Quantum Walks and its Applications in Quantum Networks

Undergraduate Thesis [1] - André Nogueira Ribeiro
Supervisor: Prof. Fabio Kon (IME/USP)
Co-supervisor: Prof. Antonio Jorge Gomes Abelém (UFPA)
Supervisor at University of Massachusetts: Prof. Don Towsley(UMass)

Abstract

This undergraduate thesis explores the foundations and applications of quantum computing, starting with an overview of key concepts like qubits, superposition, and entanglement, and their role in quantum algorithms such as Shor's algorithm. The document highlights the potential of quantum computers, which can perform tasks beyond the capabilities of classical systems, despite challenges like qubit scaling, noise reduction, and coherence time. The thesis focuses on quantum walks, a quantum analog of classical random walks, and their application in quantum networks. It presents the development of a distributed quantum walk control plane, analyzing two approaches, linear and logarithmic, for encoding strategies and resource management. The research contributes to practical advancements in quantum network control and offers a foundation for future exploration of quantum internet architectures.

[1]This work was supported by FAPESP, procs. 14/50937-1, 2022/07594-2, 2023/06452-2 and the INCT of the Future Internet for Smart Cities funded by CNPq proc. 465446/2014-0.

Contents

1 Introduction

Quantum computing emerged from the idea of building a machine capable of simulating complex quantum systems, a concept proposed by physicist Richard Feynman and mathematician Yuri Manin. The challenge with simulating quantum systems on classical computers is that the computational effort increases exponentially with the size of the system, making it impractical for any significantly large problems. In addition to this focus, research in quantum algorithms has also achieved notable success. It has discovered algorithms with better complexity than classical ones being proposed, such as Shor's algorithm for number factorization.

Quantum computing is also an exciting field of study, offering a wide range of unexplored possibilities, which makes it both intriguing and promising. Currently, companies like Google and IBM have developed functional quantum machines that highlight the importance of this technology. These machines are capable of performing computations that are inaccessible to classical computers due to their distinct physical properties.

Some of the main challenges in advancing quantum computers include increasing the number of qubits (the basic unit of memory in quantum computing), reducing noise, and extending coherence time (the duration of quantum phenomena needed to execute an algorithm).

This undergraduate thesis explores the application of quantum walks in quantum networks, building upon foundational principles of quantum computing, quantum information, and quantum communication. The document begins with an introduction to quantum computing, covering key concepts such as qubits, superposition, and entanglement, as well as their applications in quantum algorithms and information processing. Quantum walks, the quantum analog of classical random walks, are then presented as powerful algorithmic tools for achieving quantum speedups and implementing distributed quantum operations.

The research focuses on the development of a distributed quantum walk control plane for quantum networks, addressing practical challenges in encoding, resource allocation, and operational efficiency. Two distinct approaches, linear and logarithmic, are presented and analyzed for their encoding strategies, resource trade-offs, and scalability. The study contributes to advancing the practical implementation of quantum walk protocols in quantum network control, laying the groundwork for further explorations into quantum internet architectures and applications.

Part of the code produced during the research can be found in the project repository. Additionally, we have a project website with more information and presentations.

2 Important Concepts

This section describes some important concepts which will be needed to understand the content of this undergraduate thesis. You may read the concepts as they appear on this document.

2.1 Linear Algebra

- Hilbert space: A Hilbert space is a mathematical concept that extends the idea of Euclidean space to infinite dimensions. It is an inner product space, meaning it comes equipped with an inner product, which allows for the definition of norms. The space is also complete, meaning that any Cauchy sequence of vectors within the space converges to a point within the space. This property of completeness ensures that Hilbert spaces are well-suited for analysis, especially in quantum mechanics, where they provide the framework for representing quantum states and operators. The inner product in a Hilbert space allows for the measurement of orthogonality, projection, and distance, making it a foundational tool in functional analysis, quantum theory, and many other areas of mathematics and physics.

- $\langle\psi|\,|\phi\rangle$: Notation for a inner product between ψ and ϕ.

- $|\psi\rangle\,|\phi\rangle$: Notation for a tensor product between ψ and ϕ.

- Normal operator: An operator A on V is normal if $A^\dagger A = AA^\dagger$.

- Unitary operator: An operator U is called unitary if it satisfies the condition: $U^\dagger U = UU^\dagger = I$, where $U^\dagger$ is the conjugate transpose of U, and I is the identity matrix. Unitary operators have several important properties: They are normal, meaning they are diagonalizable with respect to an orthonormal basis. The eigenvectors of a unitary operator associated with different eigenvalues are orthogonal, and each eigenvalue has unit modulus, meaning it has the form $e^{\imath\alpha}$, where α is a real number. Unitary operators preserve the inner product: for any vectors $|v_1\rangle$ and $|v_2\rangle$, $\langle Uv_1|Uv_2\rangle = \langle v_1|v_2\rangle$. As a result, the action of a unitary operator on a vector preserves its norm, ensuring that the vector's length remains unchanged.A complex square matrix U is unitary if its columns form an orthonormal set. Unitary evolution is fundamental in quantum mechanics, as unitary operators describe the evolution of closed quantum systems, maintaining consistent probabilities over time.

- Hermitian operator: An operator A on V is Hermitian or self-adjoint if $A^\dagger = A$.

- Positive operator: A positive operator A is defined to be an operator such that for any vector $|v\rangle$, $(|v\rangle,\ A\,|v\rangle)$ is a real, non-negative number. If $(|v\rangle,\ A\,|v\rangle)$ is strictly greater than zero for all $|v\rangle \neq 0$ then we say that A is positive definite. Any positive operator is automatically Hermitian, and therefore by the spectral decomposition has diagonal representation $\sum_i \lambda_i\,|i\rangle\,\langle i|$, with non-negative eigenvalues λ_i . [2]

- Wavefunction: On the quantum computing field, a wavefunction is a function that the x axis corresponds to the basis states and the y axis corresponds to the quantum amplitude of the state.

- Orthogonal Projector: An operator P on V is an orthogonal projector if $P^2 = P$ and $P^\dagger = P$. [3]

- With high probability (WHP): In mathematics, an event that occurs with high probability is one whose probability depends on a certain number n and goes to 1 as n goes to infinity.

- Matrix rank: In linear algebra, the rank of a matrix A is the dimension of the vector space generated (or spanned) by its columns. This corresponds to the maximal number of linearly independent columns of A.

2.2 Quantum Computation and Quantum Information

- Superposition: In quantum mechanics, superposition refers to the ability of a quantum system to exist in multiple states simultaneously. A quantum particle, such as an electron, can be in a combination of different states until it is measured, at which point it collapses into one definite state. Superposition is a non intuitive fundamental principle that enables the unique behavior of quantum systems.

- Entanglement: In quantum mechanics, entanglement is a phenomenon where two or more particles become linked in such a way that the state of one particle instantly influences the state of the other(s), no matter how far apart they are. This correlation persists even when the particles are separated by large distances. Entanglement is a key feature of quantum mechanics, enabling phenomena such as quantum teleportation and playing a crucial role in quantum computing and cryptography.

- Phase (relative): In quantum computing, phase refers to the angle that describes the position of a qubit's state in relation to a reference point. When a qubit is in superposition (a combination of both 0 and 1), its state has a magnitude (how much it is in 0 or 1) and a phase (which affects how the states interact). Phases are important because they influence the interference between quantum states, which can change the probability of measuring certain outcomes. The relative phase between states is critical for how quantum algorithms work and how qubits interact.

- Global phase: In quantum computing, the global phase refers to an overall phase factor applied to a quantum state that does not affect measurement outcomes. A quantum state is typically represented as a vector, and multiplying the entire state by a complex number of unit magnitude (e.g., $e^{i\theta}$) introduces a global phase. This phase has no observable consequences, as quantum measurements are only sensitive to relative phases between components of a superposition. Therefore, global phase usually can be ignored in practical computations and is considered irrelevant to the behavior or outcomes of a quantum system.

- Measurement: Quantum measurements are described by a collection M_m of measurement operators. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement then the probability that result m occurs is given by $p(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle$ and the state of the system after the measurement is $\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}$. The measurement operators satisfy the completeness equation, $\sum_m M_m^\dagger M_m = I$. The completeness equation expresses the fact that probabilities sum to one: $1 = \sum_m p(m) = \sum_m \langle\psi| M_m^\dagger M_m |\psi\rangle$. [2]

- Projective measurement: A projective measurement is described by an observable, M , a Hermitian operator on the state space of the system being observed. The observable has a spectral decomposition, $\sum_m mP_m$, where P_m is the projector onto the eigenspace of M with eigenvalue m. The possible outcomes of the measurement correspond to the eigenvalues, m, of the observable. Upon measuring the state $|\psi\rangle$, the probability of getting result m is given by $p(m) = \langle\psi| P_m |\psi\rangle$ .Given that outcome m occurred, the state of the quantum system immediately after the measurement is $\frac{P_m|\psi\rangle}{\sqrt{p(m)}}$. The average value of the observable M is often written $\langle M\rangle = \langle\psi| M |\psi\rangle$.[2]

- Quantum Error Correction: Quantum error correction is a set of techniques designed to protect quantum information from errors due to decoherence, noise, or other quantum disturbances. In quantum systems, qubits are fragile and susceptible to errors from their environment, which can lead to loss of information. Quantum error correction encodes logical qubits into entangled states of multiple physical qubits, allowing for the detection and correction of errors without directly measuring and collapsing the quantum state. It is a critical component in making quantum computers reliable and scalable for long computations, ensuring the integrity of quantum information over time.

- Pure or mixed state: On a Bloch sphere, pure states are represented by a point on the surface of the sphere, whereas mixed states are represented by an interior point. The completely mixed state of a single qubit is represented by the center of the sphere, by symmetry. The purity of a state can be visualized as the degree in which it is close to the surface of the sphere. In quantum mechanics, the state of a quantum system is represented by a state vector (or ket) $|\Psi\rangle$. A quantum system with a state vector $|\Psi\rangle$ is called a pure state. However, it is also possible for a system to be in a statistical ensemble of different state vectors: For example, there may be a 50% probability that the state vector is $|\Psi_1\rangle$ and a 50% chance that the state vector is $|\Psi_2\rangle$. This system would be in a mixed state. The density matrix is especially useful for mixed states, because any state, pure or mixed, can be characterized by a single density matrix.

- Density matrix: A density matrix, often denoted as ρ, is a mathematical representation used to describe the state of a quantum system, whether pure or mixed. For a pure state $|\Psi\rangle$, the density matrix is given by $\rho = |\Psi\rangle \langle\Psi|$. This matrix is Hermitian, positive semi-definite, and has a trace of 1. In contrast, for a mixed state, the density matrix is a weighted sum of pure states, representing a statistical ensemble of quantum states. Mathematically, for a system with N possible pure states $|\Psi_i\rangle$, each with probability p_i, the density matrix is:

$$\rho = \sum_{i=1}^{N} p_i |\Psi_i\rangle \langle\Psi_i| , \quad \text{where } \sum_{i=1}^{N} p_i = 1.$$

  The density matrix is particularly useful in quantum mechanics as it provides a complete description of the system's state, including information about coherence and probabilities. For pure states, $\mathrm{Tr}(\rho^2) = 1$, whereas for mixed states, $\mathrm{Tr}(\rho^2) < 1$. This property can be used to distinguish between pure and mixed states. Density matrices are essential in quantum information theory and quantum computation, enabling calculations for entanglement, decoherence, and state evolution.

- Trace (Tr()): The trace of a matrix, denoted as $\mathrm{Tr}(A)$, is the sum of the elements on its main diagonal. For a square matrix A of dimension $n \times n$, it is defined as:

$$\mathrm{Tr}(A) = \sum_{i=1}^{n} A_{ii}$$

  where A_{ii} are the diagonal elements of A. In quantum mechanics, the trace plays a key role, especially for density matrices. The trace of the density matrix ρ is always 1, ensuring the total probability of all quantum states equals 1 and the trace of the square of the density matrix, $\mathrm{Tr}(\rho^2)$, indicates whether the state is pure or mixed.

- Partial trace: It can be calculated as the $\sum_B \langle\phi_B| \rho_A \otimes \rho_B |\phi_B\rangle = \rho_A \otimes \sum_B \langle\phi_B| \rho_B |\phi_B\rangle$. In a more clarified way $tr_A(L_{AB}) = \sum_i \langle i|_A L_{AB} |i\rangle_A = \langle 0|0\rangle\langle 0|0\rangle(|1\rangle\langle 0|)_B + \langle 1|0\rangle\langle 1|1\rangle (|0\rangle\langle 0|)_B$
  $= (|1\rangle\langle 0|)_B$.

- Fidelity: A second measure of distance between quantum states is the fidelity. The fidelity is not a metric on density operators, but we will see that it does give rise to a useful metric. This section reviews the definition and basic properties of the fidelity. The fidelity of states ρ and σ is defined to be $F(\rho, \sigma) = tr\sqrt{\rho^{\frac{1}{2}} \sigma \rho^{\frac{1}{2}}}$. Fidelity does satisfy many of the same properties as the trace distance. For example, it is invariant under unitary transformations: $F(U\rho U^\dagger, U\sigma U^\dagger) = F(\rho, \sigma)$. The fidelity between a pure state $|\psi\rangle$ and an arbitrary state ρ is $\langle\psi| \rho |\psi\rangle$.[2]

- Quantum circuit depth: The depth of a circuit is a metric that calculates the longest path between the data input and the output. Each gate counts as a unit.

3 Introduction to Quantum Computing

Quantum computing represents a transformative approach to computation, fundamentally different from classical computing. At its core, quantum computing leverages principles of quantum mechanics, such as superposition, entanglement, and interference, to process information in ways that classical computers cannot. This approach opens up new possibilities for solving complex problems in fields like cryptography, material science, and artificial intelligence. This section introduces the foundational concepts of quantum computing, beginning with the qubit and extending to the mathematical framework that describes quantum states and operations. Through an exploration of quantum gates and their applications, we lay the groundwork for understanding how quantum algorithms can achieve exponential speedups over their classical counterparts, demonstrating the remarkable potential of quantum computing.

3.1 Qubit

A qubit (quantum bit) is the fundamental unit of memory in a quantum computer, and unlike classical bits, which are limited to being in a state of either 0 or 1, a qubit can exist in a superposition of both 0 and 1 simultaneously. This unique property is a consequence of quantum mechanics, which allows qubits to represent not just one of two discrete states, but an infinite number of possibilities between 0 and 1.

In more technical terms, a qubit's state is described as a quantum superposition, where it can be in a combination of both 0 and 1, weighted by complex numbers called probability amplitudes. However, this superposition exists only until the qubit is measured. Upon measurement, the qubit collapses to a definite state, either 0 or 1, with probabilities determined by its probability amplitudes before the measurement.

The power of quantum computing lies in the ability to leverage these superposition states without directly measuring them. Through quantum algorithms, it's possible to manipulate and process data while qubits remain in superposition, exploring multiple possible solutions simultaneously. This parallelism can provide a significant computational advantage over classical computers, in problems such as factoring large numbers or simulating quantum systems.

Moreover, qubits also exhibit another quantum property called entanglement, where the state of one qubit is intrinsically linked to the state of another, regardless of the distance between them. This further enhances the computational potential of quantum systems, enabling qubits to work together in ways that classical bits cannot, exponentially increasing the power of quantum computations.

