation schemes for each dimension of qudit, and instead the same scheme can be used to generate Bell pairs and transfer this entanglement to the qudit pairs. In a practical setting this is also desirable, as only a source of Bell pairs is needed to generate entangled qudits of any dimension, saving on the space required in a quantum computer should different dimensions of entanglement be required for different computations. Furthermore, if the entanglement can be freely transferred between dimensions, that is, not just accumulated in but also retrieved from higher dimensional entangled states, then this would further enhance the versatility of quantum computers, giving them a store of entanglement that can be drawn from when required.

2.2.2 Measuring Entanglement

In order to analyse the efficiency of an entanglement transfer protocol, it is useful to have a method of measuring entanglement. Numerous methods of doing so have been proposed and are used, and a full mathematical discussion on entanglement measures is given in [15]. In this report, the measure of entanglement used is the *logarithmic negativity*, presented by Vidal [16].

Definition 2.2 (Logarithmic Negativity)**.** *Let ρ be a density matrix describing a bipartite quantum system $\mathcal{H} = \mathcal{H}^{(A)} \otimes \mathcal{H}^{(B)}$. The logarithmic negativity, $E_{\mathcal{N}}(\rho)$, of ρ is defined as*

$$E_{\mathcal{N}}(\rho) = \log_2 \left\| \rho^{\Gamma_A} \right\|_1 . \tag{32}$$

where X^{Γ_A} denotes the partial transpose of X with respect to the subsystem $\mathcal{H}^{(A)}$ and $\|X\|_1 = Tr(\sqrt{X^\dagger X})$ denotes the trace norm of X.

The logarithmic negativity is also referred to as the log negativity. It is an appropriate choice for the purpose of this report as it is an entanglement measure that is computable for mixed as well as pure states, and satisfies (strong) additivity

$$E_{\mathcal{N}}(\rho \otimes \sigma) = E_{\mathcal{N}}(\rho) + E_{\mathcal{N}}(\sigma), \tag{33}$$

where ρ, σ are the density matrices describing two bipartite systems. A proof for this is given in [16]. Another useful property of log negativity is that it is easy to compute, particularly in the case that a state is maximally entangled in a Hilbert space of some dimension $d \times d$ (not neccessarily the dimension of the full Hilbert space in which the bipartite system resides).

Claim 2.3. *A maximally entangled state in $d \times d$ dimensions has a log negativity of $\log_2 d$.*

Proof. First, note that for a Hilbert space of dimension $d \times d$, the state

$$|\psi\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} |j\rangle \otimes |j\rangle \in \mathcal{H}^{(A)} \otimes \mathcal{H}^{(B)} \tag{34}$$

is maximally entangled, since its reduced density matrix in either subspace is maximally mixed. So to calculate the log negativity of any maximally entangled state it is sufficient to calculate the log negativity of the density matrix, ρ associated with $|\psi\rangle$. Since $|\psi\rangle$ is a pure state,

$$\begin{aligned}\rho &= |\psi\rangle\langle\psi| && (35)\\ &= \frac{1}{d}\sum_{j,k}|j\rangle\langle k|\otimes|j\rangle\langle k|\,. && (36)\end{aligned}$$

To calculate $E_{\mathcal{N}}(\rho)$, first find the partial transpose on $\mathcal{H}^{(A)}$ (the choice is not significant, the partial transpose on $\mathcal{H}^{(B)}$ yields the same result as the subspaces are equally entangled with one another).

$$\rho^{\Gamma_A} = \frac{1}{d}\sum_{j,k}|k\rangle\langle j|\otimes|j\rangle\langle k| \qquad (37)$$

From this the trace norm $\|X\|_1 = \mathrm{Tr}\left(\sqrt{XX^\dagger}\right)$ can be determined.

$$\begin{aligned}\rho^{\Gamma_A}\left(\rho^{\Gamma_A}\right)^\dagger &= \frac{1}{d^2}\sum_{j,k}|k\rangle\langle k|\otimes|j\rangle\langle j| && (38)\\ &= \frac{1}{d^2}I_d\otimes I_d && (39)\\ &= \frac{1}{d^2}I_{d^2} && (40)\\ \implies \|\rho^{\Gamma_A}\|_1 &= \frac{1}{d}\,\mathrm{Tr}\left(I_{d^2}\right) && (41)\\ &= \frac{1}{d}\cdot d^2 && (42)\\ &= d && (43)\end{aligned}$$

Therefore, this gives

$$\begin{aligned}E_{\mathcal{N}} &= \log_2\left\|\rho^{\Gamma_A}\right\|_1 && (44)\\ &= \log_2 d. && (45)\end{aligned}$$

□

This combined with the strong additivity condition is what makes the choice of log negativity an appropriate one as it allows for a direct comparison of entanglement in a state before and after transfer. A single Bell pair resides in a Hilbert space of dimension 2 × 2 and has log negativity 1. Therefore, a collection of n Bell pairs exist in a Hilbert space of 2^{2n} and have a log negativity of n (additivity). The minimum dimension of the Hilbert space that this entanglement is to be transferred into is also 2^{2n} (if it is bigger then it can always be reduced to a space of that dimension). Therefore the qudit pair resides in a Hilbert space of $d \times d = 2^{2n} = 2^n \times 2^n$. Using the log negativity, a maximally entangled state in a Hilbert space of dimension $2^n \times 2^n$ has a log negativity of n (claim 2.3). Therefore the maximal log negativity in the qudit pair is equal to the sum of the log negativity of the Bell states that entanglement is transferred from. This means that if three Bell pairs have their entanglement transferred, the maximal possible log negativity of the resulting qudit pair is 3. If this upper bound is achieved then optimal transfer has occurred. If it is not achieved, then entanglement has been lost in the transfer. This might seem quite an obvious statement, but it is not true for all entanglement measures. For example, negativity, which is an alternative entanglement measure closely related to log negativity, cannot be used to make direct comparisons in the same way.

2.3 Quantum Random Walks

Quantum random walks, as touched on in section 1, are a model for universal quantum computation, meaning that any computation that is possible for a quantum computer can be simulated by a QW. Like much of quantum computing, QWs can be motivated from a classical predecessor, the *classical random walk*. Here the classical random walk is used specifically to motivate the discrete QW since the protocol presented in section 3 makes use of such a QW. Hence all future references to QWs are assumed to be of the discrete variant.

2.3.1 Motivation from the Classical Random Walk

In the classical random walk, a walker is constrained to moving up and down a discrete number line, starting their walk at the origin. To determine whether to take a step to the left (-1) or the right (+1), the walker flips an unbiased coin, moving to the right if the coin lands on heads and to the left if the coin lands on tails. This process of flipping a coin and taking a step can be repeated until a desired stopping point is reached, e.g. after a given number of coin flips. The walk can also continue on forever and in this limit, the walker will reach every point on the number line. In this report, however, only finite length walks are considered. Without much thought it can be concluded that after T coin flips the walker can be at most $\pm T$ steps away from the origin, corresponding to the case where every step taken is in the same direction. The probability distribution describing the probability of the walker being in any given position away from the origin is given by a binomial distribution, an example of which is shown in orange in figure 2 for 100 coin flips.

2.3.2 Quantum Walk on a Line

Having described the classical random walk, it can now be 'quantised' into the *quantum random walk*. The QW is considered as a composite system of the coin, belonging to the Hilbert space $\mathcal{H}_C$ and walker, $\mathcal{H}_W$,

$$\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_W. \tag{46}$$

Note that this use of subscript does not depart from the earlier usage to indicate the dimension of the Hilbert space, since the $\mathcal{H}_C$ is spanned by two states (the 'heads' state and the 'tails' state), and $\mathcal{H}_W$ is spanned by all the points on the number line to be walked upon, so can be thought of as an unspecified d dimensional space. Therefore the labels C and W, also serve to indicate the dimension of the labelled space.