3.2 Mathematical model

Studying quantum computing requires a understanding of linear algebra, as it forms the mathematical foundation for describing quantum systems. I will briefly cover the key concepts. Every quantum system can be represented as a vector in a Hilbert space. A qubit, the fundamental unit of information in a quantum computer, can be represented as a two-dimensional vector, and the operations, or quantum gates, that act on this qubit are represented by unitary matrices. Unitary matrices preserve the norm of the vector, ensuring the system's total probability remains 1.The state of a qubit is expressed as:

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle$$

where $|0\rangle$ and $|1\rangle$ are basis vectors, defined as $(1, 0)$ and $(0, 1)$ respectively. These basis states correspond to the classical bit values 0 and 1. The complex numbers a and b are the probability amplitudes. The squared magnitudes, $|a|^2$ and $|b|^2$, represent the probabilities of the qubit collapsing to the state 0 or 1, respectively, upon measurement. For these probabilities to make physical sense, the normalization condition must be satisfied:

$$|a^2| + |b^2| = 1$$

This ensures that the sum of probabilities for all possible outcomes equals 1.

When we move beyond a single qubit, the dimensionality of the system increases exponentially. A system of two qubits is represented by a four-dimensional vector, and in general, a system of n qubits is described by a vector in a 2^n-dimensional space. This exponential growth in dimensionality one thing which gives quantum computers advantage when compared to classical computers.

A common and intuitive way to visualize the state of a single qubit is through the Bloch sphere. The Bloch sphere represents the qubit as a point on the surface of a three-dimensional sphere, where the angles of the point correspond to the probability amplitudes and the global phase of the qubit. The global phase, while often not directly measurable, plays a crucial role in the evolution of the quantum system.

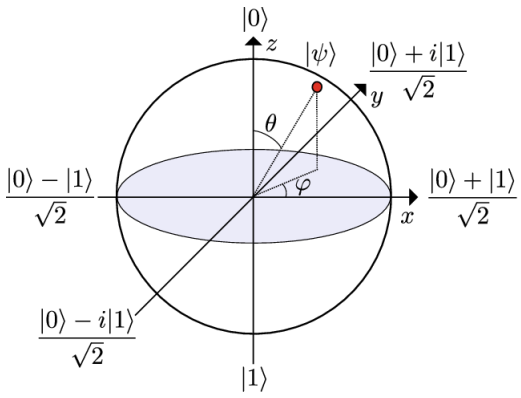


Figure 1: Bloch Sphere - source.

In addition to these representations, understanding quantum systems requires grasping the probabilistic nature of quantum mechanics. The outcomes of quantum measurements are not deterministic but are governed by probability distributions, dictated by the amplitudes of the quantum state vector. This probabilistic behavior, though counterintuitive from a classical perspective, is a key aspect of quantum computing and forms the basis for algorithms that can outperform their classical counterparts.

3.3 Quantum Gates

In a quantum circuit, gates play a crucial role in manipulating the quantum information that passes through them. Unlike classical gates, which operate on bits, quantum gates operate on qubits and are represented mathematically by unitary matrices. These matrices are square, meaning that a gate acting on a single qubit is represented by a 2×2 matrix, while a gate acting on two qubits is represented by a 4×4 matrix. The unitary nature of these matrices ensures that the operations are reversible, a key property of quantum computations.

Quantum gates are the building blocks of quantum algorithms, and understanding how they function is essential for constructing and analyzing quantum circuits. Below are some important quantum gates commonly used in quantum computations:

The Hadamard gate creates a superposition of quantum states, transforming a qubit from a definite state (either $|0\rangle$ or $|1\rangle$) into an equal probability superposition of both states. This gate is crucial for many quantum algorithms, such as Grover's search algorithm and Shor's factoring algorithm. Its matrix representation is:

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

The Pauli-X gate, also known as the quantum NOT gate, flips the state of a qubit. If the qubit is in the state $|0\rangle$, the X gate transforms it to $|1\rangle$, and vice versa. Its matrix representation is:

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

The CNOT (Controlled-Not) gate is a two-qubit gate that operates conditionally. Considering two quibits, control and target, it flips the state of the target qubit if and only if the control qubit is in the state $|1\rangle$. The CNOT gate plays a fundamental role in quantum entanglement and error correction. Its matrix representation for two qubits is:

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

The Toffoli gate is a three-qubit gate, also known as the controlled-controlled-X gate. It flips the state of the third qubit (the target) if and only if the first two qubits (the control qubits) are both in the state $|1\rangle$. This gate is essential for implementing reversible classical logic and can be used for quantum error correction.

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

The phase gate adds a phase factor to the quantum state without changing the probabilities of measuring the qubit in $|0\rangle$ or $|1\rangle$. Its matrix representation is:

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

The T gate is a phase gate that applies a phase shift of $\frac{\pi}{4}$ to the state $|1\rangle$. It is an important component for creating universal quantum gates and can be represented as:

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

Quantum gates provide the essential operations needed to construct quantum circuits and perform quantum algorithms. Each gate manipulates the quantum state of one or more qubits in specific ways, and understanding these transformations is fundamental to working with quantum information. Below are examples of operations using some of the quantum gates introduced, along with the resulting states of the qubits:

Consider a single qubit initially in the state $|0\rangle$. Applying the Hadamard gate H to this qubit creates a superposition state:

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

This operation transforms the qubit from a definite state $|0\rangle$ to an equal superposition of $|0\rangle$ and $|1\rangle$, a foundational step in quantum algorithms that require exploring multiple states simultaneously.

Suppose a qubit is in the state $|0\rangle$. Applying the Pauli-X gate, which acts as a quantum version of the classical NOT gate, flips the qubit's state:

$$X|0\rangle = |1\rangle$$

Similarly, if the qubit were initially in the state $|1\rangle$, the Pauli-X gate would transform it back to $|0\rangle$:

$$X|1\rangle = |0\rangle$$

Consider two qubits in the state $|0\rangle \otimes |0\rangle = |00\rangle$. First, we apply the Hadamard gate to the first qubit, resulting in the state:

$$(H \otimes I)|00\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$$

Then, applying the CNOT gate with the first qubit as the control and the second qubit as the target gives:

$$\text{CNOT}\left(\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)\right) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

This operation creates an entangled state, a key resource for quantum communication and computation.

Now consider three qubits in the state $|110\rangle$. Applying the Toffoli gate (CCX) to this state, with the first two qubits as controls and the third as the target, flips the state of the third qubit because both control qubits are in the state $|1\rangle$:

$$CCX\,|110\rangle = |111\rangle$$

If either of the control qubits were $|0\rangle$, the third qubit would remain unchanged, illustrating the conditional nature of the Toffoli gate.

Finally, consider a qubit in the state $|1\rangle$. Applying the Phase gate S to this qubit adds a phase factor of i to the $|1\rangle$ state:

$$S\,|1\rangle = i\,|1\rangle$$

Similarly, applying the T gate to $|1\rangle$ introduces a phase shift of $\frac{\pi}{4}$:

$$T\,|1\rangle = e^{i\frac{\pi}{4}}\,|1\rangle$$

These phase shifts are critical for creating complex interference patterns in quantum algorithms and are essential components in constructing universal quantum circuits.

Through the application of these gates, we observe how quantum operations differ significantly from classical logic gates. Quantum gates enable a variety of powerful transformations, including superposition, entanglement, and phase manipulation. Understanding and applying these gates allows for the construction of sophisticated quantum algorithms that leverage the unique properties of quantum mechanics to perform computations beyond the capabilities of classical systems.

4 Quantum Algorithms

Quantum algorithms are computational processes designed to harness the unique properties of quantum mechanics to solve certain problems more efficiently than classical algorithms.

One of the primary goals in developing quantum algorithms is to identify processes that can be executed with a polynomial number of quantum gates. In quantum computing, gates are the operations that manipulate qubits and thus form the steps of an algorithm. Algorithms requiring a polynomial number of gates are desirable because they remain efficient as problem sizes grow, ensuring that the quantum resources needed stay manageable.

By focusing on algorithms that require polynomial resources, researchers aim to make practical quantum computing a reality, enabling breakthroughs in fields such as cryptography, optimization, and materials science. In this section I will present some fundamentals and important quantum algorithms. Some of this content was based on learning provided by the IBM quantum material and qiskit.org

4.1 Deutsch's algorithm

Deutsch's algorithm is one of the earliest and simplest quantum algorithms, showcasing how quantum computing can solve certain types of problems more efficiently than classical computing. The problem it addresses is a simple example that demonstrates the power of quantum parallelism.

Imagine a black-box function, $f(x)$, that takes a single binary input x (either 0 or 1) and returns either 0 or 1. So, $f(x)$ has two possible inputs (0 and 1) and two possible outputs (0 or 1). There are four possible types of functions $f(x)$:

1. $f(0) = 0$ and $f(1) = 0$ (constant function returning 0 for both inputs)
2. $f(0) = 1$ and $f(1) = 1$ (constant function returning 1 for both inputs)
3. $f(0) = 0$ and $f(1) = 1$ (balanced function returning different values for each input)
4. $f(0) = 1$ and $f(1) = 0$ (also a balanced function)

The goal is to determine whether $f(x)$ is constant (both outputs are the same) or balanced (outputs are different).

Classically, we would need to check both inputs of $f(x)$ (i.e., $f(0)$ and $f(1)$) to determine if it is constant or balanced. This requires two function evaluations.

Quantumly, however, Deutsch's algorithm allows us to determine if $f(x)$ is constant or balanced with just one evaluation of $f(x)$. This efficiency comes from the ability of a qubit to exist in a superposition, allowing us to evaluate both inputs simultaneously. Below is how it works.

1. We start with two qubits:
   - The first qubit, in state $|0\rangle$, will hold the input to the function $f(x)$.
   - The second qubit, in state $|1\rangle$, is used to store the result of the function evaluation.

   So, our initial state is:
   $$|\psi_0\rangle = |0\rangle \otimes |1\rangle = |0, 1\rangle$$
2. Next, we apply the Hadamard gate to both qubits. The Hadamard gate creates a superposition state by transforming:
   - $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
   - $|1\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

   After applying Hadamard gates to both qubits, our state becomes:
   $$|\psi_1\rangle = \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle)$$

   This sets up the first qubit in a superposition of inputs 0 and 1, allowing us to evaluate both cases simultaneously.

3. Now, we apply the "oracle" or function gate, which encodes $f(x)$ into the second qubit. This operation, called the quantum oracle, flips the second qubit's sign based on the value of $f(x)$ without collapsing the state:

$$|\psi_2\rangle = \frac{1}{2}\left(|0\rangle \otimes (|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle) + |1\rangle \otimes (|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle)\right)$$

4. Next, we apply a Hadamard gate to the first qubit again. This final Hadamard gate transforms the state in such a way that, if $f(x)$ is constant, the first qubit will return to $|0\rangle$; if $f(x)$ is balanced, it will end up in $|1\rangle$.

   After applying this Hadamard gate, we measure the first qubit:

   - If the measurement outcome is 0, $f(x)$ is constant.
   - If the measurement outcome is 1, $f(x)$ is balanced.

The magic of Deutsch's algorithm lies in the ability of the quantum oracle and superposition to use interference (through the Hadamard gates) to reveal whether $f(x)$ is constant or balanced with just one evaluation. This is impossible in classical computing, where we would need at least two evaluations to determine the nature of $f(x)$.

Figure 2 presents the quantum circuit associated with Deutsch's algorithm.

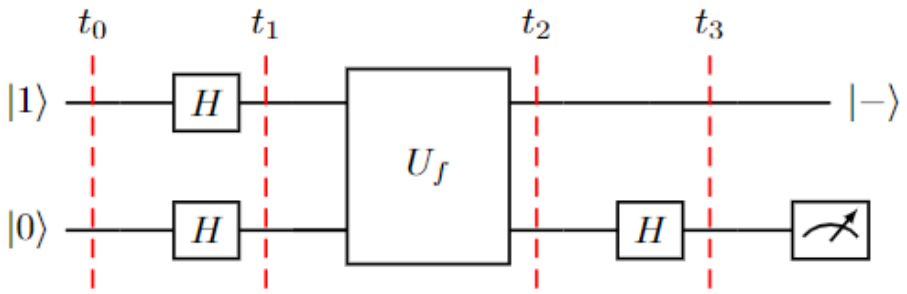


Figure 2: Deutsch's algorithm circuit - source.

4.2 Deutsch-Jozsa algorithm

The Deutsch-Jozsa algorithm is an extension of Deutsch's algorithm. It deals with a specific type of function. Suppose we have a function $f : \{0,1\}^n \to \{0,1\}$, meaning it takes an n-bit binary string as input and outputs either 0 or 1. The function $f(x)$, again, can be of two types: constant or balanced. In this case balanced means that the function outputs 0 for half of the inputs and 1 for the other half. Constant has the same meaning as before.

The goal is to determine if $f(x)$ is constant or balanced. Classically, we would need to evaluate $f(x)$ up to $2^{n-1} + 1$ times in the worst case to be certain, because we'd need to check enough inputs to ensure that it's not alternating between 0s and 1s.

In contrast, the Deutsch-Jozsa algorithm can determine whether $f(x)$ is constant or balanced with just one evaluation of the function, demonstrating an exponential speedup over classical approaches. Below is how it works

1. We start with $n+1$ qubits. The first n qubits represent the input to the function $f(x)$, and the last qubit will help with the evaluation:

   - Set the first n qubits to $|0\rangle$.
   - Set the last qubit to $|1\rangle$.

   So our initial state is:

$$|\psi_0\rangle = |0\rangle^{\otimes n} \otimes |1\rangle$$

2. Apply the Hadamard gate to each of the $n+1$ qubits. The Hadamard gate puts each qubit into a superposition, so we end up with all possible inputs (for the first n qubits) in superposition simultaneously. This transforms the state to:

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

The first n qubits are now in a superposition of all possible inputs, while the last qubit is in a superposition that allows it to act as a phase kickback in the next step.

3. Apply the Oracle (Quantum Function): Now we use a quantum oracle that applies the function $f(x)$ to the superposition state. The oracle is a black-box function that, for each input x, flips the phase of the state if $f(x) = 1$, but does nothing if $f(x) = 0$. This modifies our state to:

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Here, $(-1)^{f(x)}$ introduces a phase factor based on the value of $f(x)$, changing the amplitude of each input state in a way that allows us to determine whether $f(x)$ is constant or balanced.

4. Apply the Hadamard gate again to the first n qubits, transforming our state once more. This step is crucial, as it amplifies the amplitudes in a way that depends on whether $f(x)$ is constant or balanced. After this transformation, the state will collapse into one of two distinct results based on the type of $f(x)$.

5. Finally, we measure the first n qubits:

   - If the measurement outcome is $|0\rangle^{\otimes n}$, $f(x)$ is constant.
   - If the measurement outcome is anything other than $|0\rangle^{\otimes n}$, $f(x)$ is balanced.

The Deutsch-Jozsa algorithm exploits quantum parallelism and interference. By putting the input qubits in a superposition, we can evaluate $f(x)$ for all possible inputs at once. The phase kickback and interference through the Hadamard gates allow us to detect whether the function outputs are the same (constant) or balanced across the inputs, without needing multiple evaluations.

Figure 3 depicts the quantum circuit associated with Deutsch-Jozsa algorithm.

4.3 Simon's algorithm

Simon's algorithm is one of the foundational quantum algorithms, demonstrating an exponential speedup over classical algorithms for a specific type of problem. It was a precursor to Shor's algorithm and is significant because it proved that quantum computing could solve certain problems exponentially faster than classical computing.

In Simon's problem, we are given a black-box function $f : \{0,1\}^n \to \{0,1\}^n$ with the promise that there exists a secret string $s \in \{0,1\}^n$ such that:

- If $f(x) = f(y)$, then $x \oplus y = s$.

This means that $f(x)$ is 2-to-1: each output is produced by exactly two distinct inputs, and the difference (XOR) between these two inputs is the secret string s. The goal of Simon's algorithm is to determine the value of s.

Classically, solving Simon's problem requires $O(2^{n/2})$ evaluations of $f(x)$ to find s with high probability. In contrast, Simon's algorithm can find s using only $O(n)$ queries to $f(x)$ by taking advantage of quantum parallelism and interference.

1. Start with $2n$ qubits, where the first n qubits represent the input x and the second n qubits will store the output of $f(x)$. Initialize all qubits to $|0\rangle$:

$$|\psi_0\rangle = |0\rangle^{\otimes 2n}$$

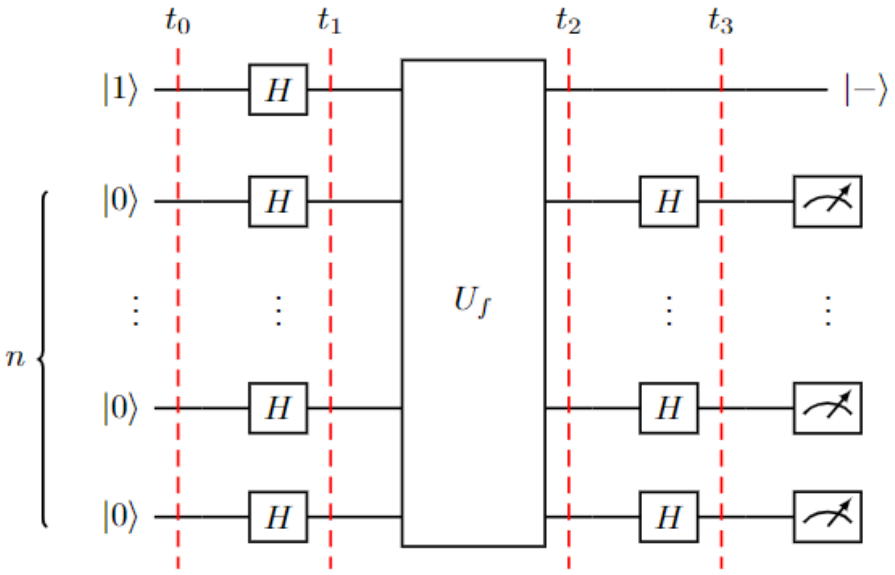


Figure 3: Deutsch Jozsa's algorithm circuit - source.

2. Apply Hadamard gates to the first n qubits, putting them into a superposition of all possible inputs. The state becomes:

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |0\rangle^{\otimes n}$$

3. Next, apply the oracle for $f(x)$, which maps $|x\rangle \otimes |0\rangle$ to $|x\rangle \otimes |f(x)\rangle$. After the oracle query, the state becomes:

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |f(x)\rangle$$

   Since $f(x)$ is 2-to-1, pairs of inputs differ by the secret string s and map to the same output. The result is an entangled state where knowing one x value gives a relationship to the other.

4. Measure the second n qubits. This collapses the state, and we are left with a superposition of the first register that includes only those x-values satisfying $x \oplus y = s$. This reduces our state to:

$$|\psi_3\rangle = \frac{1}{\sqrt{2}} \left(|x\rangle + |x \oplus s\rangle\right)$$

   for some random x.

5. Apply Hadamard gates to the first n qubits. This transforms each $|x\rangle$ and $|x \oplus s\rangle$ component, giving us a new state where we can measure information about s:

$$|\psi_4\rangle = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} (-1)^{x \cdot z} \left(|z\rangle + (-1)^{s \cdot z} |z\rangle\right)$$

   The measurement yields a bitstring z that satisfies $z \cdot s = 0 \mod 2$, giving us information about the secret string s.

6. Repeat the process n times to get n linearly independent equations for s. Solve this system of linear equations to determine s.

This provides an exponential speedup, requiring only $O(n)$ evaluations instead of $O(2^{n/2})$.

Simon's algorithm was one of the first to demonstrate an exponential quantum speedup. It showcases how quantum algorithms can solve specific types of problems faster than classical algorithms by leveraging superposition and interference. The principles used in Simon's algorithm laid the groundwork for more complex algorithms, such as Shor's algorithm for factoring.

Figure 4 represents the quantum circuit associated with Simon's algorithm.

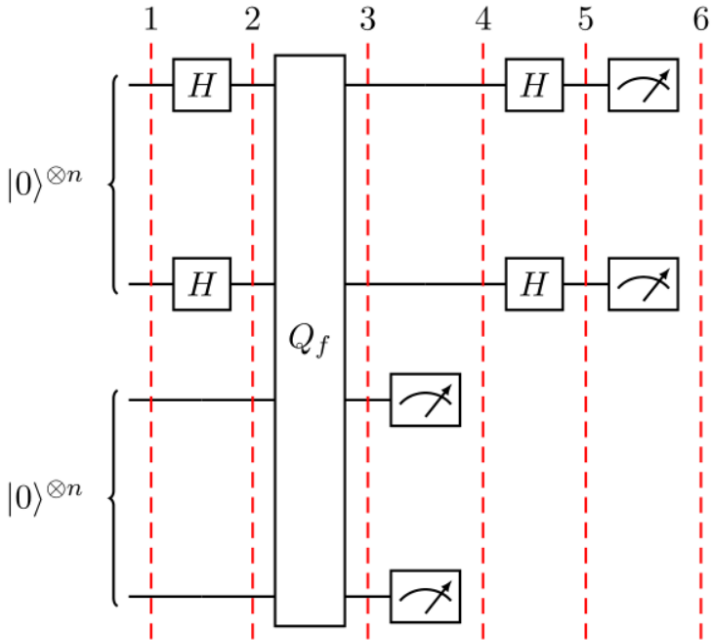


Figure 4: Simon's algorithm circuit - source.

4.4 Grover's algorithm

Grover's algorithm is a quantum search algorithm that provides a quadratic speedup over classical search algorithms. It is used to search an unsorted database or solve search problems, where the goal is to locate a specific "marked" item among N possible entries. While a classical search requires $O(N)$ operations to find the item in the worst case, Grover's algorithm can find it in $O(\sqrt{N})$ operations.

The problem assumes a black-box function (oracle) $f(x)$ defined as:

$$f(x) = \begin{cases} 1 & \text{if } x \text{ is the marked item,} \\ 0 & \text{otherwise.} \end{cases}$$

The oracle function identifies the marked item by returning 1 for the correct input $x = x_0$ and 0 otherwise. The goal is to find x_0 using as few queries to the oracle as possible.

Grover's algorithm is optimal for problems where the marked item needs to be located within an unstructured search space, making it useful for various applications in cryptography, optimization, and more.

1. Begin with n qubits in the $|0\rangle$ state. Apply a Hadamard gate to each qubit to create an equal superposition of all possible states:

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

where $N = 2^n$, the total number of possible states for n qubits.

2. **A**pply the oracle to flip the phase of the marked item. The oracle changes the amplitude of the target state $|x_0\rangle$ by introducing a phase of -1. The state after the oracle application becomes:

$$|\psi_1\rangle = \frac{1}{\sqrt{N}} \sum_{x \neq x_0} |x\rangle - \frac{1}{\sqrt{N}} |x_0\rangle$$

This marks the target item by shifting its amplitude, setting it up for amplification in the next steps.

3. Apply the Grover diffusion operator, which increases the amplitude of the marked state while decreasing the amplitudes of the other states. This operator consists of three steps:

   - Apply a Hadamard transform to all qubits.
   - Perform a phase flip on the $|0\rangle$ state (invert the amplitudes about the average).
   - Apply the Hadamard transform to all qubits again.

   After the amplitude amplification step, the state evolves closer to the target state. The diffusion operator is represented by:

$$D = 2 |\psi_0\rangle \langle\psi_0| - I$$

4. Repeat the oracle and diffusion steps approximately $O(\sqrt{N})$ times. Each iteration increases the amplitude of the marked state and decreases the amplitudes of the unmarked states, converging on the solution.

5. After performing the oracle and amplitude amplification steps $O(\sqrt{N})$ times, measure the qubits. With high probability, the result will be the marked item x_0.

The oracle marks the target state by inverting its phase, and the diffusion operator amplifies the amplitude of the marked state by rotating it closer to the target. By iterating these steps $O(\sqrt{N})$ times, Grover's algorithm amplifies the target state's amplitude, making it significantly more likely to be observed upon measurement.

Grover's algorithm illustrates the power of quantum computing for search problems. By reducing the search complexity from $O(N)$ to $O(\sqrt{N})$, it provides a quadratic speedup over classical methods. Although it is not an exponential improvement, this quadratic speedup is highly valuable for large search spaces and has potential applications in cryptography, database searching, and optimization.

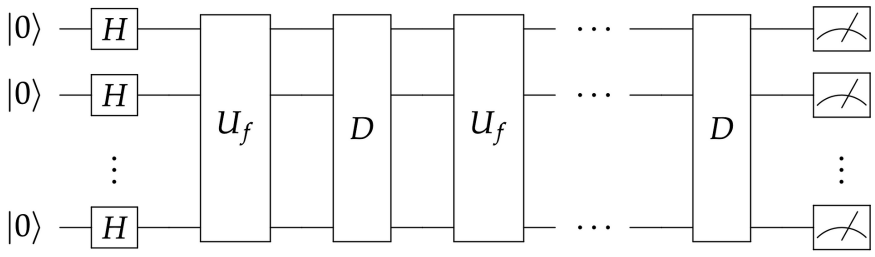


Figure 5: Grover's algorithm cirucuit - source.

For an easier understanding, Grover's algorithm has a geometric interpretation, which provides insight into why it efficiently locates the marked item. The algorithm's core process—phase inversion and amplitude amplification—can be visualized as a sequence of rotations in a multi-dimensional space.

Imagine that the quantum state space is represented by a vector space where each possible basis state (each possible value of x in the search space) corresponds to a unique axis in the space. In this space, we can define two important vectors:

- ($|x_0\rangle$): This vector represents the quantum state corresponding to the marked item, which we want to identify.

- ($|s\rangle$): This vector represents an equal probability amplitude over all possible states, constructed as:

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

where $N = 2^n$, the total number of possible states for n qubits.

The initial state of the algorithm, after applying Hadamard gates to all qubits, is $|s\rangle$, a uniform superposition over all possible items.

We can think of $|s\rangle$ as forming an angle with $|x_0\rangle$ (the solution state). Since $|s\rangle$ is an equal superposition of all possible states, it can be decomposed into two components:

- A component along the direction of $|x_0\rangle$.
- A component perpendicular to $|x_0\rangle$ (we can call this the "non-solution" space).

Let $|w\rangle$ represent the direction orthogonal to $|x_0\rangle$, which spans all states except the marked state. Thus, we can decompose $|s\rangle$ as:

$$|s\rangle = \alpha |x_0\rangle + \beta |w\rangle$$

where α and β are real coefficients representing the projection of $|s\rangle$ onto $|x_0\rangle$ and $|w\rangle$, respectively.

Grover's algorithm uses two main operations that act as rotations in this vector space:

1. The oracle marks the solution state by flipping the phase of $|x_0\rangle$, effectively rotating $|s\rangle$ through the angle between $|s\rangle$ and $|x_0\rangle$. This is equivalent to reflecting $|s\rangle$ across the hyperplane orthogonal to $|x_0\rangle$.
2. The Grover diffusion operator acts as a reflection about $|s\rangle$. This reflection moves the vector closer to $|x_0\rangle$ by effectively doubling the amplitude of the marked state (the component of $|s\rangle$ in the $|x_0\rangle$ direction).

Each application of the oracle and diffusion operators constitutes one "Grover iteration." These two reflections, applied in sequence, effectively rotate the state vector by a fixed angle closer to $|x_0\rangle$ in each iteration.

The rotation angle per Grover iteration is approximately:

$$\theta = 2 \arcsin\left(\frac{1}{\sqrt{N}}\right)$$

This angle depends on the size of the search space, N, and determines how quickly the algorithm converges to the solution state. To maximize the probability of measuring the marked state $|x_0\rangle$, Grover's algorithm requires approximately:

$$k = \frac{\pi}{4}\sqrt{N}$$

iterations, or applications of the oracle and diffusion operators, to rotate the state vector as close as possible to $|x_0\rangle$. The figure 6 makes it more tangible.

After $O(\sqrt{N})$ iterations, the state vector will be very close to $|x_0\rangle$, and a measurement of the state will yield the marked item with high probability. This geometric interpretation highlights why Grover's algorithm is optimal for unstructured search: each iteration moves the state vector closer to the target solution, and the quadratic speedup comes from the number of rotations required to reach the solution state, which scales as $\sqrt{N}$.

4.5 Shor's algorithm

Shor's algorithm is a quantum algorithm that efficiently finds the prime factors of a large integer N, solving the integer factorization problem. This problem is crucial because the security of many classical cryptographic systems, such as RSA, relies on the difficulty of factoring large integers. Shor's algorithm achieves exponential speedup over the best-known classical algorithms, making it a key example of quantum computing's potential.

Given a large integer N, Shor's algorithm aims to find a nontrivial factor of N. The algorithm relies on the following insight from number theory:

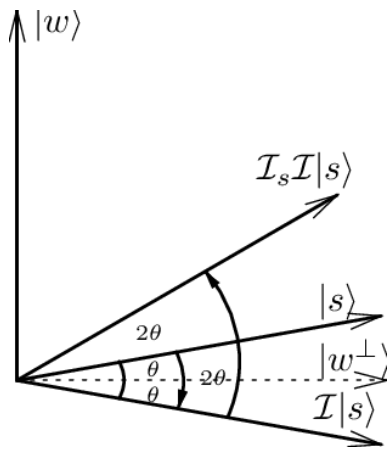


Figure 6: Grover's algorithm geometric intuition - source.

- For a random integer a (where $1 < a < N$), if a and N are coprime, then there exists a smallest positive integer r (called the order) such that:

$$a^r \equiv 1 \pmod{N}$$

- If r is even, then with high probability, the factors of N are given by the greatest common divisor of N with $a^{r/2} - 1$ and $a^{r/2} + 1$.

Thus, finding the period r allows us to factor N. Classically, finding r would require exponential time, but quantum computing can efficiently determine r using a technique called the Quantum Fourier Transform (QFT). Following are the algorithm steps.

1. Choose a random integer a where $1 < a < N$ and $\gcd(a, N) = 1$.
2. The main quantum component of Shor's algorithm is finding the period r of the function $f(x) = a^x \mod N$. To find r, we set up a quantum state that represents this function.
3. Initialize two quantum registers:
   - The first register holds an equal superposition of all possible values of x (from 0 to some $2^n - 1$), created using Hadamard gates.
   - The second register starts in the $|0\rangle$ state and will store the values $f(x) = a^x \mod N$.