To aid distinguishability between coin states and position states, they are labelled by

$$\langle \mathcal{H}_C \rangle = \{ |\uparrow\rangle, |\downarrow\rangle \} \tag{47}$$

$$\langle \mathcal{H}_W \rangle = \{ |k\rangle \,|\, -T \leq k \leq +T, k \in \mathbb{Z} \}, \tag{48}$$

where $\langle U \rangle$ denotes a set of states which span U, and T again is the number of 'coin flips'. Therefore, the states $|\uparrow\rangle, |\downarrow\rangle$ take the place of heads and tails on our quantum 'coin'. Note in particular that there is no constraint placed on the dimensionality of $\mathcal{H}_W$, and therefore the walker can be taken to be a qudit, whereas $\dim \mathcal{H}_C = 2$ and the coin is taken to be a qubit. Now that the Hilbert space of the system has been contructed, operators are required in order for the system to evolve. In the classical random walk, the first operation needed was a coin flip so in similar fashion the quantum walk too needs a coin flip operator, $C \in \mathcal{H}_C$ (not to be confused with the C-U controlled-unitary operator). There is a continuum of choices for C, and indeed QWs can be constructed on a whole host of different graphs - see for a greater discussion [17]. For walks on a line, constraining the coin to have real coefficients leaves the Hadamard coin as the only choice of coin available. This takes on the same form as the Hadamard transform defined in section 2.1.5 and is given here via an alternative expression,

$$H = \frac{1}{\sqrt{2}} \Big[|\uparrow\rangle\langle\uparrow| + |\uparrow\rangle\langle\downarrow| + |\downarrow\rangle\langle\uparrow| - |\downarrow\rangle\langle\downarrow| \Big] \tag{49}$$

$$= \frac{1}{\sqrt{2}} \Big[\Big(|\uparrow\rangle + |\downarrow\rangle \Big) \langle\uparrow| + \Big(|\uparrow\rangle - |\downarrow\rangle \Big) \langle\downarrow| \Big] \tag{50}$$

where $|0\rangle \rightarrow |\uparrow\rangle$ and $|1\rangle \rightarrow |\downarrow\rangle$. As before, if the coin state is $|\uparrow\rangle$ then it becomes an equal in phase superposition of $|\uparrow\rangle + |\downarrow\rangle$, if the coin state is in $|\downarrow\rangle$ then it becomes equal antiphase superposition of $|\uparrow\rangle - |\downarrow\rangle$ and this is made particularly clear by equation 50. Following this, a shift operator $S \in \mathcal{H}$ is needed to move the walker dependent on the state of the coin.

$$S = \sum_k |\uparrow\rangle\langle\uparrow| \otimes |k+1\rangle\langle k| + |\downarrow\rangle\langle\downarrow| \otimes |k-1\rangle\langle k| \,. \tag{51}$$

Again, representing S in this way makes manifest its effect on the walker. If the coin state is $|\uparrow\rangle$, the walker takes a step in the +1 direction, if it is $|\downarrow\rangle$ then a step in the -1 direction is taken. The probability distribution of such a walk is plotted in blue in figure 2, where the initial coin state is $|\downarrow\rangle$, and is compared to a classical random walk.

Whilst the above choice of S is the most common on the number line, there are alternatives such as

$$\tilde{S} = \sum_k |\uparrow\rangle\langle\uparrow| \otimes |k\rangle\langle k| + |\downarrow\rangle\langle\downarrow| \otimes |k+1\rangle\langle k| \,. \tag{52}$$

Th subtle difference between $\tilde{S}$ and S is that $\tilde{S}$ can only move in the +1 direction of the number line and has no 'left moving' part. This means that $\tilde{S}$ is restricted to the non-negative integers and can occupy all $|x\rangle$ for $0 \leq x \leq T$, where T is the number of time steps in our QW. This is compared with S which only occupies even or odd numbered positions dependent on T. In fact $\tilde{S}$ is an operator that was introduced earlier in section 2.1.4,

$$\tilde{S} = C\text{-}X. \tag{53}$$

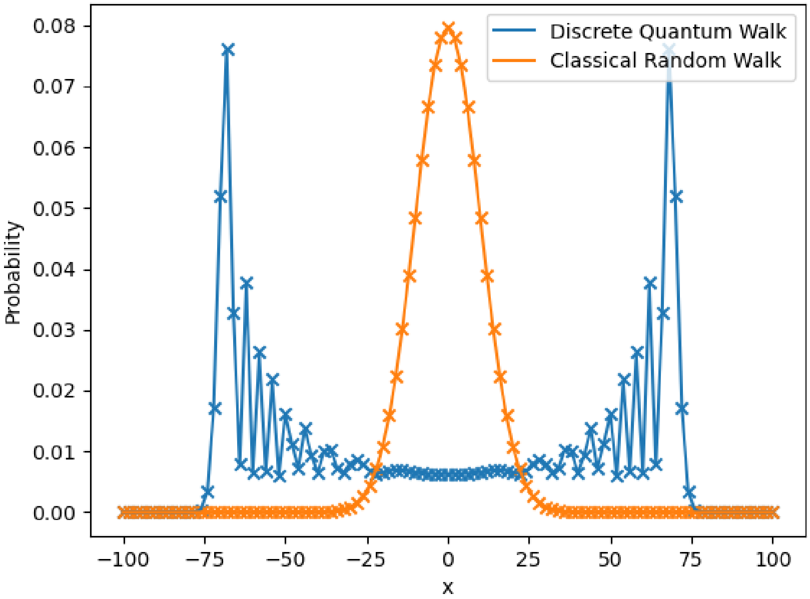


Figure 2: The probability distributions of a classical and Hadamard-coined quantum walk on a line, both originating at the origin. The quantum walk coin is initially in the $\frac{1}{\sqrt{2}}\left(\left|\uparrow\right\rangle + i\left|\downarrow\right\rangle\right)$ state. Odd points have been omitted as they all have zero probability.

2.4 Ancilla-Based Quantum Computing

The quantum walk model of quantum computation is but one of many universal quantum computing models. Another model is known as *ancilla-based quantum computing* (AQC), which, in particular, aims to resolve two conflicting demands when building quantum computers: qubits must be well isolated to prevent decoherence, yet it must also be possible for them to interact with one another in certain settings, such as implementing the C-X operator. However, a qubit cannot distinguish between unwanted and wanted interactions, resulting in the need for a balancing act that adequetely fulfils both of these aims.

AQC aims to resolve this conflict by using additional qubits, known as ancilla qubits, which mediate interactions between the main register qubits. Using this model, it is possible to implement two complementary registers of qubits. The main register can be designed to be strongly isolated from interactions, bar a limited number of interactions with the ancilla register, whose qubits can be easily manipulated but decohere much faster. Quantum computations can be performed by *delocalising* quantum information from the main register across both the register and the ancilla. After performing a computation on the ancilla qubit, the quantum information is then *relocalised* back into the main register. The ancilla can then be reset to the initial state and used again for other computations. A simple example of a circuit designed for ancilla-based quantum computing is shown in figure 3.

The circuit on the left hand side of figure 3 does not directly implement any unitary transformations on an arbitrary qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, but instead indirectly acts on $|\psi\rangle$

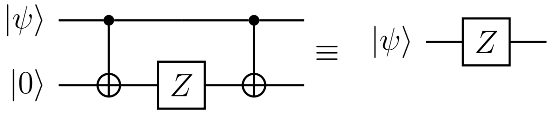


Figure 3: Two circuits that implement a Z gate acting on an arbitrary qubit $|\psi\rangle$. The circuit on the left does so indirectly via interactions with an ancilla qubit.

via the ancilla qubit, to which the quantum information of $|\psi\rangle$ is shared to via a C-X gate. Acting a Z gate on the ancilla and relocalising the quantum information with another C-X gate 'passes on' the effects of the Z gate. Note however that this would not directly work to implement an X gate on $|\psi\rangle$ if the Z gate were directly switched with an X gate. This is because only amplitudes can be delocalised and relocalised; computational basis states cannot be directly shifted by ancilla interactions. If however, $|\psi\rangle$ is mapped to the Hadamard basis, where the basis states differ by their amplitudes, before the Z circuit and mapped back afterwards, as in figure 4, then it is possible to effect an X operator in the AQC setting. These

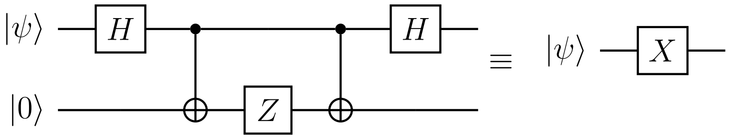


Figure 4: Implementing an X gate on $|\psi\rangle$ in the AQC setting by first mapping $|\psi\rangle$ to the Hadamard basis before and after the Z circuit given in figure 3.

two circuits highlight the somewhat counterintuitive nature of AQC. A controlled operation is utilised, where the qudit is the control and the qubit is the target, yet it is the qudit state that is changed and the qubit is left 'untouched'.

This model of quantum computing does not have to be reserved to qubit computing alone and can be extended to qudits simply by using the general forms of operators, as described in section 2.1.4. An further advantage of AQC is that it is particularly well placed to used registers of qudits with mismatched dimensions. Therefore, it is an extremely suitable model of quantum computation for entanglement transfer as entangled qubits can be used as the ancilla (to be regularly replaced once the entanglement is transferred), and qudits of arbitrary dimension can be the main register to ensure the entanglement transferred is long-lived.

3 Entanglement Transfer using Quantum Walks

Here the protocol presented by Giordani et al. [10] is summarised. The aim of the protocol is to generate higher dimensional entanglement via the use of quantum walks. The imagined setup of this protocol is that there are two labs, A and B, which have a shared source of entangled qubits but are otherwise spatially separated and cannot interact with one another.

In this section the following notation is employed:

- $|\psi\rangle_J$ is a state belonging to the subspace $\mathcal{H}_J = \mathcal{H}_J^{(A)} \otimes \mathcal{H}_J^{(B)}, J \in \{C, W\}$. It is a state that describes the combined state of the coins or walkers.
- $|\psi\rangle^{(K)}$ is a state belonging to the subspace $\mathcal{H}^{(K)} = \mathcal{H}_C^{(K)} \otimes \mathcal{H}_W^{(K)}, K \in \{A, B\}$. It is a state that describes the combined state of one of the quantum walks.
- $|\psi\rangle_J^{(K)}$ is a state belonging to the subspace $\mathcal{H}_J^{(K)}$.

In this mathematical framework the overall Hilbert space is comprised of two quantum walk subspaces,

$$\begin{aligned} \mathcal{H} &= \mathcal{H}^{(A)} \otimes \mathcal{H}^{(B)} && (54) \\ &= \mathcal{H}_C^{(A)} \otimes \mathcal{H}_W^{(A)} \otimes \mathcal{H}_C^{(B)} \otimes \mathcal{H}_W^{(B)}. && (55) \end{aligned}$$

Due to the difficulty, however, in writing entangled states when the entangled spaces are not adjacent, states will be written with the following ordering

$$\mathcal{H} = \mathcal{H}_W^{(A)} \otimes \mathcal{H}_W^{(B)} \otimes \mathcal{H}_C^{(A)} \otimes \mathcal{H}_C^{(B)}, \tag{56}$$

i.e. with the walker subspaces adjacent to each other and the coin subspaces adjacent to each other.

The basic premise of this protocol is this:

1. Entangle the two coin spaces of the walkers $\mathcal{H}_C^{(K)}$. (Figure 5.)
2. Proceed with the quantum walk for some determined number of steps.
3. Use a projective measurement $\mathcal{P}_\gamma = |\gamma\rangle\langle\gamma|, |\gamma\rangle \in \mathcal{H}_C^{(A)}$ to then transfer the entanglement so that it solely exists in the subspace $\mathcal{H}_W^{(A)} \otimes \mathcal{H}_C^{(B)} \otimes \mathcal{H}_W^{(B)}$.
4. In similar fashion, find a projection $\mathcal{P}_\delta = |\delta\rangle\langle\delta|, |\delta\rangle \in \mathcal{H}_C^{(B)}$ to transfer the entanglement to exist between the two walker subspaces, $\mathcal{H}_W^{(i)}$, only.
5. Accumulate entanglement in the walker subspaces by once more entangling the two coin spaces and repeating the protocol.

In this way, arbitrary amounts of higher dimensional entanglement can be generated.
The ordering of steps 3 and 4 is not overly important since the full operators describing the projective measurements are given by

$$\begin{aligned} &|\gamma\rangle\langle\gamma| \otimes I \otimes I \otimes I, && (57) \\ &I \otimes |\delta\rangle\langle\delta| \otimes I \otimes I && (58) \end{aligned}$$

which clearly commute. As is the case with many quantum walk based protocols, particular attention must be paid to the choice of coin used for the quantum walk, as it will have a large impact on the success of the protocol. The shift operator used for the quantum walks is $\tilde{S}$, as outlined in section 2.3.2. (Technically it is $\tilde{S} \otimes \tilde{S}$, one for each walk, but for the sake of brevity $\tilde{S}$ will be used.)

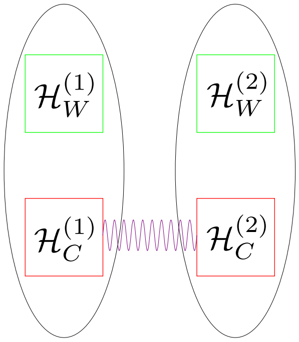


Figure 5: The initial prepared state has entanglement solely between the two coin subspaces. Figure is an edited version of FIG 3 from [10].

3.1 Transfer using identity coin operator

To illustrate the basic principles of the protocol, consider the case where

$$C = \underbrace{I_2 \otimes I_2}_{\text{`Coin' operators}} \otimes I_d \otimes I_d = I. \tag{59}$$

Note that this is not actually an instance of using quantum walk dynamics, since an identity coin equates to no coin at all.

1. A state $|\psi(0)\rangle$ is prepared with the walkers at the origin and coin states entangled,

$$|\psi(0)\rangle = |0\rangle_W^{(A)} |0\rangle_W^{(B)} \otimes \underbrace{\frac{1}{\sqrt{2}}\Big[|\uparrow\rangle_C^{(A)} |\uparrow\rangle_C^{(B)} + |\downarrow\rangle_C^{(A)} |\downarrow\rangle_C^{(B)} \Big]}_{\text{Bell State}}. \tag{60}$$

2. The ‘coin’, I, is applied and followed by the shift operator $\tilde{S}$ to advance the quantum walk. Explicitly (dropping the indices and combining kets together) the walk evolves to the state

$$|\psi(1)\rangle = \tilde{S}I\,|\psi\rangle = \frac{1}{\sqrt{2}}\Big[|0,0\rangle |\uparrow,\uparrow\rangle + |1,1\rangle |\downarrow,\downarrow\rangle \Big]. \tag{61}$$

3. Using the operator $\mathcal{P}_\gamma = |\gamma\rangle\langle\gamma|$ the part of $|\psi(1)\rangle$ residing in the $\mathcal{H}_C^{(A)}$ subspace is projected onto the state $|\gamma\rangle$.

   For this example, choose $|\gamma\rangle = \frac{1}{\sqrt{2}}\big[|\uparrow\rangle + |\downarrow\rangle \big]$ which then gives

$$\mathcal{P}_\gamma\,|\psi(1)\rangle = \frac{1}{2}\Big[|0,0\rangle |\gamma,\uparrow\rangle + |1,1\rangle |\gamma,\downarrow\rangle \Big]. \tag{62}$$

4. Similarly, project the other coin onto $|\delta\rangle$ which in this instance is taken to be the same state $|\delta\rangle = \frac{1}{\sqrt{2}}\big[|\uparrow\rangle + |\downarrow\rangle \big]$,

$$\mathcal{P}_\delta\mathcal{P}_\gamma\,|\psi(1)\rangle = \frac{1}{2\sqrt{2}}\Big[\frac{1}{\sqrt{2}}\Big(|0,0\rangle + |1,1\rangle \Big) |\gamma\rangle |\delta\rangle \Big]. \tag{63}$$

Renormalising gives the final state

$$\underbrace{\frac{1}{\sqrt{2}}\Big[\,|0,0\rangle+|1,1\rangle\,\Big]_W}_{\text{Bell State}}\otimes|\gamma\rangle_C^{(A)}\otimes|\delta\rangle_C^{(B)}\,, \tag{64}$$

which has a Bell state in the $\mathcal{H}_W$ subspace, and the coin states are separable. Therefore the entanglement that originally resided in the coin subspace has been transferred to the walker one.

3.2 Accumulation

The true motivation behind this protocol is the ability to accumulate the entanglement transferred from the lower dimensional coin subspace to the higher dimensional walker one. This is done by repeating the entire process with some small changes. Again I is used as the coin.