   The initial state after applying Hadamard gates to the first register is:

$$|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |0\rangle$$

4. Using a quantum circuit, apply the function $f(x) = a^x \mod N$ to entangle the first register with the output in the second register:

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |f(x)\rangle$$

   This state encodes periodicity information related to the order r.

5. To extract the period r, we apply the Quantum Fourier Transform to the first register. The QFT transforms a superposition of states into a frequency space that reveals periodic patterns.

Quantum Fourier Transform (QFT)

The Quantum Fourier Transform is the quantum analog of the discrete Fourier transform (DFT) and is essential for finding periodicity in quantum algorithms. For a state $|x\rangle = \sum_{k=0}^{N-1} x_k |k\rangle$, the QFT is defined as:

$$\text{QFT}(|x\rangle) = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i x y/N} |y\rangle$$

where $N = 2^n$ for an n-qubit system.

In Shor's algorithm, the QFT allows us to move from a superposition over the computational basis to one where the periodicity of $f(x) = a^x \mod N$ becomes measurable.

The QFT circuit consists of Hadamard gates and controlled phase rotations, with the following steps for an n-qubit register:

(a) Apply a Hadamard gate to the first qubit.

(b) Apply controlled phase rotations between each qubit, such that each subsequent qubit has a phase shift based on the values of the previous qubits.

(c) Repeat this process for each qubit, applying decreasing rotations as the circuit progresses.

(d) Finally, perform a "swap" operation to reverse the order of the qubits.

The QFT transforms the state so that measuring it will reveal information about the period r in the frequency domain.

6. After applying the QFT, measure the first register. The result will likely yield an integer k that is related to the period r by:

$$k \approx \frac{j}{r}$$

for some integer j. Using continued fractions, we can compute r from k with high probability.

7. Once r is found, check if it is even. Then compute the factors of N as $\gcd(a^{r/2} - 1, N)$ and $\gcd(a^{r/2} + 1, N)$. If these values yield nontrivial factors, the algorithm is complete. Otherwise, repeat the process with a new random a.

Shor's algorithm leverages the quantum superposition and the Quantum Fourier Transform to efficiently find the period r, a task that is exponentially hard on classical computers. By encoding the periodicity of $f(x) = a^x \mod N$ in a quantum state and using the QFT to extract this periodicity, Shor's algorithm allows us to find factors of N with only polynomial quantum resources.

Shor's algorithm demonstrates the power of quantum computing for number-theoretic problems, achieving an exponential speedup in factoring large numbers. The algorithm combines classical number theory, modular exponentiation, and the Quantum Fourier Transform, which plays a central role in revealing periodicity. This algorithm has substantial implications for cryptography, as it provides an efficient means of breaking widely-used cryptographic systems like RSA.

Figure 7 presents the associated quantum circuit.

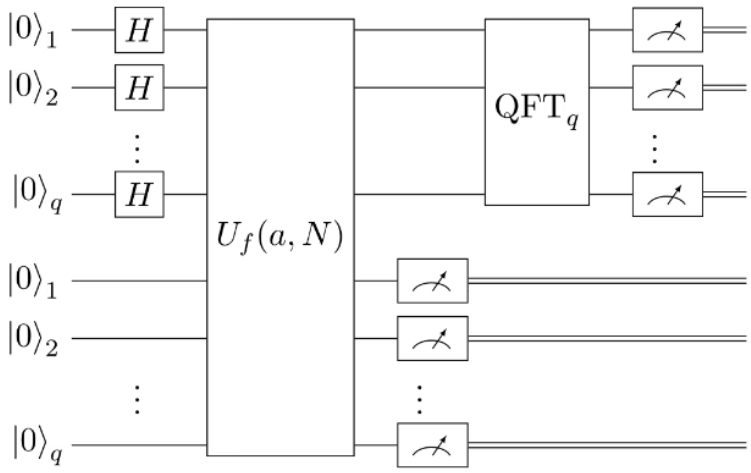


Figure 7: Shor's algorithm circuit - source.

5 Introduction to Quantum Information

The vision behind developing a quantum internet is to enable new technologies through quantum communication between any two points on the planet. Significant applications of quantum communication have already been identified, including secure communication, clock synchronization, secure user identification, exponential savings in communication, and secure access to remote quantum computers via the cloud.

Quantum computing operates under principles that differ substantially from classical computing, as proposed by Alan Turing. Quantum communication utilizes the laws of quantum physics to protect and transmit information. This can be done through the transfer of a quantum state (entangled or otherwise), the creation of an entangled state, or the use of a pre-established entangled state.

However, achieving quantum communication at scale presents several challenges: the difficulty of creating and maintaining quantum states, the challenge of transmitting qubits, and the need for integrating quantum systems with classical systems. [4]

Before delving deeper, it is essential to understand Bell states. Also known as EPR pairs or maximally entangled states, Bell states are four specific quantum states of two qubits, named after physicist John S. Bell, who discovered them in 1964. Bell states are critical in quantum information and communication because they are maximally entangled, meaning their states cannot be described independently. When one qubit in a Bell state pair is measured, the state of the other qubit is instantly determined, regardless of the distance between them. Bell states have applications in quantum teleportation, quantum key distribution, and other quantum communication protocols.

The four Bell states are:

$$|\Phi^+\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$$
$$|\Phi^-\rangle = \frac{|00\rangle-|11\rangle}{\sqrt{2}}$$
$$|\Psi^+\rangle = \frac{|01\rangle+|10\rangle}{\sqrt{2}}$$
$$|\Psi^-\rangle = \frac{|01\rangle-|10\rangle}{\sqrt{2}}$$

A fundamental protocol in this field is quantum teleportation, which allows for the exact transfer of a qubit's quantum state to another qubit, essential because a qubit's state cannot be cloned. This protocol reflects a security advantage: quantum information cannot be copied or verified mid-process, making it inherently secure. Additionally, the entanglement property enables rapid and efficient communication, as it implies a correlation that is independent of the distance between the entangled particles. Figure 8 presents the associated circuit.

Another interesting exercise is the CHSH game (Clauser-Horne-Shimony-Holt), a game involving two players, Alice and Bob, who can communicate beforehand. Once the game begins, each player receives a random bit (0 or 1) from a referee and must return one of two possible values. Without further communication, they win if the exclusive OR of their outputs equals the logical AND of their inputs.

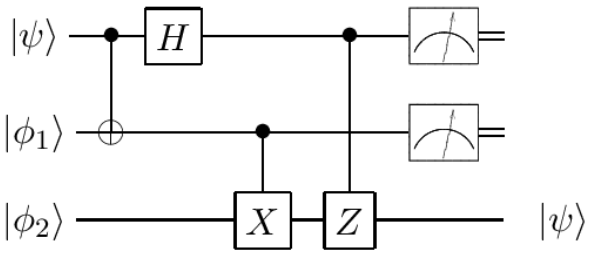


Figure 8: Quantum teleportation cirucuit - source.

Classical strategies allow them to win 75% of the time on average, but by using entangled qubits, they can increase their success rate, demonstrating a clear advantage of quantum communication.

Next, I will explain two critical protocols in the field of quantum communication and the quantum internet, *entanglement swapping* and *quantum key distribution* (QKD).

5.1 Entanglement Swapping

Entanglement Swapping is a quantum communication protocol that enables two distant parties to share entanglement without any direct interaction between them. This protocol is fundamental for building quantum communication networks and creating long-distance entanglement across large distances, which is essential for secure quantum communication. Here's a detailed explanation of the process:

First, it involves two pairs of entangled qubits. We label these pairs as:

- Pair AB, where qubits A and B are entangled.
- Pair CD, where qubits C and D are entangled.

Initially, the entangled pairs are in the following state:

$$|\Phi^+\rangle_{AB} \otimes |\Phi^+\rangle_{CD} = \frac{1}{2}(|00\rangle_{AB} + |11\rangle_{AB}) \otimes (|00\rangle_{CD} + |11\rangle_{CD})$$

1. The qubits from each pair are separated and sent to different locations: - Qubit A is sent to Alice. - Qubit D is sent to Bob. - Qubits B and C are sent to an intermediate station, often referred to as a "measurement node."

2. At the measurement node, a joint measurement is performed on qubits B and C to project them into one of the four possible Bell states. This measurement entangles the qubits at the measurement node and collapses them into a known entangled state. The four possible Bell states for qubits B and C are:
$$|\Phi^+\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$$
$$|\Phi^-\rangle = \frac{|00\rangle-|11\rangle}{\sqrt{2}}$$
$$|\Psi^+\rangle = \frac{|01\rangle+|10\rangle}{\sqrt{2}}$$
$$|\Psi^-\rangle = \frac{|01\rangle-|10\rangle}{\sqrt{2}}$$

3. When qubits B and C are measured in a Bell state, this measurement "swaps" the entanglement. This means that qubits A and D, which were previously unentangled, are now entangled with each other. The measurement on B and C projects the state of A and D into an entangled state. Importantly, this occurs without any direct interaction between qubits A and D.

4. After the measurement, qubits A and D are in one of the four possible Bell states, depending on the outcome of the measurement on qubits B and C. The specific entangled state of A and D can be determined based on the measurement result at the measurement node. Alice and Bob can then use this shared entanglement for further quantum communication tasks, such as quantum teleportation or quantum key distribution (QKD).

Entanglement Swapping has critical implications for quantum communication networks because it allows entanglement to be established over long distances without the need for a direct quantum link between the communicating parties. Instead of relying on a single quantum channel connecting Alice and Bob, entanglement swapping enables the creation of long-distance entanglement through intermediate nodes, which can be chained together to create a quantum repeater network.

By implementing entanglement swapping at multiple nodes, a "quantum repeater" architecture can be created, where entanglement is extended across increasingly larger distances. This process is crucial for overcoming the distance limitations of direct quantum communication, as qubits transmitted over long distances (e.g., via optical fiber or free-space optical links) are prone to errors and decoherence. Through entanglement swapping, secure quantum communication can be maintained across vast distances with high fidelity, providing a foundation for a global quantum internet.

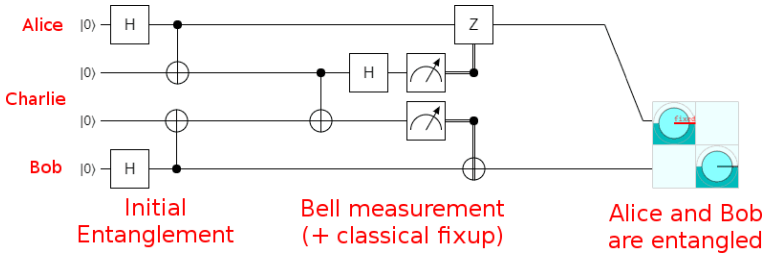


Figure 9: Entanglement Swapping Protocol - source.

5.2 Quantum Key Distribution

Quantum Key Distribution (QKD) is a quantum cryptographic protocol that enables two parties to securely share an encryption key, making it impossible for a third party to intercept without detection. QKD relies on the principles of quantum mechanics to generate cryptographic keys that are intrinsically secure, as any attempt to eavesdrop on the key would disturb the quantum states and be immediately detectable.

QKD involves two main participants, commonly called Alice (the sender) and Bob (the receiver). They share a quantum channel (such as optical fiber or free-space optical links) for the transmission of qubits and a classical communication channel for error correction and verification. The basic idea is to use the properties of quantum mechanics to generate a shared, random key between Alice and Bob that can be used to encrypt and decrypt messages securely.

The most widely known implementation of QKD is the BB84 protocol, developed by Charles Bennett and Gilles Brassard in 1984. The steps of this protocol are as follows:

1. Alice prepares a sequence of qubits, each randomly polarized in one of two bases:

   - The rectilinear basis (0° and 90°) where $|0\rangle$ and $|1\rangle$ represent the two states.
   - The diagonal basis (45° and 135°) where $|+\rangle$ and $|-\rangle$ represent the two states.

   Alice randomly chooses the basis and polarization for each qubit and sends the sequence of qubits to Bob over the quantum channel.

2. Upon receiving the qubits, Bob measures each qubit in a randomly chosen basis (either rectilinear or diagonal). Due to the principles of quantum mechanics, if Bob chooses the correct basis (matching Alice's choice), he will measure the correct state. If he chooses the wrong basis, he will measure a random result.

3. After all qubits have been transmitted and measured, Alice and Bob communicate over the classical channel to disclose the bases they used for each qubit. They do not reveal the actual measurement outcomes, only the choice of basis. They discard any measurements where they used different bases and retain only the results where their bases matched.

4. To ensure that no eavesdropping has occurred, Alice and Bob compare a subset of their remaining bits (the ones with matching bases) to check for errors. If the error rate is within an acceptable

range (indicating that eavesdropping is unlikely), they proceed with the key generation process. Otherwise, they discard the key and restart the protocol.

5. After correcting for any detected errors, Alice and Bob perform a privacy amplification process to further secure the key. This step reduces any partial information an eavesdropper might have obtained by compressing the shared key, ensuring that only Alice and Bob have a secure and private encryption key.

The security of QKD is guaranteed by the laws of quantum mechanics, particularly the no-cloning theorem and the disturbance caused by measurement. The no-cloning theorem states that it is impossible to create an identical copy of an unknown quantum state. Therefore, an eavesdropper (Eve) cannot intercept and reproduce the qubits without introducing detectable disturbances.

Furthermore, if Eve tries to measure the qubits in transit, she will introduce errors due to the random choice of measurement bases by Alice and Bob. This disturbance will reveal her presence, as the error rate in the transmitted key will increase. By comparing a subset of their key bits, Alice and Bob can detect any anomalies that indicate eavesdropping.

QKD has important applications in secure communications, especially in sectors requiring high levels of confidentiality, such as government communications, financial transactions, and military operations. By enabling the secure sharing of encryption keys, QKD provides a foundation for secure communication in a world where classical encryption could potentially be broken by quantum computers.

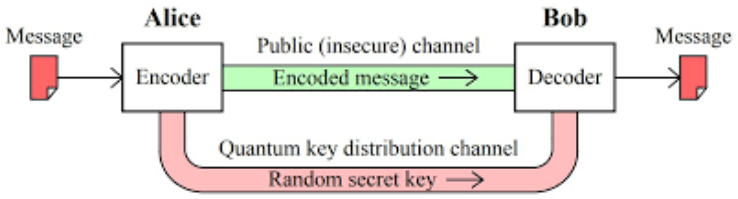


Figure 10: Quantum Key Distribution (QKD) Protocol. Alice sends a sequence of qubits to Bob, who measures them. They then compare their measurement bases over a classical channel to generate a secure, shared encryption key - source.

5.3 Quantum Internet

A quantum internet is a network designed to enable the secure and efficient transmission of quantum information, allowing for quantum communication and computing on a global scale. Unlike the classical internet, which transmits bits, a quantum internet transmits qubits, the fundamental units of quantum information. Establishing a quantum internet involves overcoming significant challenges and requires three essential components: a physical medium for qubit transmission, methods to extend transmission distances, and quantum processors connected to the network to process quantum information.