1. Starting with the final state obtained from the first iteration of the protocol (equation 64) the coin subspaces are re-entangled, obtaining a new initial state $|\psi(0)\rangle$,

$$\frac{1}{\sqrt{2}}\Big[\,|0,0\rangle+|1,1\rangle\,\Big]_W\otimes|\gamma\rangle_C^{(A)}\otimes|\delta\rangle_C^{(B)} \tag{65}$$

$$\xrightarrow{\text{Entangle coins}}\frac{1}{2}\Big[\,|0,0\rangle+|1,1\rangle\,\Big]_W\otimes\Big[\,|\uparrow,\uparrow\rangle+|\downarrow,\downarrow\rangle\,\Big]_C \tag{66}$$

$$:=|\psi(0)\rangle\,. \tag{67}$$

2. Now take two steps instead of one in the walk.

$$|\psi(2)\rangle=(\tilde{S}I)^2\,|\psi(0)\rangle \tag{68}$$

$$=\frac{1}{2}\Big[\big(\,|0,0\rangle+|1,1\rangle\,\big)\,|\uparrow,\uparrow\rangle+\big(\,|2,2\rangle+|3,3\rangle\,\big)\,|\downarrow,\downarrow\rangle\Big]. \tag{69}$$

3–4. Using the same projection operators in the two coin subspaces, $\mathcal{P}_\gamma\in\mathcal{H}_C^{(A)},\mathcal{P}_\delta\in\mathcal{H}_C^{(B)}$, and renormalising gives the final state

$$\frac{1}{2}\Big[\,|0,0\rangle+|1,1\rangle+|2,2\rangle+|3,3\rangle\,\Big]\otimes|\gamma\rangle\otimes|\delta\rangle\,. \tag{70}$$

Using claim 2.3, the log negativity of

$$\frac{1}{2}\Big[\,|0,0\rangle+|1,1\rangle+|2,2\rangle+|3,3\rangle\,\Big] \tag{71}$$

is 2, setting $d=4$ (since both of the qudits only have 4 states of non-zero amplitude they can be simulated by ququarts even if their true dimension is greater than 4). Therefore, 2 units of log negativity have been transferred from the Bell states (each of log negativity 1) to the qudit pair and optimal transfer has been achieved. The process can be repeated to accumulate arbitrarily large amounts of entanglement into our walker subspace. The number of steps needed in the quantum walk for each iteration is as follows.

Claim 3.1. *The n^{th} iteration (counting from 1) of the protocol requires 2^{n-1} steps in the quantum walk.*

Proof. Each step in a quantum walk with shift operator $\tilde{S}$ increases the number of basis states with non-zero amplitude by 1, provided that the amplitude $|\downarrow\rangle$ coin basis state is non-zero. Therefore the dimension of each walker can be taken to be $s+1$, where s is the total number of steps taken in the walk, since each walker space can be reduced to the subspace spanned by the $s+1$ non-zero amplitude basis states. Using claim 2.3, the upper bound on the log negativity of the combined walker states after the n^{th} iteration of the walk is given by $\log_2(s+1)$. This implies the following condition on the total number of steps

$$\log_2(s+1) = n \implies s = 2^n - 1. \tag{72}$$

This is an equality rather than an inequality $\log_2(s+1) \geq n$, since the state of the walkers is to be maximally entangled. This means that the combined walker Hilbert space $\mathcal{H}_W$ must be reducible to one of dimension $(s+1) \times (s+1)$. This is only possible when taking the minimum number of steps in each quantum walk (if more steps are taken then there are more than $s+1$ basis states of non-zero amplitude).

Let s_{n-1} be the total number of steps taken up to the $(n-1)^{\text{th}}$ iteration of the protocol. Assuming that s_{n-1} satisfies equation 72 implies $s_{n-1} = 2^{n-1} - 1$. Therefore, the number of steps for the n^{th} iteration is given by

$$\begin{aligned} s - s_{n-1} &= 2^n - 1 - (2^{n-1} - 1) && (73)\\ &= 2^n - 2^{n-1} && (74)\\ &= 2 \times 2^{n-1} - 2^{n-1} && (75)\\ &= 2^{n-1}. && (76) \end{aligned}$$

□

3.3 Retrieval

Although accumulating the entanglement in higher dimensions is of significant use, it is also possible to imagine that retrieving the entanglement back into the qubits would also be of use. For example, consider the case where the method of generating entangled qubits is not deterministic but an algorithm requiring large numbers of Bell states is to be executed. If the accumulation of entanglement can be reversed to get Bell pairs back, then the entangled Bell pairs can be stored in the qudits when they are successfully generated and then retrieved all at once to ensure the algorithm can be performed. In principle, it is possible to use a similar quantum walk setup where the entangled qudits are the coins and the qubits are the walkers to get entanglement back out. However due to the non-unitary nature of the protocol (the projective measurements are irreversible), this setup is not simple to design.

3.4 Results

Despite its simple construction, the state of a quantum walk becomes extremely complex after even just a few steps in most cases. This is doubly true for two quantum walks running

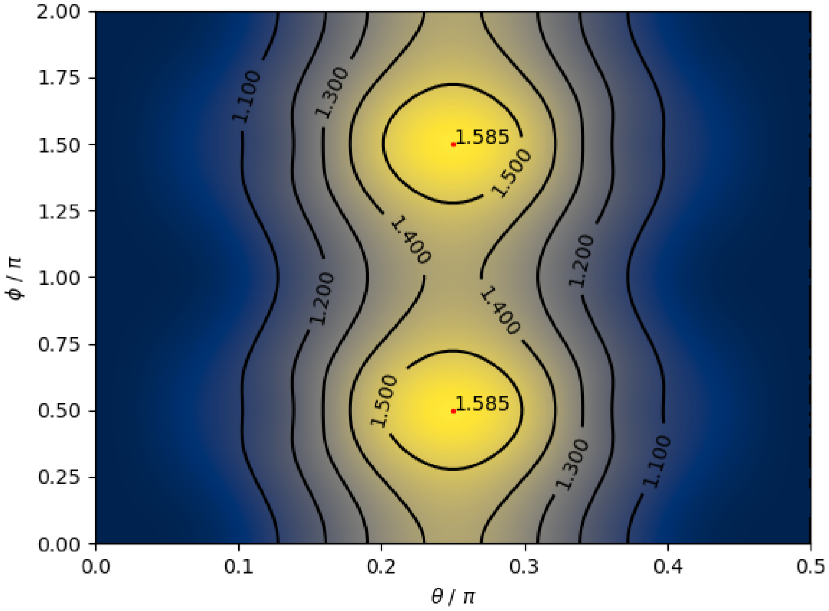


Figure 6: Transferring two ebits using the quantum walk protocol proposed by [10]. θ, ϕ are the parameters describing $|\gamma\rangle$ that defines the projection operator $\mathcal{P}_\gamma$.

in conjunction. Hence to analyse the protocol in settings that are not as trivial as the identity coin example, numerical simulations were run using Python and some associated packages [18, 19]. Figure 6 shows that the protocol, when using a Hadamard coin, can generate a state with log negativity up to approximately 1.585 when the protocol is done twice, assuming both quantum coins are projected onto the same state $|\gamma\rangle = \cos\theta\,|0\rangle + e^{i\phi}\sin\theta\,|1\rangle$. *****Simulations using different projective operators $\mathcal{P}_\gamma, \mathcal{P}_\delta$ show 1.585 is still the maximum possible.******** Hence a maximally entangled state cannot be generated with the Hadamard coin. *****Other coin choices gave similar results********* Clearly whilst this proposal can be used to generate states of higher dimensional entanglement, the constraints of using quantum walk dynamics to achieve this transfer are ones that significantly hamper the efficiency of transfer. Hence, it is worth looking at alternative models of quantum computing to adapt this protocol to in order to examine whether optimal transfer can be realised under a different set of constraints.

4 Entanglement Transfer using Ancilla-Based Quantum Computing

In this section an alternative entanglement transfer scheme is outlined. It is designed to operate in the same setting at the QW based protocol, where there are two labs, again called A and B, which share a common source of entangled qubits and each have their own qudits. Analogously to the QW protocol, the qudit can be thought of as a walker driven by the ancilla qubit 'coin'.

The protocol can be succintly summarised by the circuit schematic given in figure 7, and essentially is two copies of the same circuit.

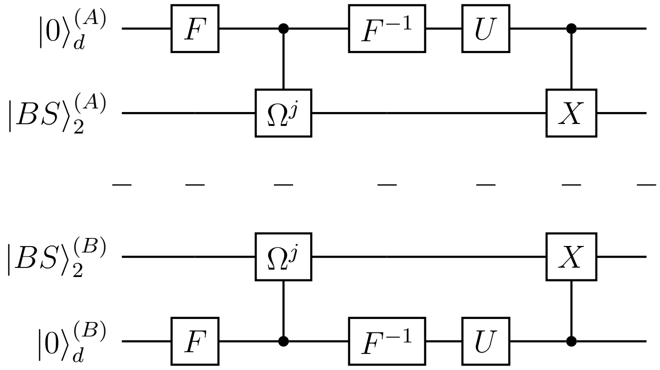


Figure 7: The AQC circuit for entanglement transfer. The schematic shows two separated circuits imagined to exist in spatially separated labs, A and B. The qudits and qubits belonging to each individual lab are labelled by their superscripts. The qubits form an entangled Bell state.

The majority of the gates shown in the schematic are discussed in section 2.1.5, with the exception of the gate Ω^j. The action of Ω in the computational basis is given by the matrix

$$\Omega = \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix} \tag{77}$$

$$\implies \Omega^j = \begin{pmatrix} 1 & 0 \\ 0 & \omega^j \end{pmatrix}. \tag{78}$$

Essentially Ω is the similar to the Z operator in that it applies relative phase difference, but instead of a phase of -1, a phase corresponding to the d^{th} root of unity is introduced on the $|1\rangle_2$ states, with d corresponding to the dimension of the control qudits. (Indeed, when $d = 2$, $\Omega = Z$.) Therefore the C-Ω^j operator is one that acts as

$$C\text{-}\Omega^j(|x\rangle_d |y\rangle_2) = |x\rangle_d \otimes \Omega^{xj} |y\rangle_2 \tag{79}$$

$$= |x\rangle_d \otimes \omega^{xyj} |y\rangle_2 \tag{80}$$

where $|x\rangle_d$ is the control qudit and $|y\rangle_2$ the target qubit.

Claim 4.1. *In the Fourier basis, the C-Ω^j operator acts on the product state $|+_k\rangle_d \otimes |1\rangle_2$ to give $|+_{k+j}\rangle_d \otimes |1\rangle_2$*

Proof. The Fourier basis state $|+_k\rangle_d$ is given by

$$|+_k\rangle_d = \frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} \omega^{km} |m\rangle . \tag{81}$$

Therefore

$$C\text{-}\Omega^j \left(|{+_k}\rangle_d |1\rangle_2\right) = \frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} C\text{-}\Omega^j \left(\omega^{km} |m\rangle_d \otimes |1\rangle_2\right) \tag{82}$$

$$= \frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} \omega^{mj} \left(\omega^{km} |m\rangle_d \otimes |1\rangle_2\right) \tag{83}$$

$$= \frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} \omega^{(k+j)m} |m\rangle_d \otimes |1\rangle_2 \tag{84}$$

$$= \left(\frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} \omega^{(k+j)m} |m\rangle_d\right) \otimes |1\rangle_2 \tag{85}$$

$$= |{+_{k+j}}\rangle_d \otimes |1\rangle_2 \,. \tag{86}$$

□

This exhibits again the counter-intuitive nature of AQC discussed in section 2.4. Despite the qudit being the control and the qubit being the target, the only change effected is on the qudit, shifting the Fourier basis states. Claim 4.1 also demonstrates the role of the index j, which essentially dictates by how many states each Fourier basis state is shifted.

The following sections demonstrate explictly how the circuit operates.

4.1 Transfer

Assume that there are two qudits in the state $|0\rangle_d$ in labs A and B, and each lab shares a qubit each from an entangled Bell pair. The composite state is therefore given by

$$|00\rangle_d^{(A,B)} \otimes \frac{1}{\sqrt{2}} \left(|00\rangle_2^{(A,B)} + |11\rangle_2^{(A,B)}\right) . \tag{87}$$

Note that in the above expression, the qudits and qubits in both labs, A and B, are in fact in the same states. This is true for the entire running of the circuit and as such there is no need to explictly differentiate the qudits or ancilla qubits in either lab so the superscript labelling is dropped for the rest of this section.

The qudits are first Fourier transformed into the Fourier basis to give

$$|{+_0}{+_0}\rangle_d \otimes \frac{1}{\sqrt{2}} \left(|00\rangle_2 + |11\rangle_2\right) \tag{88}$$

$$= \frac{1}{\sqrt{2}}\Big(|{+_0}{+_0}\rangle_d \otimes |00\rangle_2 + |{+_0}{+_0}\rangle_d \otimes |11\rangle_2 \Big). \tag{89}$$

The change of basis is needed because to have entanglement in the qudits, they must be in superposition of at least two states of non-zero amplitude. Since only amplitudes can be delocalised across the qudits and ancilla qubits, in order to give other qudit states a non-zero amplitude, it is necessary to be in a basis where the basis states are related to one another by differences in their amplitudes. The Fourier basis is one such basis where the basis states are in equal superposition of all the computational basis states but differ by the relative phases in the superposition. Hence the need to be in the Fourier basis when using the delocalised amplitudes to effect changes in the qudit basis states. The index j on the Ω has an analogy to the QW protocol where the iteration number dictates how many steps need to be taken in the QW. Using the same line of reasoning as outlined in Claim 3.1, on the n^{th} iteration of the protocol, $j = 2^{n-1}$ so in this instance (where $n = 1$) $j = 1$. Using claim 4.1, acting C-Ω in both A and B gives the state

$$\frac{1}{\sqrt{2}} \left(|{+_0}{+_0}\rangle_d \otimes |00\rangle_2 + |{+_1}{+_1}\rangle_d \otimes |11\rangle_2 \right). \tag{90}$$

The qudits are then transformed back into the computational basis,

$$\frac{1}{\sqrt{2}} \left(|00\rangle_d \otimes |00\rangle_2 + |11\rangle_d \otimes |11\rangle_2 \right). \tag{91}$$

The operator U is a correctional gate that essentially pairs all the even numbered computational qudit basis states with $|0\rangle_2$ and the odd numbered computational basis states with $|1\rangle_2$. (A more complete description is given in section 4.2.) In this instance, U can be ignored since no corrections are needed, so C-X is directly applied, giving

$$\frac{1}{\sqrt{2}} \left(|00\rangle_d \otimes |00\rangle_2 + |11\rangle_d \otimes |00\rangle_2 \right) \tag{92}$$

$$= \underbrace{\frac{1}{\sqrt{2}} \left(|00\rangle_d + |11\rangle_d \right)}_{\text{Bell state}} \otimes |00\rangle_2 . \tag{93}$$

As shown in equation 93, the qudits now effectively form a Bell pair and the qubits are no longer entangled - the entanglement has been transferred to the qudits.

4.2 Accumulation

Assume that one Bell pair has been transferred to our qudits and a second Bell pair is to be transferred. Taking the final state given by equation 93 and replacing the ancilla qubits with another entangled Bell pair gives

$$\frac{1}{2} \left(|00\rangle_d + |11\rangle_d \right) \otimes \left(|00\rangle_2 + |11\rangle_2 \right). \tag{94}$$

Again, the qudits are transformed to the Fourier basis before the C-Ω^j gate. As noted in the discussion on C-Ω^j above, for this iteration (where $n = 2$) $j = 2$, giving the state

$$\frac{1}{2} \Big[\left(|{+_0}{+_0}\rangle_d + |{+_1}{+_1}\rangle_d \right) \otimes |00\rangle_2 + \left(|{+_2}{+_2}\rangle_d + |{+_3}{+_3}\rangle_d \right) \otimes |11\rangle_2 \Big]. \tag{95}$$

After transforming back to the computational basis, the total state becomes

$$\frac{1}{2} \Big[\left(|00\rangle_d + |11\rangle_d \right) \otimes |00\rangle_2 + \left(|22\rangle_d + |33\rangle_d \right) \otimes |11\rangle_2 \Big]. \tag{96}$$

Unlike the previous example for $n = 1$, here the correctional operator U is needed so that the C-X will ensure all the qubit states are $|0\rangle_2$, allowing our qubit state to be completely separable from its associated qudit. In order for the C-X to do this, it must act as the identity on $|0\rangle_2$, implement an X operation on $|1\rangle_2$. Since X is self inverse, the control must be in an even labelled basis state (turning X into the identity) when the target is $|0\rangle_2$ and in an odd labelled basis state (leaving X unchanged) when the target is $|1\rangle_2$. Hence in this case, U must implement

$$|1\rangle \mapsto |2\rangle \tag{97}$$

$$|2\rangle \mapsto |1\rangle\,, \tag{98}$$

which then gives the state

$$\frac{1}{2}\Big[(|00\rangle_d + |22\rangle_d) \otimes |00\rangle_2 + (|11\rangle_d + |33\rangle_d) \otimes |11\rangle_2\Big]. \tag{99}$$

As desired, U has paired even numbered qudit basis states with $|00\rangle_2$ and odd numbered qudit basis states with $|11\rangle_2$. Therefore after the C-X operation the state becomes

$$\frac{1}{2}(|00\rangle_d + |22\rangle_d + |11\rangle_d + |33\rangle_d) \otimes |00\rangle_2\,. \tag{100}$$

The two qudits are now in an entangled state with a log negativity of 2 (claim 2.3). This protocol can be repeated again and again with the sole change needed being the index j of C-Ω^j and the operator U. For $n > 2$ there is no unique choice of U, as long as the end result is that even numbered states are paired with $|0\rangle_2$ and odd numbered states are paired with $|1\rangle_2$. This can be re-expressed as requiring odd numbered states less than $\frac{d}{2}$ and even numbered states greater than or equal to $\frac{d}{2}$ to be swapped in some way.