A functional quantum internet requires three main elements:

1. Physical Connection for Qubit Transmission: Qubits can be transmitted through various channels, including optical fibers and free-space optical communication. Optical fiber, commonly used in conventional telecommunications, can transmit qubits over short to medium distances, but standard fibers are inadequate for quantum communication over long distances due to high error rates and decoherence. To address this, researchers have developed specialized optical fibers, such as photonic crystal fibers, designed to minimize decoherence and better support qubit transmission. However, even with these advancements, error rates increase quadratically with distance in optical fibers, making long-distance transmission challenging.

   Free-space optical communication, which uses lasers to transmit information through open air, is an alternative to fiber optics. This approach has been used to establish quantum communication links between ground stations and satellites in orbit, enabling much longer distances than achievable with fiber optics. While promising, free-space optical communication requires precise alignment and is subject to environmental disturbances, such as atmospheric conditions.

2. Quantum Repeaters for Extended Distances: In both fiber-optic and free-space systems, extending transmission distances within a quantum network requires *quantum repeaters*. Quantum repeaters are specialized devices that amplify and maintain quantum signals, effectively increasing the range of quantum communication. Each quantum repeater consists of various components, including information processors, entanglement generation mechanisms, and storage for qubits. The primary challenge for quantum repeaters is to preserve the fragile quantum information during signal amplification, which is accomplished using *quantum error correction protocols* to maintain the integrity of the transmitted qubits.

   Quantum repeaters create entanglement between distant nodes in the network by linking shorter segments of entangled qubits, a process known as entanglement swapping. This allows for the extension of quantum communication over vast distances, making long-distance secure quantum communication feasible.

3. Quantum Processors for Information Processing: The final element of a quantum internet is the quantum processors connected to the network, which handle and process quantum information. Quantum processors act as nodes in the quantum network, receiving, manipulating, and transmitting quantum data. They play a crucial role in performing quantum operations required for tasks like quantum key distribution (QKD), entanglement generation, and quantum error correction. Quantum processors are essential for the execution of distributed quantum algorithms, enabling applications that rely on shared quantum resources.

The purpose of a quantum internet is not to replace classical communication but to complement it. A quantum internet would enable applications that are impossible to achieve with classical networks alone, offering enhanced security and efficiency for specific types of data transmission and computational tasks. The integration of quantum and classical networks allows for hybrid systems where quantum communication protocols, such as QKD, secure classical channels or provide computational advantages. [4]

The development of a quantum internet holds promise for a range of applications, including:

- Secure Communication: Quantum networks enable ultra-secure communication methods resistant to eavesdropping, as the principles of quantum mechanics prevent undetected interception.
- Distributed Quantum Computing: A quantum internet would allow remote access to quantum computers, enabling distributed quantum computing where complex computations can be performed across multiple quantum processors.
- Enhanced Computational Protocols: Quantum entanglement and superposition in a quantum internet enable computational protocols, such as quantum teleportation and entanglement swapping, for efficient communication and resource sharing.

In summary, a quantum internet would represent a paradigm shift in secure communication and computational power, enabling breakthroughs in fields that require advanced information security, distributed quantum computing, and new computational techniques. By combining quantum and classical systems, the quantum internet would open new frontiers in secure and efficient global communication.

As in all quantum field, quantum internet does suffer with noise. Following are some important types of noise which are studied in this area. Those provide an enormous challenge for making the technology possible. Also it makes the mathematical model more complex.

- Bit flip: The bit flip channel flips the state of a qubit from $|0\rangle$ to $|1\rangle$ (and vice versa) with probability $1-p$. On the Bloch sphere representation you can see that the states on the $\hat{x}$ axis are left alone, while the $\hat{y}-\hat{z}$ plane is uniformly contracted by a factor of $1-2p$.
- Phase flip: The phase flip channel has operation elements $E_0 = \sqrt{p}\ I, E_0 = \sqrt{1-p}\ Z$. On the Bloch sphere, the states on the $\hat{z}$ axis are left alone, while the $\hat{x}-\hat{y}$ plane is uniformly contracted by a factor of $1-2p$.
- Bit-phase flip: The bit-phase flip channel has operation elements $E_0 = \sqrt{p}\ I, E_0 = \sqrt{1-p}\ Y$, where $Y = \imath XZ$. On the Bloch sphere, the states on the $\hat{y}$ axis are left alone, while the $\hat{x}-\hat{z}$ plane is uniformly contracted by a factor of $1-2p$.

- Depolarizing channel: The depolarizing channel is an important type of quantum noise. Imagine we take a single qubit, and with probability p that qubit is depolarized. That is, it is replaced by the completely mixed state, $I/2$. With probability $1-p$ the qubit is left untouched. The state of the quantum system after this noise is $\varepsilon(p) = \frac{pI}{2} + (1-p)\rho$. On the Bloch sphere, the entire sphere contracts uniformly as a function of p. [2]

- Amplitude damping: In this case, the E_1 operation changes a $|1\rangle$ state into a $|0\rangle$ state, corresponding to the physical process of losing a quantum of energy to the environment. E_0 leaves $|0\rangle$ unchanged, but reduces the amplitude of a $|1\rangle$ state; physically, this happens because a quantum of energy was not lost to the environment, and thus the environment now perceives it to be more likely that the system is in the $|0\rangle$ state, rather than the $|1\rangle$ state. On the Bloch sphere, the entire sphere shrinks towards the north pole, the $|0\rangle$ state. [2]

- Phase damping: A noise process that is uniquely quantum mechanical, which describes the loss of quantum information without loss of energy, is phase damping. A very simple model for this kind of quantum noise is the following. Suppose that we have a qubit $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ upon which the rotation operation $Rz(\theta)$ is applied, where the angle of rotation θ is random. The random phase kicking causes the expected value of the off-diagonal elements of the density matrix to decay exponentially to zero with time. That is a characteristic result of phase damping. [2]

6 Introduction to Quantum Walks

Quantum walks are a powerful algorithmic tool in quantum computing. They have proven to be universal for quantum computing [8] and have provided quantum speed-ups in a myriad of applications. Despite all their generality and properties, quantum walks are simple. In particular, discrete-time coined quantum walks are described by the product of two operators, known as the coin and shift operators, which are sufficient to implement arbitrary quantum algorithms.

They are the quantum analog of classical random walks but, unlike them, which are probabilistic and rely on flipping a coin to determine the next position, quantum walks are inherently different due to the principles of quantum superposition and entanglement. This results in a richer, more complex behavior that can be useful in achieving quantum algorithms.

6.1 One Dimensional Discrete Time Random Walks

A one dimensional discrete time random walk is a simple yet powerful model that describes the path of a "walker" moving along a line. At each discrete time step, the walker makes a move to the left or right, depending on a random event, such as the outcome of a coin flip. This stochastic process is widely used in physics, economics, computer science, and many other fields to model systems that evolve over time under randomness.

In a one dimensional discrete time random walk, we define the following:

- Let X_t denote the position of the walker at time t.
- The initial position $X_0 = 0$ is usually set to the origin.
- At each time step t, the walker moves either one unit to the left or one unit to the right, based on a probabilistic rule.

The movement rule is simple:

- With probability $p = \frac{1}{2}$, the walker moves one step to the right, increasing its position by 1.
- With probability $q = \frac{1}{2}$, the walker moves one step to the left, decreasing its position by 1.

We can describe the position of the walker after t steps as:

$$X_t = X_0 + \sum_{i=1}^{t} S_i$$

where S_i represents the i-th step, with $S_i = +1$ for a step to the right and $S_i = -1$ for a step to the left. Since each step is independent and equally likely, S_i is a random variable with an expectation $E[S_i] = 0$ and variance $\mathrm{Var}(S_i) = 1$.

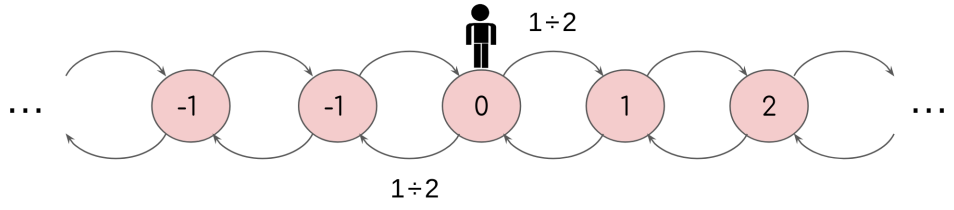


Figure 11: One Dimensional Discrete Time Random Walk with start position equal to zero.

Below is an example of a one dimensional discrete time random walk over five steps. Starting at $X_0 = 0$, suppose we flip a fair coin to determine each step:

- **Step 1**: The coin lands on heads, so the walker moves right. $X_1 = X_0 + 1 = 1$.
- **Step 2**: The coin lands on tails, so the walker moves left. $X_2 = X_1 - 1 = 0$.

- **Step 3**: The coin lands on heads, so the walker moves right. $X_3 = X_2 + 1 = 1$.
- **Step 4**: The coin lands on heads again, so the walker moves right. $X_4 = X_3 + 1 = 2$.
- **Step 5**: The coin lands on tails, so the walker moves left. $X_5 = X_4 - 1 = 1$.

After five steps, the walker's final position is $X_5 = 1$.

For a large number of steps, the probability distribution of the walker's position X_t tends to follow a normal (Gaussian) distribution centered around the initial position, with a variance that increases with time. Specifically, after t steps, the expected value of X_t is zero:

$$E[X_t] = 0$$

and the variance is given by:

$$\text{Var}(X_t) = t$$

Thus, the standard deviation of the position after t steps is $\sqrt{t}$. This implies that the probability of finding the walker at a specific position spreads out over time, following a bell curve centered at the origin.

The probability $P(X_t = k)$ that the walker is at position k after t steps can be calculated using the binomial distribution. Since each step has an equal chance of being to the left or right, we can express this probability as:

$$P(X_t = k) = \binom{t}{\frac{t+k}{2}} p^{\frac{t+k}{2}} q^{\frac{t-k}{2}}$$

where $\binom{t}{\frac{t+k}{2}}$ is the binomial coefficient, and $\frac{t+k}{2}$ and $\frac{t-k}{2}$ must be integers. This formula gives the exact probability distribution for the position of the walker at each step.

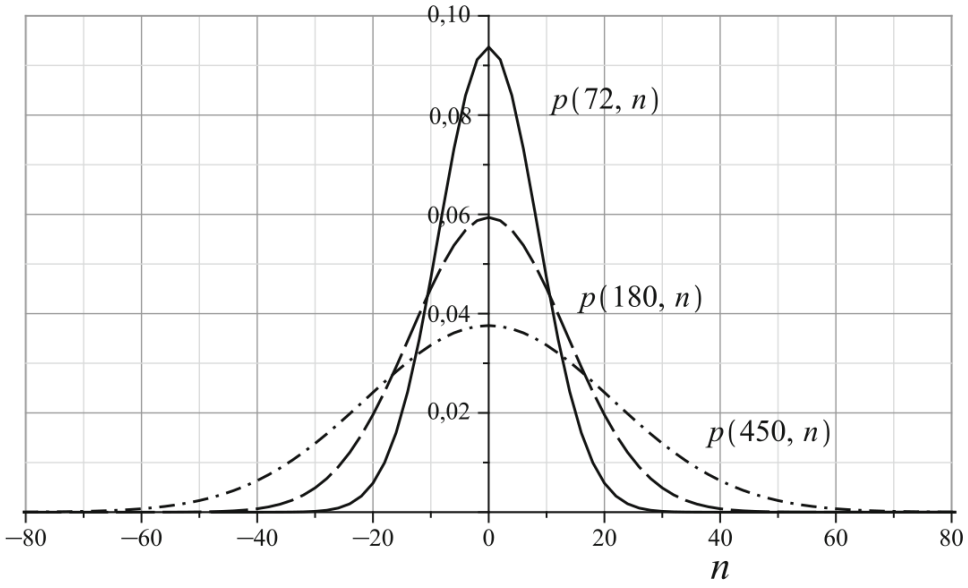


Figure 12: Probability Distribution of a One Dimensional Discrete Time Random Walk with start position equal to zero.[3]

For an easier understanding, consider a walker starting at position $X_0 = 0$ and taking four steps. The possible positions after four steps are $X_4 = -4, -2, 0, 2$, and 4. The probabilities for each position can be calculated as follows:

- $P(X_4 = 4) = \binom{4}{4} \left(\frac{1}{2}\right)^4 = \frac{1}{16}$
- $P(X_4 = 2) = \binom{4}{3} \left(\frac{1}{2}\right)^4 = \frac{4}{16} = \frac{1}{4}$
- $P(X_4 = 0) = \binom{4}{2} \left(\frac{1}{2}\right)^4 = \frac{6}{16} = \frac{3}{8}$

- $P(X_4 = -2) = \binom{4}{1}\left(\frac{1}{2}\right)^4 = \frac{4}{16} = \frac{1}{4}$
- $P(X_4 = -4) = \binom{4}{0}\left(\frac{1}{2}\right)^4 = \frac{1}{16}$

These probabilities give us a discrete probability distribution for the position of the walker after four steps. For large t, this distribution approaches a normal distribution with mean 0 and variance t.

t \ n	-5	-4	-3	-2	-1	0	1	2	3	4	5
0						1					
1					$\frac{1}{2}$		$\frac{1}{2}$				
2				$\frac{1}{4}$		$\frac{1}{2}$		$\frac{1}{4}$			
3			$\frac{1}{8}$		$\frac{3}{8}$		$\frac{3}{8}$		$\frac{1}{8}$		
4		$\frac{1}{16}$		$\frac{1}{4}$		$\frac{3}{8}$		$\frac{1}{4}$		$\frac{1}{16}$	
5	$\frac{1}{32}$		$\frac{5}{32}$		$\frac{5}{16}$		$\frac{5}{16}$		$\frac{5}{32}$		$\frac{1}{32}$

Figure 13: Intuition on the probability distribution of a one dimensional discrete time random walk with start position equal to zero.[3]

6.2 Quantum Walks

Quantum walks exist in both discrete-time and continuous-time variants. Here, we focus on the discrete-time quantum walk, starting with the one-dimensional case and then extending to multi-dimensional quantum walks.

In a one-dimensional quantum walk, a "walker", represented by a quantum state, moves along a line of discrete positions based on a quantum "coin" flip. This quantum coin introduces a superposition of states, allowing the walker quantum state to represent multiple positions simultaneously, unlike in a classical random walk.

The state of the walker in a one-dimensional quantum walk consists of:

- Position state $|x\rangle$: Represents the position of the walker along the line, where x is an integer.
- Coin state $|c\rangle$: Represents the direction of the walk, with $|0\rangle$ indicating a move to the left and $|1\rangle$ indicating a move to the right.

The combined state of the walker at a given time t is represented as a superposition of coin and position states:

$$|\psi(t)\rangle = \sum_x \left(\alpha_x(t)\,|0\rangle \otimes |x\rangle + \beta_x(t)\,|1\rangle \otimes |x\rangle\right)$$

where $\alpha_x(t)$ and $\beta_x(t)$ are complex amplitudes associated with each position and coin state.

On the evolution of a quantum walk, two main operators govern the evolution of the quantum walk:

- Coin operator (C): The coin operator acts on the coin state, creating a superposition of directions. A common choice for this operator is the Hadamard operator H:

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

  When applied to the coin state, H creates an equal superposition of moving left and right.