4.3 Retrieval

Given that this is a circuit that solely utilises unitary transformations, retrieval of entangled Bell pairs is trivially done by running the circuit backwards. Furthermore, there is no requirement to use the same ancilla qubits to retrieve the entanglement. By the end of the circuit the ancilla qubits are in the $|0\rangle$ state, therefore any pair of ancilla qubits in the $|0\rangle$ state may be used to retrieve the entanglement.

5 Further Uses of the AQC Circuit

Although the goal of this research was to design a protocol in a similar setting to the QW based protocol that could optimally generate maximally entangled states, further probing of the AQC circuit shows that it has uses beyond storage of entanglement. In this section, a scenario is considered where the quantum information contained in a collection of qubits is to be stored. By taking one half of the circuit given for the two lab setting and adding a gate U^{-1} that undoes the basis state map seen before, as shown in figure 8, arbitrary qubit states can be stored in the qudit, provided that $d \geq 2^n$, where n is the number of qubit states to

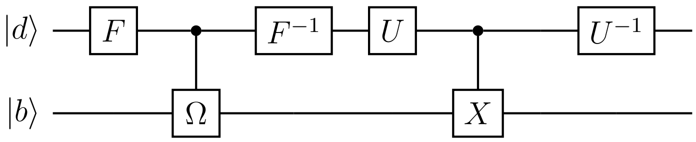


Figure 8: The AQC circuit with one qudit and qubit which can store arbitrary qubit states in the qudit.

be stored. Furthermore, in the case that two or more qubit states are stored via this circuit then it is in fact possible to retrieve the qubit states in a different order to that in which they were stored, as long as the original order they were stored in is known. Therefore, this circuit can be utilised in turning qudits into quantum random access memory.

5.1 Quantum Random Access Memory

The procedure for utilising the circuit as a quantum random access memory is as follows. Assume that there is a qudit in state $|0\rangle_d$ and two qubits, qubit 1 and qubit 2, in states $a|0\rangle_2 + b|1\rangle_2$ and $c|0\rangle_2 + d|1\rangle_2$ respectively. First run the circuit with the qudit and qubit 1,

$$|0\rangle_d \otimes (a|0\rangle + b|1\rangle) \longrightarrow (a|0\rangle_d + b|1\rangle_d) \otimes |0\rangle_2 . \tag{101}$$

Then replace qubit 1 with qubit 2 and run the circuit again,

$$(a|0\rangle_d + b|1\rangle_d) \otimes (c|0\rangle_2 + d|1\rangle_2) \longrightarrow (ac|0\rangle_d + bc|1\rangle_d + ad|2\rangle_d + bd|3\rangle_d) \otimes |0\rangle_2 . \tag{102}$$

In order to retrieve qubit 2 back, this can be done simply by running the circuit backwards. However, if the state of qubit 1 is to be retrieved then a specific unitary operator is first required. The form of the unitary transform can be found as follows. Rewrite each of the qudit basis state numbers in binary, i.e.

$$|0\rangle = |00\rangle \tag{103}$$
$$|1\rangle = |01\rangle \tag{104}$$
$$|2\rangle = |10\rangle \tag{105}$$
$$|3\rangle = |11\rangle . \tag{106}$$

Rewriting the final state of equation 102 in this way gives

$$ac|0\rangle + bc|1\rangle + ad|2\rangle + bd|3\rangle = ac|00\rangle + bc|01\rangle + ad|10\rangle + bd|11\rangle , \tag{107}$$

which can be rewritten as the product state

$$\underbrace{(c|0\rangle + d|1\rangle)}_{\text{Qubit 2}} \otimes \underbrace{(a|0\rangle + b|1\rangle)}_{\text{Qubit 1}} . \tag{108}$$

Quite remarkably the qudit has an intuitive alternative expression as the product state of qubit 1 and qubit 2. If the circuit is run backwards, qubit 2 will be retrieved, since it was the last qubit stored. Therefore, to retrieve qubit 1 instead, the qudit state should be expressable as

$$\underbrace{(a\left|0\right\rangle + b\left|1\right\rangle)}_{\text{Qubit 1}} \otimes \underbrace{(c\left|0\right\rangle + d\left|1\right\rangle)}_{\text{Qubit 2}}. \tag{109}$$

This can be thought of as switching the positions of our two qubits, which leads us to the unitary transformation needed. A map, M, is required which will switch the positions of our two qubits. M is found by mapping each of the binary qudit state representations $\{\left|00\right\rangle, \left|01\right\rangle, \left|10\right\rangle, \left|11\right\rangle\}$ to the basis state with the binary digits switched.

$$\left|00\right\rangle \mapsto \left|00\right\rangle \tag{110}$$

$$\left|01\right\rangle \mapsto \left|10\right\rangle \tag{111}$$

$$\left|10\right\rangle \mapsto \left|01\right\rangle \tag{112}$$

$$\left|11\right\rangle \mapsto \left|00\right\rangle. \tag{113}$$

To verify that this does give the desired result, take the RHS of equation 107 and act M on it to obtain

$$M\left(ac\left|00\right\rangle + ad\left|01\right\rangle + bc\left|10\right\rangle + bd\left|11\right\rangle\right) = ac\left|00\right\rangle + ad\left|10\right\rangle + bc\left|01\right\rangle + bd\left|11\right\rangle \tag{114}$$

$$= \underbrace{(a\left|0\right\rangle + b\left|1\right\rangle)}_{\text{Qubit 1}} \otimes \underbrace{(c\left|0\right\rangle + d\left|1\right\rangle)}_{\text{Qubit 2}}, \tag{115}$$

as required. M can be expressed in terms of the original qudit state labelling by converting the binary back

$$\left|0\right\rangle \mapsto \left|0\right\rangle \tag{116}$$

$$\left|1\right\rangle \mapsto \left|2\right\rangle \tag{117}$$

$$\left|2\right\rangle \mapsto \left|1\right\rangle \tag{118}$$

$$\left|3\right\rangle \mapsto \left|3\right\rangle. \tag{119}$$

The matrix representation of M is given by

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{120}$$

Generalising this to larger numbers of qubits stored is done via the same thought process. Imagine that there is now third qubit to be stored in the qudit, qubit 3, in the state

$$e\left|0\right\rangle + f\left|1\right\rangle. \tag{121}$$

When qubits 1, 2 and 3 are stored in that same order, it results in the state

$$ace\left|0\right\rangle + ade\left|1\right\rangle + bce\left|2\right\rangle + bde\left|3\right\rangle + ace\left|4\right\rangle + adf\left|5\right\rangle + bcf\left|6\right\rangle + bdf\left|7\right\rangle, \tag{122}$$

which again, when converted to binary numbers, can be expressed as the product state

$$\underbrace{(e\left|0\right\rangle + f\left|1\right\rangle)}_{\text{Qubit 3}} \otimes (c\left|0\right\rangle + d\left|1\right\rangle) \otimes \underbrace{(a\left|0\right\rangle + b\left|1\right\rangle)}_{\text{Qubit 1}}. \tag{123}$$

If the state of qubit 1 is needed, then M must switch the first and last qubits, and therefore must implement the map

$$\left|1\right\rangle = \left|001\right\rangle \mapsto \left|100\right\rangle = \left|4\right\rangle \tag{124}$$
$$\left|3\right\rangle = \left|011\right\rangle \mapsto \left|110\right\rangle = \left|6\right\rangle \tag{125}$$
$$\left|4\right\rangle = \left|100\right\rangle \mapsto \left|001\right\rangle = \left|1\right\rangle \tag{126}$$
$$\left|6\right\rangle = \left|110\right\rangle \mapsto \left|011\right\rangle = \left|3\right\rangle \tag{127}$$

with all other basis states mapped to themselves as they are identical when switching the first and last digits of their binary representations. Having operated M on the qudit state, the circuit can be run backwards in order to retrieve qubit 1. As with retrieving Bell pairs in section 4, any qubit in the state $\left|0\right\rangle$ can be used to retrieve qubit 1. A proof that this circuit can store any number of qubit states is given in appendix A.

In a practical setting it may be more efficient to build a gate, P, that permutes the order of the qubits instead of swapping two digits. One example of P would be the map that permutes each qubit one place the right,

$$\left|1\right\rangle = \left|001\right\rangle \mapsto \left|100\right\rangle = \left|4\right\rangle \tag{128}$$
$$\left|2\right\rangle = \left|010\right\rangle \mapsto \left|001\right\rangle = \left|1\right\rangle \tag{129}$$
$$\left|3\right\rangle = \left|011\right\rangle \mapsto \left|101\right\rangle = \left|5\right\rangle \tag{130}$$
$$\left|4\right\rangle = \left|100\right\rangle \mapsto \left|010\right\rangle = \left|2\right\rangle \tag{131}$$
$$\left|5\right\rangle = \left|101\right\rangle \mapsto \left|110\right\rangle = \left|6\right\rangle \tag{132}$$
$$\left|6\right\rangle = \left|110\right\rangle \mapsto \left|011\right\rangle = \left|3\right\rangle. \tag{133}$$

Again $\left|0\right\rangle, \left|7\right\rangle$ are mapped to themselves as the permutation does not affect them. In this way, any of the qubit states can be retrieved by continually applying P until the qubit state to be retrieved is in the right 'place'. Utilising a single gate would also be more appropriate for the ancilla-based quantum computing setting where a minimal gate set is desired for the control qudits. However, this would require multiple applications of the same gate which, without adequate error correction, would lead to greater decoherence of the quantum information than if just a single swapping gate M were applied.

6 Discussion

In the example given in section 3, it is shown that the protocol proposed in the QW setting can optimally transfer all the entanglement to the qudit pair. However, this is with the significant caveat that the example does not use quantum walk dynamics in order to transfer the entanglement, as using the identity as a coin is akin to having no coin at all. In analysing the protocol with the Hadamard coin, it was only possible to transfer one Bell state's worth of entanglement optimally and numerical simulations of the protocol for storing two Bell states, found that the resulting state had around 1.585 units of log negativity (figure 6). Furthermore, the projective measurements employed as part of the protocol mean that it is not a unitary protocol, therefore not deterministic and non-trivial to reverse in order to retrieve the entanglement out from the entangled qudits. The experimental implementation suggested in the paper also required the use of post selection, where undesirable states were discarded and the protocol run again. All this in combination results in a protocol which is rather inefficient in achieving its aims, and serves more as a proof of concept in that it is possible to use quantum walk dynamics to transfer entanglement, but falls short in being a suitable implementation for the task.

Adapting the identity coin example to the ancilla-based quantum computing circuit given in section 4 resulted in a circuit that solved these inefficiencies. The entanglement can always be transferred optimally and exclusive utilisation of unitary operators means that the circuit can just be reversed with any ancilla qubit in the $|0\rangle$ state.

6.1 Further Steps

As discussed in Section 5, the circuit given in figure 7 has potential uses beyond the original scope in which is has been designed. It would be of interest to see if it can also be applied to transferring multipartite entanglement to higher dimensions, for example the entanglement in transferring GHZ states or W states. Further analysis to see if Bell states could be used to generate multipartite entangled states would also be interesting to carry out. The analysis of this work could be brought to greater completion if the circuit was generalised to transferring entanglement between qudit pairs of arbitrary dimension instead of just entangled Bell pairs, although this would likely be more for academic rather than practical purposes, since the practical goal of the circuit is to take advantage of Bell pair generating schemes to generate higher dimensional entanglement.

6.2 Conclusions

Overall, the QW protocol aims to solve an interesting problem with a suitable set of contraints that might be realistic ones to consider in future as quantum computing technologies develop. In principle it does away with the need to repeatedly design different entanglement schemes for qudits of differing dimension, since the same scheme can used to accumulate entanglement in any dimension, just by using entangled Bell states to transfer the entanglement from. However, it was shown that in practice this protocol scaled inefficiently as more entanglement was transferred when actually using quantum walk dynamics, and optimal transfer is not possible.. It also suffers from difficulties in transferring entanglement back to the qubits due to its non-unitary nature. Post-selection is also needed further decreasing the efficiency of the protocol.

Instead, an alternative scheme was proposed to operate in the same physical setting, but based on an AQC model which came with a slightly different set of constraints. It was shown that this alternative scheme is able to achieve optimal transfer of entanglement, no matter the number of Bell states to be stored. Retrieval of entangled Bell pairs is also simple to do, since the entirety of the protocol was unitary, and retrieval amounted to reversing the circuit given in figure 7. Furthermore, it was also shown that the circuit had further uses outside of entanglement transfer and could be utilised to turn qudits into quantum random access memory. This increased versatility gives greater practical benefits to the AQC circuit as only one circuit is needed to achieve multiple aims. Its relative simplicity also makes it an effective circuit to implement on a practical quantum computer, with only a select few gates needed to implement it.

Acknowledgements

I would like to thank Dr Viv Kendon for her insight, advice and guidance during the course of this research, especially during the challenging conditions we found ourselves having to work in.

References

[1] P. W. Shor. ‘Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer’. In: *SIAM Journal on Computing* 26.5 (1997). URL: http://dx.doi.org/10.1137/S0097539795293172.

[2] C. H. Bennett and S. J. Wiesner. ‘Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states’. In: *Phys. Rev. Lett.* 69 (1992). URL: https://link.aps.org/doi/10.1103/PhysRevLett.69.2881.

[3] C. H. Bennett and G. Brassard. ‘Quantum cryptography: Public key distribution and coin tossing’. In: *Theoretical Computer Science - TCS* 560 (1984). DOI: 10.1016/j.tcs.2011.08.039.

[4] C. H. Bennett et al. ‘Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels’. In: *Phys. Rev. Lett.* 70 (1993). URL: https://link.aps.org/doi/10.1103/PhysRevLett.70.1895.

[5] X. S. Liu et al. ‘General scheme for superdense coding between multiparties’. In: *Physical Review A* 65.2 (2002). URL: http://dx.doi.org/10.1103/PhysRevA.65.022304.

[6] N. Shenvi, J. Kempe, and K. B. Whaley. ‘Quantum random-walk search algorithm’. In: *Phys. Rev. A* 67 (2003). URL: https://link.aps.org/doi/10.1103/PhysRevA.67.052307.

[7] A. M. Childs. ‘Universal Computation by Quantum Walk’. In: *Physical Review Letters* 102.18 (2009). URL: http://dx.doi.org/10.1103/PhysRevLett.102.180501.

[8] V. Kendon. ‘Decoherence in quantum walks – a review’. In: *Mathematical Structures in Computer Science* 17.06 (2007). URL: http://dx.doi.org/10.1017/S0960129507006354.

[9] A. M. Childs et al. ‘Exponential algorithmic speedup by a quantum walk’. In: *Proceedings of the thirty-fifth ACM symposium on Theory of computing - STOC ’03* (2003). URL: http://dx.doi.org/10.1145/780542.780552.

[10] T. Giordani et al. ‘Entanglement transfer, accumulation and retrieval via quantum-walk-based qubit-qudit dynamics’. In: *arXiv e-prints* (2020). arXiv: 2010.07127 [quant-ph].

[11] I. Bengtsson and K. Zyczkowski. *A brief introduction to multipartite entanglement*. 2016. arXiv: 1612.07747 [quant-ph].

[12] D. E. Browne and M. B. Plenio. 'Robust generation of entanglement between two cavities mediated by short interactions with an atom'. In: *Physical Review A* 67.1 (2003). URL: http://dx.doi.org/10.1103/PhysRevA.67.012325.