- Shift operator (S): The shift operator acts on both the coin and position states, moving the walker to the left or right based on the coin state. The shift operator S is defined as:

$$S(|0\rangle \otimes |x\rangle) = |0\rangle \otimes |x-1\rangle\,, \quad S(|1\rangle \otimes |x\rangle) = |1\rangle \otimes |x+1\rangle$$

This means that a $|0\rangle$ on the coin space indicates that the "walker" should move decrease his position. Similarly, a $|1\rangle$ on the coin space makes the "walker" position increase by one after a shift operation. Because of that, we can think that each combined quantum state (coin and position) represents an arc on a directed graph where the nodes are the possible positions and the arcs represent the allowed directions. Figure 14 shows that ideia.

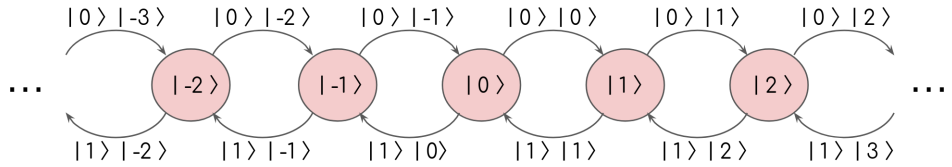


Figure 14: One dimensional discrete time quantum walk representation.

The evolution of the quantum walk over one time step is given by:

$$|\psi(t+1)\rangle = S \cdot (C \otimes I)\,|\psi(t)\rangle$$

where I is the identity operator on the position space. The identity operator is the one which does not change the state.

Consider an example of a one dimensional discrete time quantum walk where the walker starts at position $x = 0$ with the initial state:

$$|\psi(0)\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$$

After applying the Hadamard coin operator and the shift operator for a single time step, the state evolves as:

$$|\psi(1)\rangle = \frac{1}{2}\left(|0\rangle \otimes |-1\rangle + |1\rangle \otimes |1\rangle\right)$$

At $t = 2$, another application of the coin and shift operators results in a superposition state with the walker having non-zero amplitudes at multiple positions, illustrating the parallel nature of quantum walks.

To determine the position of the walker in a quantum walk, we rely on the quantum mechanical principle of measurement. The quantum state of the walker after $t = 2$ time steps is a superposition of multiple position states, each with a complex amplitude. These amplitudes encode the probabilities of finding the walker at a specific position when measured.

As stated before, after t steps, the quantum state of the walker is represented as:

$$|\psi(t)\rangle = \sum_x \left(\alpha_x(t)\,|0\rangle \otimes |x\rangle + \beta_x(t)\,|1\rangle \otimes |x\rangle\right)$$

- $|x\rangle$ represents the position of the walker.
- $\alpha_x(t)$ and $\beta_x(t)$ are the complex amplitudes corresponding to the two coin states $|0\rangle$ (move left) and $|1\rangle$ (move right) at position x.

To find the walker's position, we measure the state $|\psi(t)\rangle$. The probability $P(x)$ of finding the walker at position x is given by the sum of the squared magnitudes of the amplitudes associated with both coin states at that position:

$$P(x) = |\alpha_x(t)|^2 + |\beta_x(t)|^2$$

This ensures that the probability of finding the walker at any position is determined by the total amplitude contribution from both possible directions.

For instance, if the state of the walker at $t = 2$ is:

$$|\psi(2)\rangle = \frac{1}{2}\,|0\rangle \otimes |-2\rangle + \frac{\sqrt{2}}{4}\,|0\rangle \otimes |0\rangle + \frac{\sqrt{2}}{4}\,|1\rangle \otimes |0\rangle + \frac{1}{2}\,|1\rangle \otimes |2\rangle$$

then the probabilities of finding the walker at each position are:

$$P(-2) = \left|\frac{1}{2}\right|^2 = \frac{1}{4}, \quad P(0) = \left|\frac{\sqrt{2}}{4}\right|^2 + \left|\frac{\sqrt{2}}{4}\right|^2 = \frac{1}{8} + \frac{1}{8} = \frac{1}{4}, \quad P(2) = \left|\frac{1}{2}\right|^2 = \frac{1}{4}$$

The sum of probabilities over all positions equals 1, ensuring consistency with the probabilistic interpretation of quantum mechanics.

When the quantum state is measured, the walker collapses to a specific position x with a probability $P(x)$. This collapse is non-deterministic, meaning the exact position cannot be predicted beforehand, but the probabilities reflect the likelihood of each outcome. The interference effects inherent in quantum walks lead to a probability distribution that is typically more spread out and skewed compared to classical random walks, enabling faster exploration of the state space. We usually say that it has a balistic behavior as the probaility of encountering the walker further from the initial position is higher than the probability to find it closer to the initial position.

This measurement process highlights the probabilistic and non-deterministic nature of quantum mechanics, which is central to understanding and utilizing quantum walks in algorithms and simulations.

t \ n	-5	-4	-3	-2	-1	0	1	2	3	4	5
0						1					
1					$\frac{1}{2}$		$\frac{1}{2}$				
2				$\frac{1}{4}$		$\frac{1}{2}$		$\frac{1}{4}$			
3			$\frac{1}{8}$		$\frac{1}{8}$		$\frac{5}{8}$		$\frac{1}{8}$		
4		$\frac{1}{16}$		$\frac{1}{8}$		$\frac{1}{8}$		$\frac{5}{8}$		$\frac{1}{16}$	
5	$\frac{1}{32}$		$\frac{5}{32}$		$\frac{1}{8}$		$\frac{1}{8}$		$\frac{17}{32}$		$\frac{1}{32}$

Figure 15: Initial steps of a one dimensional quantum walk using hadamard coin and initial position $|0\rangle \otimes |0\rangle$.[3]

Moving to multi-dimensional quantum walks, the walker moves in a space with multiple dimensions (e.g., a two-dimensional grid). Each dimension has its own coin state, allowing the walker to explore more complex spaces. These walks are particularly useful in problems involving graphs and spatial structures, such as image processing and quantum search algorithms on networks.

For a two-dimensional quantum walk, the position state is given by $|x, y\rangle$, where x and y represent the coordinates on a grid. The coin state now has four possible outcomes, as it must determine movement in both the x- and y-directions. A typical choice is to use a 4-dimensional coin operator, such as:

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

The shift operator S acts based on the coin state, moving the walker along the grid. For example:

$$S(|0\rangle \otimes |x, y\rangle) = |0\rangle \otimes |x + 1, y\rangle \quad \text{(right)}$$

$$S(|1\rangle \otimes |x, y\rangle) = |1\rangle \otimes |x - 1, y\rangle \quad \text{(left)}$$

$$S(|2\rangle \otimes |x, y\rangle) = |2\rangle \otimes |x, y + 1\rangle \quad \text{(up)}$$

$$S(|3\rangle \otimes |x, y\rangle) = |3\rangle \otimes |x, y - 1\rangle \quad \text{(down)}$$

Consider a walker initially located at $(x, y) = (0, 0)$ with an equal superposition over the four possible coin states:

$$|\psi(0)\rangle = \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle) \otimes |0, 0\rangle$$

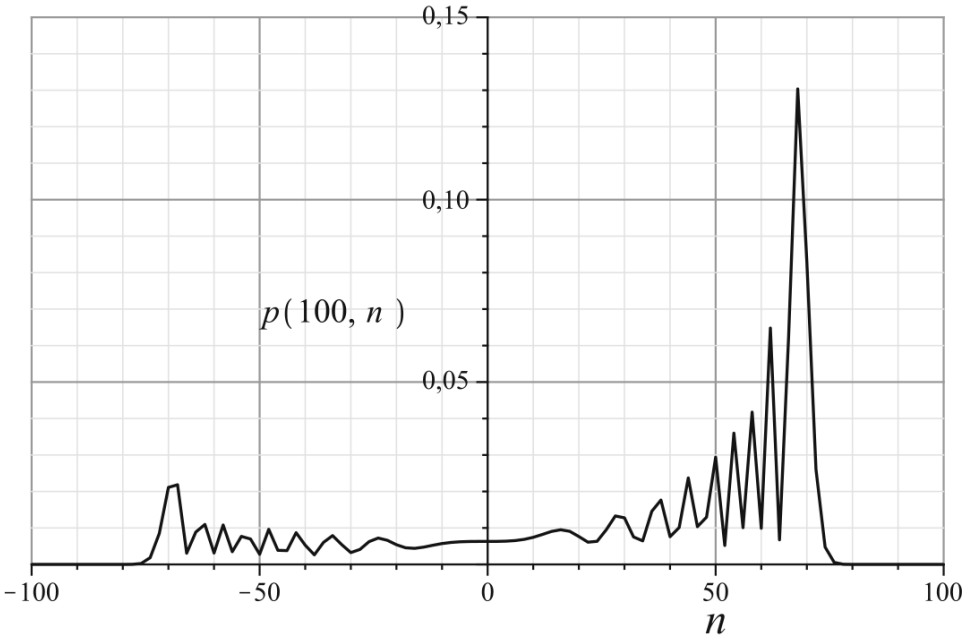


Figure 16: Probability distribution of a one dimensional quantum walk using hadamard coin and initial position $|0\rangle \otimes |0\rangle$.[3]

After applying the coin and shift operators, the state at $t = 1$ evolves to:

$$|\psi(1)\rangle = \frac{1}{2}\left(|0\rangle \otimes |1,0\rangle + |1\rangle \otimes |-1,0\rangle + |2\rangle \otimes |0,1\rangle + |3\rangle \otimes |0,-1\rangle\right)$$

After multiple steps, the walker's state becomes a complex superposition over various positions, and the resulting probability distribution shows multiple peaks due to interference effects, enabling efficient exploration of the grid.

In higher-dimensional quantum walks, the principles remain the same: each dimension has an associated coin state, and the shift operator moves the walker in each direction based on the coin state. These higher-dimensional walks are useful in applications such as quantum search algorithms on complex networks and in simulating quantum dynamics on large graphs.

To conclude, quantum walks have applications in various quantum algorithms and protocols. Some key examples include:

- Quantum Network Protocols: Quantum walks enable information routing and entanglement distribution across quantum networks.
- Graph Algorithms: Quantum walks offer faster solutions to problems on graphs, such as graph traversal and element distinctness.[3]

6.3 Spatial Search Using Quantum Walks

Spatial search using quantum walks is an algorithm designed to find a marked location on a graph [3]. This method utilizes the properties of quantum walks to achieve a quadratic speedup over classical algorithms, similar to Grover's algorithm in unsorted databases. The key idea is to harness the interference and superposition of quantum walks to efficiently locate the target.

Consider a graph $G = (V, E)$, where V is the set of vertices, E is the set of edges, and $|V| = N$. The goal is to find a marked vertex $m \in V$, starting from an initial uniform superposition over all vertices:

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{v \in V} |v\rangle .$$

The spatial search algorithm using quantum walks is structured as follows:

- Quantum Walk Operators: The algorithm relies on two unitary operators: the *coin operator* C and the *shift operator* S. The coin operator acts on the internal degree of freedom (coin space) of the walker, while the shift operator moves the walker on the graph based on the coin state.
- Marking the Target: To focus the quantum walk on the marked vertex m, a phase flip operation is applied to the marked vertex:
$$U_m = I - 2\left|m\right\rangle\left\langle m\right|.$$
This operator inverts the phase of the marked vertex, analogous to the oracle in Grover's search algorithm.
- Evolution operator: The evolution of the quantum walk is governed by a combined operator:
$$U = S \cdot (C \otimes I).$$
After applying U_m, the state evolves iteratively under the influence of U.
- Amplitude Amplification: As the quantum walk progresses, the amplitude of the marked vertex increases due to constructive interference, while the amplitude of other vertices decreases due to destructive interference.
- Measurement: After $O(\sqrt{N})$ iterations, the quantum state is highly concentrated on the marked vertex. A measurement in the computational basis collapses the state, revealing the marked vertex with high probability.

The spatial search algorithm benefits from the ballistic behavior of quantum walks, where the probability distribution is more spread out compared to classical random walks. This property enables faster exploration of the graph and more efficient localization of the marked vertex.

Additionally, the success probability depends on the structure of the graph. For regular graphs, such as hypercubes and complete graphs, the algorithm performs optimally. For irregular graphs, the performance can vary based on connectivity and symmetry.

Spatial search algorithms using quantum walks have broad applications, including:

- Searching for specific nodes in a network or database.
- Solving graph traversal problems in quantum computing.
- Applications in quantum simulation and optimization problems.

7 Distributed Quantum Walk Control Plane for Quantum Networks

The primary outcome of this project is the research paper included at the end of this document [9]. To provide a better understanding and context for the paper, I will introduce its main ideas here.

This project focuses on exploring quantum walks within the framework of quantum networks. Previous studies have demonstrated that quantum walks can be employed to control distributed quantum operations across network nodes, provided these nodes have access to ideal quantum gates and memories [7]. Our work builds upon this foundation by extending the initial formulation to include a more practical and physical implementation of the control plane [8].

What follows is an introduction to the protocol outlined in the paper, as well as an explanation of the contributions we made to complement the original formulation.

7.1 Quantum Walks Utility in this scenario

The protocol employs a quantum walk as a quantum control signal to execute distributed quantum operations. We generalize the discrete-time coined quantum walk model, which incorporates the interaction between a quantum walker system in the network graph with quantum registers inside the network nodes. The protocol abstracts hardware implementation and the transmission of quantum information through channels. The propagation of the walker system across the network represents the transmission of the control signal, while the application of coin operators integrates interactions between the control layer and the quantum registers. [7].

7.2 Mathematical Background

Consider a directed graph $G = (V, E)$ obtained from a non-directed graph by introducing two directed edges for each initial one. The sets $N^+(v) \subseteq V and N^-(v) \subseteq V$ denote the sets of outward and inward neighbors of v, respectively.

A complex vector in a Hilbert space $Hw \subseteq Hv \otimes Hc$ evolves through a discrete-time coined quantum walk on graph G, defined by the graph structure. The vertex space Hv it is used to codify the vertices of the graph by assigning to each vertex a basis vector of Hv, which has dimension $|V|$. The coin space Hc denotes the degrees of freedom of the walker(possible movements for the walker) and has a dimension given by the maximum degree of the graph $D = max d(v) : v \in V$. The basis for Hw is $|v, c\rangle : v \in V, c \in Cv$, where $Cv = 0, ..., d(v) - 1$ is the integer set for the number of outward edges of a node v.

The basis for the spaces Hc and Hv are denoted as $|c\rangle$ and $|v\rangle$, respectively. Assuming $|\Psi(t)\rangle$ is the walker wavefunction at discrete time instant t, the evolution of the quantum walk is given by the action of two unitary operators $S : Hw \to Hw$ and $W : Hw \to Hw$ on the system state vector

$$|\Psi(t+1)\rangle = SW\,|\Psi(t)\rangle$$

[6]

7.3 Coin and Shift operators

The walker's degrees of freedom are affected by the coin operator (W). The general form of the coin operator is represented by:

$$W = \sum_{v \in V} |v\rangle \langle v| \otimes Wv$$

where Wv is a unitary operator. The coin operator mixes the amplitude of a particular state $|v, c\rangle$ with states $|v, c'\rangle$ such that $c, c' \in Cv$, which are the degrees of freedom of the same vertex. [5]

The shift operator, also known as the swap operator (S), is responsible for moving the mixed amplitudes generated by the coin operator W to outward edges. To define the operator, we introduce the function$\eta : V \times C \to V$, which maps outward edges to outward neighbors of each vertex. Thus, for a vertex v and an index $c, \eta(v, c)$ denotes the c-th outward neighbor of v, which we denote by u.