[13] L. F. Wei et al. 'Macroscopic Einstein-Podolsky-Rosen pairs in superconducting circuits'. In: *Physical Review A* 73.5 (2006). URL: http://dx.doi.org/10.1103/PhysRevA.73.052307.

[14] A. Messina. 'A single atom-based generation of Bell states of two cavities'. In: *The European Physical Journal D - Atomic, Molecular and Optical Physics* 18.3 (2002). URL: http://dx.doi.org/10.1140/epjd/e20020044.

[15] M. B. Plenio and S. Virmani. 'An Introduction to entanglement measures'. In: *Quant. Inf. Comput.* 7 (2007). arXiv: quant-ph/0504163.

[16] G. Vidal and R. F. Werner. 'Computable measure of entanglement'. In: *Physical Review A* 65.3 (2002). URL: http://dx.doi.org/10.1103/PhysRevA.65.032314.

[17] B. Tregenna et al. 'Controlling discrete quantum walks: coins and initial states'. In: *New Journal of Physics* 5 (2003). URL: http://dx.doi.org/10.1088/1367-2630/5/1/383.

[18] C. R. Harris et al. 'Array programming with NumPy'. In: *Nature* 585.7825 (2020). URL: https://doi.org/10.1038/s41586-020-2649-2.

[19] P. Virtanen et al. 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python'. In: *Nature Methods* 17 (2020). DOI: 10.1038/s41592-019-0686-2.

Appendices

A Proof that the circuit can store any number of qubits

Here it is proven by induction that the circuit presented in section 5 can store $n \in \mathbb{N}$ qubit states into the qudit, as long as the dimension of the Hilbert space $d \geq 2^n$.

Proof.
$n = 1$
It has already been shown in equation 101 that a single arbitary qubit state can be stored giving

$$(a_1 |0\rangle_d + b_1 |1\rangle_d) \otimes |0\rangle_2 \,, \tag{A.1}$$

where the coefficients have been relabelled $a \to a_1, b \to b_1$.

$n = k$
Assume that n qubit states have been stored in the qudit, giving the qudit state

$$\left(\sum_{j=0}^{2^k-1} c_j |j\rangle_d \right) \otimes |0\rangle_2 \,. \tag{A.2}$$

Furthermore, assume that this can be re-expressed as a tensor product of n qubits by converting the computational basis state labels to their binary equivalents and 'expanding' to give

$$(a_k |0\rangle + b_k |1\rangle) \otimes (a_{k-1} |0\rangle + b_{k-1} |1\rangle) \otimes \cdots \otimes (a_1 |0\rangle + b_1 |1\rangle) \otimes |0\rangle_2 \tag{A.3}$$

$$= \bigotimes_{i=k}^{1} (a_i |0\rangle + b_i |1\rangle) \otimes |0\rangle_2 \,, \tag{A.4}$$

where the coefficients a_i and b_i match the amplitudes of the i^{th} qubit state that was stored. Note that the assumption that the state can be expanded in such a way implies equation A.2 where

$$c_j = \prod_{i=1}^{k} (a_i \delta^0_{j_{2_i}} + b_i \delta^1_{j_{2_i}}), \tag{A.5}$$

where j_{2_i} is the i^{th} digit of the binary expression of j, and δ^m_n is the Kronecker delta.

$n = k + 1$
The entire state prior to running the circuit with an additional qubit to be stored is given by

$$\left(\bigotimes_{i=k}^{1} (a_i |0\rangle + b_i |1\rangle) \right) \otimes (a_{k+1} |0\rangle_2 + b_{k+1} |1\rangle_2) \tag{A.6}$$

$$= \left(\bigotimes_{i=k}^{1} (a_i |0\rangle + b_i |1\rangle) \right) \otimes a_{k+1} |0\rangle_2 + \left(\bigotimes_{i=k}^{1} (a_i |0\rangle + b_i |1\rangle) \right) \otimes b_{k+1} |1\rangle_2 \,. \tag{A.7}$$

This state is now put through the circuit. Note that the summand which is tensored with $|0\rangle_2$ is not affected by the C-Ω^j since the target is $|0\rangle_2$. Therefore only the summand tensored with $|1\rangle_2$ is advanced, and by claim 4.1, the basis states are shifted up by j steps. From claim 3.1, for the C-Ω^j gate, $j = 2^k$. Consider what this means in terms of computational basis states. When written terms of computational basis states, equation A.7 is given by the sum

$$\left(\sum_{i=0}^{2^k-1} c_i \left|i\right\rangle_d\right) \otimes a_{k+1} \left|0\right\rangle_2 + \left(\sum_{i=0}^{2^k-1} c_i \left|i\right\rangle_d\right) \otimes b_{k+1} \left|1\right\rangle_2 . \tag{A.8}$$

Shifting all of the computational basis states in the second summand by $j = 2^k$ yields

$$\left(\sum_{i=0}^{2^k-1} c_i \left|i\right\rangle_d\right) \otimes a_{k+1} \left|0\right\rangle_2 + \left(\sum_{i=0}^{2^k-1} c_i \left|i+2^k\right\rangle_d\right) \otimes b_{k+1} \left|1\right\rangle_2 . \tag{A.9}$$

In binary, adding 2^k to a number that is less than 2^k can be thought of as placing an additional 1 at the start of the string of binary digits. For example, where $(x)_2$ denotes that the number x is written in binary,

$$7 + 8 = (2^3 - 1) + (2^3) = (111)_2 + (1000)_2 = (1111)_2. \tag{A.10}$$

In this way, the action of the C-Ω^j gate on the tensor representation of the qudit given in equation A.7 can be thought of as adding a $|1\rangle$ in front of all of the binary representations of the computational basis states tensored with $|1\rangle_2$. Similarly, a $|0\rangle$ can be added in front of the binary representations of the computational basis states tensored with $|0\rangle_2$, since it still represents the same number $(01 = 1)$.

$$|0\rangle \otimes \left(\bigotimes_{i=k}^{1} (a_i \left|0\right\rangle + b_i \left|1\right\rangle)\right) \otimes a_{k+1} \left|0\right\rangle_2 + |1\rangle \otimes \left(\bigotimes_{i=k}^{1} (a_i \left|0\right\rangle + b_i \left|1\right\rangle)\right) \otimes b_{k+1} \left|1\right\rangle_2 \tag{A.11}$$

$$= a_{k+1} \left|0\right\rangle \otimes \left(\bigotimes_{i=k}^{1} (a_i \left|0\right\rangle + b_i \left|1\right\rangle)\right) \otimes \left|0\right\rangle_2 + b_{k+1} \left|1\right\rangle \otimes \left(\bigotimes_{i=k}^{1} (a_i \left|0\right\rangle + b_i \left|1\right\rangle)\right) \otimes \left|1\right\rangle_2 . \tag{A.12}$$

It is only possible to add these additional kets if the $d \geq 2^{k+1}$, else the Hilbert space is not large enough to accomodate the additional states being represented, hence the need for that contraint. The next step of the circuit is to use some map U to pair the even numbered computational basis states with $|0\rangle_2$ and the odd numbered computational basis states with $|1\rangle_2$ to convert the $|1\rangle_2$ to a $|0\rangle_2$ via a C-X operation, after which U is uncomputed. On the qudit state, since no amplitude changes are involved this is akin to doing a $UU^{-1} = I$ and so leaves the qudit state untouched. Therefore the final state is given by

$$a_{k+1} \left|0\right\rangle \otimes \left(\bigotimes_{i=k}^{1}(a_i \left|0\right\rangle + b_i \left|1\right\rangle)\right) \otimes \left|0\right\rangle_2 + b_{k+1} \left|1\right\rangle \otimes \left(\bigotimes_{i=k}^{1}(a_i \left|0\right\rangle + b_i \left|1\right\rangle)\right) \otimes \left|0\right\rangle_2 \tag{A.13}$$

$$= (a_{k+1} \left|0\right\rangle + b_{k+1} \left|1\right\rangle) \otimes \left(\bigotimes_{i=k}^{1}(a_i \left|0\right\rangle + b_i \left|1\right\rangle)\right) \otimes \left|0\right\rangle_2 \tag{A.14}$$

$$= \left(\bigotimes_{i=k+1}^{1} (a_i \left|0\right\rangle + b_i \left|1\right\rangle)\right) \otimes \left|0\right\rangle_2 \tag{A.15}$$

So the statement is true by induction. □