Additionally, we introduce $\sigma : V \times V \to C$, which maps an inward edge of a vertex to one of its outward edges.

The action of the shift operator is then formally defined as follows: for a state vector $|u, c\rangle$, the shift operator transforms it into $|\eta(u, c), \sigma(u, \eta(u, c))\rangle$. Note that η and σ can be defined arbitrarily as long as the operator remains unitary and respects the graph edges. In fact, different choices of η and σ can lead to different dynamics of state amplitudes. [5]

7.4 Paper

Back to our work, the research goal was defined as implementing the quantum walk protocol described on [7]. By implementation, we mean a move from the logical approach to a more physical approach, including encoding considering the actual physical qubits and showing how we would perform the operations considering the non-locality. Also, one can find a presentation about the quantum walk control plane implementation at https://interscity.org/quantum-internet/.

The paper [9] published and presented at the First Workshop on Quantum Networks, part of the 2024 "Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos", event organized by the Brazilian Computer Society, explores two different implementations of the quantum walk control plane for quantum networks: the linear approach and the logarithmic approach. Both methods are analyzed for their encoding schemes, operational characteristics, and trade-offs in resource usage.

7.5 Linear Approach

The linear approach leverages a straightforward encoding strategy, associating each edge in the network graph with a single qubit. This approach prioritizes simplicity in qubit representation and make the operations easier.

In the linear approach:

- Each edge (u, v) in the directed graph is encoded using one qubit.
- If the qubit associated with an edge (u, v) is in the state $|1\rangle$, it indicates that the quantum walker is on that edge.
- The total number of encoding qubits required is proportional to $O(|V| \cdot d_{\max})$, where $|V|$ is the number of vertices and $d_{\max}$ is the maximum degree of the graph.

The linear approach employs:

- Local coin operator: Operates within a single node to mix the amplitudes associated with the outgoing edges.
- Non-Local shift operator: Performs the movement of the walker across edges. This is implemented using gate teleportation, which requires two EPR pairs per edge to perform a swap operation remotely. The circuit used to do the shift operation is on https://algassert.com/post/1717 and it is optimal [1].

The linear approach offers simple and intuitive encoding, making it easier to implement and understand. Additionally, it requires fewer EPR pairs for operations compared to the logarithmic approach, which can simplify the resource requirements in certain scenarios.

Despite its simplicity, the linear approach has a high qubit overhead for encoding, as each edge in the graph requires a dedicated qubit. This makes the approach less efficient when the number of edges grow significantly, leading to scalability issues.

7.6 Logarithmic Approach

The logarithmic approach offers a more compact encoding by reducing the number of qubits required per node. This is achieved by leveraging a logarithmic representation of the graph's degrees.

In the logarithmic approach:

- Each node uses $\lceil \log_2(d(v) + 1) \rceil$ qubits, where $d(v)$ is the degree of the vertex v.

- The encoding of an edge (u, v) is the same as the encoding of (v, u), enabling bidirectional representation.
- The total number of encoding qubits required is proportional to $O(|V| \cdot \log_2(d_{\max}))$.

The logarithmic approach utilizes:

- Local coin operator: Similar to the linear approach, operates within the node to mix amplitudes.
- Non-Local shift operator: A more complex implementation than in the linear approach, involving complex unitary transformations. In the worst case, teleportation techniques can be used to perform the shift operation.

The logarithmic approach significantly reduces the number of encoding qubits. It provides an efficient representation of graph degrees, which optimizes the use of quantum resources.

However, this approach requires more complex operations for the shift, often involving additional EPR pairs. Moreover, it introduces higher implementation complexity for non-local operations, which can pose challenges during practical deployment.

7.7 Comparison of Approaches

The linear and logarithmic approaches represent two different strategies for implementing the quantum walk control plane. The linear approach is straightforward but requires more qubits, while the logarithmic approach is more efficient in encoding qubit usage but introduces operational complexities.

Modeling details	**Linear**	**Logarithmic**
required encoding qubits	$O(\|V\|d_{max})$	$O(\|V\|lg(d_{max}))$
coin operation	trivial	trivial
shift operation	remote swap	selected swap/non trivial
encoding qubits per shift	one	$lg(d_{max})$
operation qubits per shift	two	$O(lg(d_{max}))^*$
one node shift	parallel	sequential or parallel(time is negligible)

Table 1: Comparison of Linear and Logarithmic Approaches.

7.8 Trade-Off Between Qubits, Time, and Circuits

Both the linear and logarithmic approaches support parallel shifts when EPR pairs are available, with the time to execute shifts being negligible. However, the logarithmic approach requires more EPR pairs per shift, likely proportional to the number of qubits used in the encoding. The total number of EPR pairs needed for a global shift depends on the number of edges and the resources required per shift. In the linear approach, two EPR pairs are needed per shift, whereas in the logarithmic approach, the exact number is not known but is expected to scale logarithmically.

Performing parallel shifts demands more qubits for simultaneous EPR pairs, while sequential shifts introduce time delays as operations must wait for an operation to end to start attempting to establish the others EPR pairs.

In more details, each edge requires a shift operation, creating a trade-off between the number of Bell pairs and the time needed. Performing all shifts simultaneously requires $O(E \cdot b(e))$ EPR pairs, where $b(e)$ is the number of Bell pairs needed per shift. Reducing qubit usage necessitates sequential shifts, increasing time due to repeated Bell pair establishment. For example, in a 2D lattice, the minimum shift qubits per node equals $b(e)$, and $O(n)$ rounds of shifts are required for an $n \times n$ lattice.

Given that, we divided the system into two groups: encoding qubits and operation qubits (EPR pairs). Three configurations van be analyzed:

1. Linear encoding with parallel shifts.
2. Logarithmic encoding with sequential shifts.
3. Logarithmic encoding with parallel shifts.

These configurations illustrate the trade-offs between resource usage and operational efficiency, guiding the choice of approach based on network size and constraints.

8 Conclusion

This section explores the potential future work stemming from the research on the distributed quantum walk control plane and conclude this thesis. By extending the current approaches, we aim to further refine and evaluate their practical implications in quantum networks.

One of the most promising avenues for future work is the simulation of real quantum networks using both the linear and logarithmic approaches. Such simulations can provide practical insights into their performance, including:

- Resource consumption, such as qubits and EPR pairs.
- Time efficiency for operations like coin and shift.
- Scalability and adaptability to varying network topologies.

To achieve this, we propose leveraging tools like Quantum Savory, a quantum network simulator developed by the University of Massachusetts Amherst. Simulating realistic scenarios will allow for a comprehensive comparison between the two approaches, highlighting their strengths and weaknesses in practical environments.

Another exciting direction is integrating ideas from Grover's spatial search into the quantum walk control plane. Grover's algorithm provides a quadratic speedup for searching unstructured databases, and its spatial extension could enhance the efficiency of quantum walk protocols. We thought about using a similar idea to explore routing options.

Also, further exploration of alternative encoding schemes could lead to more efficient implementations. For instance:

- Hybrid encoding methods combining features of linear and logarithmic approaches.
- Dynamic encoding strategies that adapt to changing network conditions.
- Encodings that minimize qubit usage while maintaining high fidelity and speed.

This work highlights the development of a distributed quantum walk control plane for quantum networks, showcasing two implementation strategies—linear and logarithmic approaches. Each method presents unique trade-offs between encoding complexity, resource consumption, and operational efficiency. Through this research, we have advanced the understanding of how quantum walks can serve as robust control signals in quantum network architectures.

We would like to express our gratitude to Professor Don Towsley and Matheus Guedes from the University of Massachusetts Amherst for their invaluable partnership, which significantly enriched this research. Special thanks are also due to my supervisor, Professor Fabio Kon (IME/USP), and my co-supervisor, Professor Antônio Jorge Gomes Abelém (UFPA), for their guidance and support throughout the project. Lastly, we acknowledge the financial support provided by FAPESP, which made this research possible.

References

[1] Daniel Collins, Noah Linden, and Sandu Popescu. "Nonlocal content of quantum operations". In: *Phys. Rev. A* (2001).

[2] C. I. Nielsen Michael A. *Quantum computation and quantum information: 10th Anniversary Edition.* New York: Cambridge University Press, 2010.

[3] R Portugal. *Quantum Walks, and Search Algorithms.* Switzerland: Springer Nature, 2018.

[4] Wehner Stephanie, Elkouss David, and Hanson Ronald. "Quantum internet: A vision for the road ahead". In: *Science* 362.6412 (2018).

[5] Matheus G Andrade, Franklin de Lima Marquezino, and Daniel R Figueiredo. "On the equivalence between quantum and random walks on finite graphs". In: *Quantum Information Processing* 19.11 (2020), p. 417.

[6] Matheus Guedes de Andrade, Franklin de Lima Marquezino, and Daniel Ratton Figueiredo. "Characterizing the Relationship Between Unitary Quantum Walks and Non-Homogeneous Random Walks". In: *Concurso de Teses e Dissertações (CTD).* SBC. 2021, pp. 43–48.

[7] Matheus Guedes de Andrade et al. "A quantum walk control plane for distributed quantum computing in quantum networks". In: *2021 IEEE International Conference on Quantum Computing and Engineering (QCE).* IEEE. 2021, pp. 313–323.

[8] Matheus Guedes de Andrade et al. "Universal Quantum Walk Control Plane for Quantum Networks". In: *arXiv preprint arXiv:2307.06492* (2023).

[9] André Ribeiro et al. "Distributed Quantum Walk Control Plane Implementation". In: *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos.* SBC. 2024.

Distributed Quantum Walk Control Plane Implementation

André Ribeiro[1], Matheus Guedes de Andrade[2], Fabio Kon[1], and Don Towsley[2]

[1]Departamento de Ciência da Computação
Universidade de São Paulo (USP)

[2]Manning College of Information and Computer Sciences
University of Massachusetts Amherst

Abstract. *The Quantum Walk Control Protocol (QWCP) enables universal distributed quantum computing in quantum networks. In its debut, QWCP was specified in terms of logical quantum operations required by a quantum network to implement the protocol. In this paper, we propose two possible implementations of the QWCP based on different ways to encode quantum walks in physical qubits. The proposed encodings require different numbers of qubits and remote entangled states to perform control operations. Our approaches help to bridge the gap between the logical description of the QCWP and possible physical realizations of the protocol.* [1]

1. Introduction

Quantum networks are on the frontier of quantum research. The field combines various disciplines with the promise of enabling quantum communication among quantum processors, leading to applications that cannot be achieved with classical communication alone [Kimble 2008]. Similar to traditional computer networks, quantum networks will require efficient communication protocols to enable scalable quantum communication. In addition to orchestrating network control operations, quantum network protocols must be tailored to overcome the inherent fragility of quantum information. Qubits are prone to errors from a variety of sources, such as memory decoherence in storage and imperfect state transduction in photonic transmission.

In this context, distributed quantum computing (DQC) is a fundamental application enabled by quantum networks [Buhrman and Röhrig 2003]. It requires control protocols to orchestrate distributed quantum operations despite the choice of quantum hardware used in the network fabric. Previous work proposed the Quantum Walk Control Protocol (QWCP) [de Andrade et al. 2023] to enable universal DQC in quantum networks. The QWCP was described in terms of logical operators and can be used to control arbitrary quantum network operations, e.g., entanglement distribution [Pant et al. 2019]. The logical description of QWCP is general, although it does not directly address how the protocol is implemented in terms of qubits.

In this paper, we address the problem of implementing the QWCP in quantum processors. We provide two ways one can map the logical description of the QWCP to

[1]This research was supported in part by the NSF grant CNS-1955744, NSF- ERC Center for Quantum Networks grant EEC-1941583, INCT of the Future Internet for Smart Cities CNPq proc. 465446/2014-0, and FAPESP procs. 14/50937-1 and 2020/04031-1.

qubits in the network nodes: the linear encoding and the logarithmic encoding implementation. We describe the number of qubits required at each node in order to implement the QWCP, together with a description of the control operators for both proposed methods. Finally, we present an initial theoretical analysis on the performance of the proposed implementations, which shows a trade-off between network resources used for control and the time required to complete control operations.

2. System Model

We follow the system model described in [de Andrade et al. 2023] for a quantum network. Consider a symmetric directed graph $G = (V, A)$, with V and A representing the nodes and arcs of the graph, respectively. Since G is symmetric, $(v, u) \in A$ if, and only if, $(u, v) \in A$. We use $\delta(v)$ to denote the set of neighbors of vertex $v \in V$ and $d(v) = |\delta(v)|$ to denote v's degree.

Edge colorings are of particular interest to the description and analysis of the methods we propose in this article. Let $G' = (V, E)$ denote the undirected version of the network graph G where arcs (u, v) and (v, u) of G are represented by the same edge in G'. Let $\Gamma'_{G,k} : E \to \{1, \ldots, |V|\}^{|E|}$ denote an edge coloring of G' where at most k edges incident to the same node have the same color. Let $\Gamma_{G,k} : A \to \{1, \ldots, |V|\}$ be a coloring of the directed graph G obtained from $\Gamma'_{G,k}$ as $\Gamma_{G,k}(v, u) = \Gamma_{G,k}(u, v) = \Gamma'_{G,k}(v, u)$. Let $\Delta_{\Gamma_{G,k}}$ denote the number of colors in $\Gamma_{G,k}$. When $k = 1$, coloring $\Gamma_{G,k}$ reduces to the usual coloring where edges incident to same node have different colors. Finding the minimum number of colors for G when $k = 1$, i.e., $\min_{\Gamma_{G,1}} \Delta_{\Gamma_{G,1}}$, is an NP-hard problem, although Vizings's theorem states that $d_{\max} \leq \min_{\Gamma_{G,1}} \Delta_{\Gamma_{G,1}} \leq d_{\max} + 1$ [Stiebitz et al. 2012]. For simplicity, we omit the dependency of a coloring with G from now on, and use Γ to denote $\Gamma_{G,1}$.

A quantum network is a set of quantum processors (nodes) that can communicate with each other via quantum channels (arcs). Each node has a fixed number of qubits and can perform local quantum operations on them. We divide those qubits into two disjoint sets, N_v and M_v, which we refer to as network qubits and data qubits, respectively. In this work, we mainly focus on network qubits—which are associated with control operations—and omit data qubits to simplify exposition when possible.

The discrete-time coined quantum walk model is a unitary evolution process defined in the Hilbert space $\mathcal{H}_G = \mathcal{H}_V \otimes \mathcal{H}_C$ that encodes the graph's arcs [Portugal 2018]. In particular, $\mathcal{H}_V$ encodes the vertices and $\mathcal{H}_C$ encodes the coin space of the walker. Specifically, for each $(v, u) \in A$, a basis vector $|v, c_{vu}\rangle$ is defined within $\mathcal{H}_G$, with c_{vu} symbolizing the degree of freedom corresponding to the arc that connects v with u. The evolution of the quantum walk is given by $|\Psi(t+1)\rangle = S(t)C(t)\,|\Psi(t)\rangle$, where C and S are the coin and shift operators, respectively. A generic coin operation on node v maps $\sum_{u\in\delta(v)} \alpha_{vu}\,|v, c_{vu}\rangle \to \sum_{u\in\delta(v)} \beta_{vu}\,|v, c_{vu}\rangle$, where $\alpha_{vu} \in \mathbb{C}$ and $\beta_{vu} \in \mathbb{C}$ respect unitary evolution. C mixes the amplitudes associated with the states representing the outward arcs of a node and shapes the probability that the walk moves from one node to another. The shift operator S maps $|v, c_{vu}\rangle \to |u, c_{uv}\rangle$. It changes amplitudes among different nodes and is related to the propagation of the quantum walk itself, i.e., S drives the movement of the quantum walk among network neighbors.

The QWCP utilizes a quantum walk system to propagate entanglement between

network nodes enabling remotely controlled quantum operations. A diverse set of operations can be performed with the QWCP and we focus on the case where the protocol is used to route quantum control information through a network path $P = \{v_0, v_1, \ldots, v_n\}$. In a nutshell, the QWCP starts in the state $|v_0, c_{v_0}\rangle \otimes (\alpha |0\rangle + \beta |1\rangle)$, where the second term denotes the state of a data qubit in B. The first coin operator coin operator $C(0)$ prepares the state $(\alpha |v_0, c_{v_0}\rangle |0\rangle + \beta |A, c_{v_0 v_1}\rangle |1\rangle)$, which is an entangled state between control and data qubits in v_0. It progresses by evolving the control state through successive applications of coin and shift operators until the quantum walk state has the form $\alpha |v_0, c_{v_0}\rangle + \beta |v_n, c_{v_n}\rangle$. The coin operators applied in the quantum walk evolution after initialization are permutations of the degrees of freedom such that, for $t > 0$, $C(t) : |v_t, c_{v_t v_{t-1}}\rangle \rightarrow |v_t, c_{v_t v_{t+1}}\rangle$. Note that the coin operators that drive the evolution of the quantum walk are local operations while shifts are non-local. For simplicity, we focus on the description of $C(t)$ for $t > 0$, although one can extend the results described in this paper for the data-control operations described in [de Andrade et al. 2023].

3. Quantum Walk Implementation

We present two ways one can implement the quantum walk protocol with networking qubits. Our models add to previous literature by mapping the mathematical definition of the protocol to operations that can be performed on qubits within a quantum processor. In essence, one can think that the description presented in [de Andrade et al. 2023] is a logical description of a quantum algorithm and our models provide two implementations of the algorithm.

3.1. Linear Encoding

We start by assuming that one can use $O(d_{\text{max}})$ qubits in a node to encode the quantum walk, where d_{max} is the maximum node degree. This assumption allows each arc of the network graph G to be represented by one physical qubit in the network. The one-to-one mapping between arcs and qubits leads to a one-hot encoding of the quantum walk: each basis vector $|v, c_{vu}\rangle$ is represented by the quantum state where the qubit associated with arc (v, u) is in state $|1\rangle$ and all other qubits are in state $|0\rangle$. Each superposition of a set of arcs $A' \subset A$ in the one-hot encoding is an entangled state of the qubits representing the arcs in A' resembling a multi-qubit W-state [Dür et al. 2000]. We refer to the state of the qubit associated with arc (v, u) as $|\psi\rangle_{vu}$. Moreover, we refer to this one-hot encoding scheme as the *linear encoding*.

We now describe the action of the coin and shift operators in the linear encoding case. Let q_{vu} denote the qubit in node v that encodes arc (v, u). Let $|\psi\rangle = |1\rangle_{vw} \bigotimes_{a \in A \setminus \{vw\}} |0\rangle_a$ denote the linear encoding of an arbitrary basis state. For a generic coin operator, $C |\psi\rangle = \sum_{u \in \delta(v)} \alpha_{vu} |1\rangle_{vu} \bigotimes_{a \in A \setminus \{vu\}} |0\rangle_a$, where $\alpha_{vu} \in \mathbb{C}$ denotes the amplitude of the basis state associated with (v, u) in $|\psi\rangle$. When $t > 0$, $C(t)$ is a permutation of the states encoding the outgoing arcs of a node, implemented in the linear encoding as $C(t) = \text{SWAP}(q_{v(v-1)}, q_{v(v+1)})$.[2] The shift operator is also a swap gate in the linear encoding, although it involves qubits in different nodes. We implement S using the remote swap gate depicted in Figure 1, which requires the expenditure of two Bell pairs. Using

[2]Note that coins performing arbitrary permutations of the outgoing arcs of a node can be implemented in the linear encoding with a sequence of swap gates.

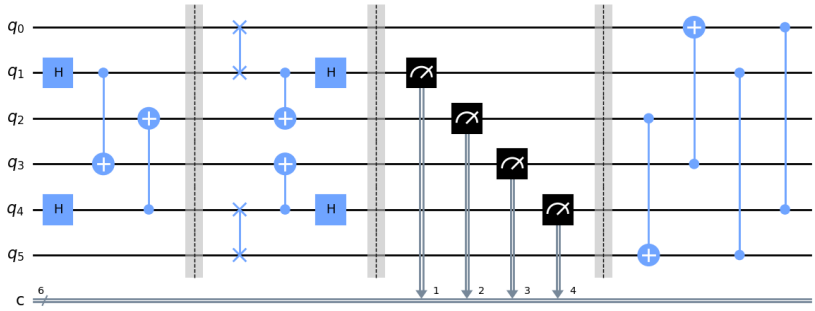
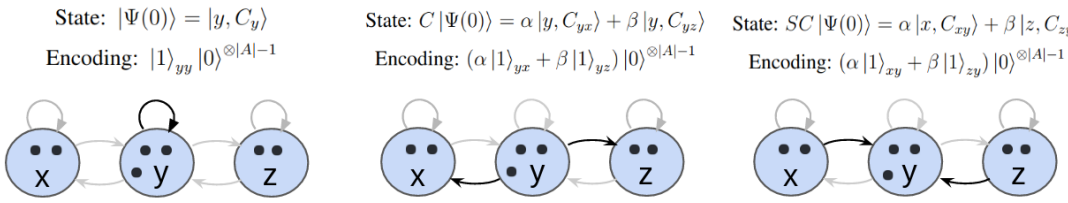


Figure 1. Swap circuit between q_0 and q_5, using two Bell pairs (q_1,q_3) and (q_2,q_4).



(a) Initial state. **(b) Coin operation.** **(c) Shift operation.**

Figure 2. Linear encoding system evolution with logical and code states.

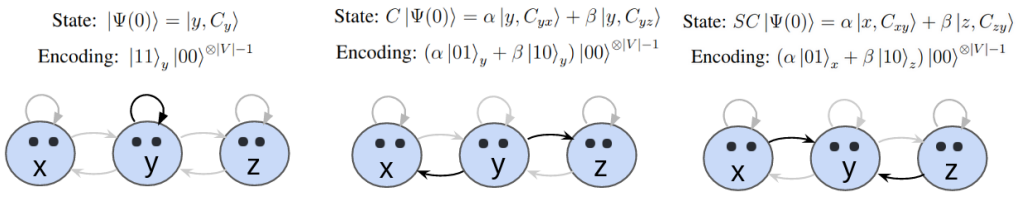
two Bell pairs to execute the remote swap gate is optimal, since this operation cannot be performed using a single Bell pair [Collins et al. 2001].

We illustrate the linear encoding through a three-node network example depicted in Figure 2. In this setup, three qubits in y encode the walker's state, while two encoding qubits in both x and z are utilized. Figure 2a shows the initial logical state when the quantum walk is initialized on the self-loop (y, y), together with the corresponding linear encoding. A coin operator that maps $|y, C_y\rangle$ into the generic superposition $\alpha |y, C_{yx}\rangle + \beta |x, C_{yz}\rangle$, with $\alpha, \beta \in \mathbb{C}$, is applied, generating the states depicted in Figure 2b. Finally, the shift operator propagates the quantum walk to nodes x and z, yielding the states depicted in Figure 2c.

3.2. Logarithmic Encoding

The linear encoding does not utilize the entire power of qubits to encode the quantum walk. We now present an encoding scheme that only requires $\lceil \log_2(\Delta_\Gamma + 1) \rceil$ qubits in each network node. unlike the linear encoding scheme, we do not provide a complete description of the operations for this encoding in terms of quantum gates. Instead, we describe them in terms of multi-qubit unitaries that can be implemented with any universal gate set.

Let $K = \lceil \log_2(\Delta_\Gamma + 1) \rceil$ and $|\psi\rangle_v$ denote the state of all K qubits in node v. The absence of the quantum walker in v is codified by the state $|\psi\rangle_v = |0 \ldots 0\rangle$. States are encoded following the coloring Γ such that $|v, c_{vu}\rangle$ is represented by the state $|\Psi\rangle_v = |\Gamma(v, u)\rangle_v$. Since arcs (u, v) and (v, u) have the same color, the state of the qubits in u encoding (u, v) is the same as the state of qubits in v encoding (v, u). Thus, a generic superposition $(\alpha |v, C_{vu}\rangle + \beta |w, c_{wl}\rangle)$ is encoded as the entangled state $(\alpha |\Gamma(v, u)\rangle_v |0 \ldots 0\rangle_w + \beta |0 \ldots 0\rangle_v |\Gamma(w, l)\rangle_w)$, for $(v, u), (w, l) \in A$.



(a) Initial state. **(b) Coin operation.** **(c) Shift operation.**

Figure 3. Logarithmic encoding system evolution with logical and code states.

A generic coin operator in v drives an evolution of the form as $\sum_{j=0}^{K-1} \alpha_j |j\rangle_v = \sum_{j=0}^{K-1} \beta_j |j\rangle_v$, where $\{\alpha_k\}$ and $\{\beta_k\}$ respect unitarity conditions. The coin operator driving the propagation of the quantum walk over path P is such that $C(t) : |\Gamma(v_t, v_{t-1})\rangle \rightarrow |\Gamma(v_t, v_{t+1})\rangle$. This operation can be performed through a sequence of swaps, where qubits q_k and q_j in v_t are swapped if the k-th bit of $\Gamma(v_t, v_{t-1})$ and j-th bit of $\Gamma(v_t, v_{t+1})$ differ. The shift operator maps state $|\Gamma(v,u)\rangle_v |0\ldots0\rangle_u$ to state $|0\ldots0\rangle_u |\Gamma(v,u)\rangle_v$. This encoding makes the shift operation considerably more complex than the linear encoding case. In the general case, the unitary that implements the shift between two nodes u and v acts on all qubits in both u and v. Furthermore, the most efficient way to implement S in terms of number of Bell pairs consumed that performs the shift operation in the logarithmic approach is not known. Nonetheless, a feasible way to implement the operator is to teleport all qubits from v to u, perform the unitary locally, and teleport the qubits back to v. This approach requires $2\lceil\log(\Delta_\Gamma) + 1\rceil$ Bell pairs. As stated in Section 2, finding an optimal coloring for the encoding is an NP-Hard problem, although any edge coloring can be used to generate an encoding for the quantum walk.

We illustrate the logarithmic encoding in Figure 3 with the same three-node network exampled used for the linear encoding in Figure 2.

4. Performance analysis

Performing coin operations in both encodings described in Section 3 is simple from a networking perspective since they only require local operations in the nodes. Shift operations are more complex, since they require the generation of entanglement between neighboring nodes. We now analyse the performance of the two proposed encodings in terms of the network resources required to implement shift operations. We now assume that network nodes have additional network qubits that can be used to create Bell pairs with their neighbors in order to implement the remote gates required by shift operators. From now on, we divide the network qubits in the nodes into two groups: encoding and operation qubits.

Shift operations in both encodings are defined for a single link and performing the shift operation for the entire network—which we refer to as the network shift throughout the remainder of this paper—requires shifts on each network link. There is trade-off between the number of Bell pairs utilized to perform the network shift and the time required to complete the operation. This trade-off stems from the fact that the number of parallel shifts that can be executed at a given node depends on how many non-local Bell pairs that node can share with its neighbors at a given step.

Let a k-parallel shift denote shifting $k \geq 1$ links on a node simultaneously. Let a

sequential shift denote shifting each link of a node one at a time, i.e., a 1-parallel shift. Since the time to execute a shift operation is negligible if a node has sufficient pre-shared Bell pairs, parallel node shifts are possible in both encodings. Executing a k-parallel shift in a node consumes kB_{enc} Bell pairs simultaneously, where B_{enc} is the number of Bell pairs consumed by a shift operation over a link in a given encoding. Note that $B_{\text{enc}} = 2$ in the linear encoding and $B_{\text{enc}} = 2\lceil \log(\Delta_\Gamma) + 1 \rceil$ in the logarithmic encoding. A sequential shift in node v incurs a time delay on the order of $O(d(v))$, while k-parallel shifts lead to a time delay of $O(d(v)/k)$. We discuss the time required to complete the network shift in terms of *shift rounds*, or simply *rounds*: a round is a maximal set of shift operations performed in different nodes such that no additional shift can be performed without additional Bell states. The number of rounds required to complete the network shift relates to edge coloring problems. In particular, the number of rounds is $\lceil \Delta_{\Gamma_k}/k \rceil$ when nodes perform k-parallel shifts. For instance, the network shift on an n-node star requires $n - 1$ rounds if only sequential shifts are allowed.

5. Conclusion

In this paper, we described two different implementations of a mathematical quantum walk control plane [de Andrade et al. 2023] based on *(i)* a linear encoding and *(ii)* a logarithmic encoding. For both cases, we specified how to arrange the qubits in the network and how to perform the operations needed to implement the quantum walk control plane for universal distributed quantum computation. Finally, we analyzed and compared the proposed encoding schemes in terms of the trade-off between network resources and operation time. As future work, we plan to conduct simulations of the control plane. Our goal is to further investigate the performance of the two models proposed when effects such as classical communication latency and memory decoherence are taken into account.

References

Buhrman, H. and Röhrig, H. (2003). Distributed quantum computing. In *International Symposium on Mathematical Foundations of Computer Science*, pages 1–20. Springer.

Collins, D., Linden, N., and Popescu, S. (2001). Nonlocal content of quantum operations. *Phys. Rev. A*, 64:032302.

de Andrade, M. G., Panigrahy, N. K., Dai, W., Guha, S., and Towsley, D. (2023). Universal quantum walk control plane for quantum networks. *arXiv preprint arXiv:2307.06492*.

Dür, W., Vidal, G., and Cirac, J. I. (2000). Three qubits can be entangled in two inequivalent ways. *Physical Review A*, 62(6):062314.

Kimble, H. J. (2008). The quantum internet. *Nature*, 453(7198):1023–1030.

Pant, M., Krovi, H., Towsley, D., Tassiulas, L., Jiang, L., Basu, P., Englund, D., and Guha, S. (2019). Routing entanglement in the quantum internet. *npj Quantum Information*, 5(1):25.

Portugal, R. (2018). *Quantum Walks, and Search Algorithms*. Springer Nature, Switzerland.

Stiebitz, M., Scheide, D., Toft, B., and Favrholdt, L. M. (2012). *Graph edge coloring: Vizing's theorem and Goldberg's conjecture*, volume 75. John Wiley & Sons.