TÉCNICO LISBOA

Distributing Multipartite Entanglement over Quantum Networks

Luís Pedro Morais Bugalho

Thesis to obtain the Master of Science Degree in

Engineering Physics

Supervisors: Prof. Yasser Rashid Revez Omar
Prof. João Carlos Carvalho de Sá Seixas

Examination Committee

Chairperson: Prof. Pedro Miguel Félix Brogueira
Supervisor: Prof. João Carlos Carvalho de Sá Seixas
Member of the Committee: Prof. Damian Markham

January 2021

"Les eaux se gonflaient et redescendaient lentement. Cette respiration calme de la mer faisait naître et disparaître des reflets huileux à la surface des eaux. Devant eux, la nuit était sans limites."

in *La Peste* by Albert Camus

Acknowledgments

Behind every thesis there is always a number of people that made it possible. I would like to thank my supervisors Professor Yasser Omar and Professor João Seixas for their guidance throughout this thesis, which I would call the first step into my future career, as well as the support from the EU H2020 Quantum Flagship project QIA – Quantum Internet Alliance (820445).

I would also thank the Physics of Information and Quantum Technologies Group, from Instituto de Telecomunicações for much appreciated feedback, specially to Dr. Bruno Coutinho who has provided large amounts patience, time and expertise throughout all the period that culminated in my thesis. Without our discussions and fruitful suggestions for literature, this thesis would not be possible. So it goes, "discussion is an exchange of knowledge".

In my personal circles, I would like to thank all of my close friends with whom many moments were shared, in and out of IST. Specially considering the period in question, they were capable of providing the necessary support and contribute not only for overall happiness, but also to realise myself as an individual capable of perceiving what surrounds us all.

To my family, my deepest thanks for always being there. Thank you for allowing and pushing me to find my path with nothing shorter than love, trust and pride.

Resumo

A internet quântica irá possibilitar novas tecnologias quânticas distribuídas em redes, particularmente através de distribuição de entanglement entre duas partes a longas distâncias. No entanto, algumas das suas possíveis aplicações em áreas como comunicação, detecção de sinais e computação podem beneficiar do entanglement multipartido ser partilhado entre vários nodos da rede. Neste projecto, abordamos o problema de distribuir optimamente este tipo de entanglement em redes quânticas com ruído, onde cada link é um par entangled. Para o fazer, descrevemos o ruído na rede com canais depolarizadores, verificando o seu efeito na distribuição dos estados GHZ. Introduzimos também ferramentas da teoria clássica de routing que são capazes de criar um framework para resolver o problema de otimalidade, permitindo a inclusão de parâmetros adicionais. Apresentamos um algoritmo para distribuição ótima de um estado GHZ de 3 qubits maximizando simultaneamente a fidelidade do estado final e a probabilidade de sucesso, simulando-o para diferentes modelos de redes quânticas. Derivamos também aproximações na forma como a complexidade dos nossos algoritmos aumentam, corroborando os tempos de execução polinomiais das simulações. Além disso, determinamos as condições necessárias para manter esta optimalidade simultânea para estados GHZ com um número maior de qubits e para outros tipos de entanglement multipartido. Este trabalho abre caminho para gerar otimamente correlações quânticas multipartidas em redes quânticas com ruído, um recurso importante para tecnologias quânticas distribuídas.

Palavras-Chave: Redes Quânticas, Routing, Entanglement Multiparte, Distribuição de Entanglement, Estados Quânticos GHZ, Fidelidade

Abstract

The quantum internet will enable quantum networked technologies, namely by distributing bipartite entanglement over large distances. However, some of its possible applications in areas such as communication, sensing and computation may benefit from multipartite entanglement being shared between several nodes. In our work, we address the problem of distributing optimally this type of entanglement over noisy quantum networks, where each link is an entangled pair. To do this, we describe the noise of the network with depolarising channels, verifying its effect on the distribution of GHZ multipartite entangled states. We also introduce tools from classical routing theory that are capable of creating a framework to address the optimality problem, allowing the inclusion of additional parameters. An algorithm for optimal distribution of a 3-qubit GHZ state maximising simultaneously the fidelity of the final state and the probability of success is presented and simulated in different models of quantum networks. We also derive approximations on the complexity scaling of our algorithms that corroborate the polynomial runtimes of the simulations. Furthermore, we determine the conditions yielding this simultaneous optimality for GHZ states with a higher number of qubits, and for other types of multipartite entanglement. This work paves the way to optimally generate multipartite quantum correlations over noisy quantum networks, an important resource for distributed quantum technologies.

Keywords: Quantum Networks, Routing, Multipartite Entanglement, Entanglement Distribution, GHZ Quantum States, Fidelity

Contents

List of Tables

List of Figures

List of Abbreviations

QIA	Quantum Internet Alliance
LU	Local unitary
LOCC	Local operations assisted with classical communication
SLOCC	Stochastic local operations assisted with classical communication
GHZ	Greenberger–Horne–Zeilinger
ER	Erdös-Rényi
SCL	Square cyclical lattice
PO	Pareto optimal
MOSP	Multi-objective shortest-path

Chapter 1

Introduction

1.1 Quantum Networks

The study of quantum networks is becoming increasingly important to develop a Quantum Internet, taking advantage of radically different quantum-based technologies. The Quantum Internet Alliance - QIA, an international effort joining forces from different researchers around the globe and in which this project is inserted, envisions the development of a Quantum Internet. While the classical internet is capable of connecting two points (or more) on earth through classical communications, a quantum internet would be able to do the same taking advantage of quantum communications which are fundamentally different from their classical counterparts. Together with quantum processing of information, it would alter the current paradigm of security and create a platform for a whole new set of applications.

Quantum communication relies on encoding the information in qubits which are the fundamental unit of quantum information, analogously as bits are the classical unit of information. Using these qubits, applications can be developed, for example protocols for ensuring secure and private communications and access to quantum computers [1–3] and quantum metrology networks [4, 5].

Because of the nature of the physical processes and technologies that are the ground base for quantum communications, a quantum network will be inherently distinct when it comes to describing the parameters that affect the communications, *e.g* fidelities of shared quantum states, decoherence times of quantum memories and probabilistic behaviour of quantum communication. All this requires finding an approach to quantum networks that is compatible with their description and current state of the art of the concealed quantum technologies. Moreover, the current stratification of the model of the quantum internet [6, 7] involves various stages of functionalities according to the current developments in the underlying engineering of devices capable of quantum communication and processing of information.

1.1.1 Entanglement

A significant concept for a number of quantum protocols is entanglement. It is a form correlation, intrinsic of quantum mechanics and without a classical analog. We say that two particles are entangled if they share this correlation, as we will describe with greater detail in the next chapter. In this thesis,

we are particularly interested in multipartite entanglement. While entanglement is usually used for two particles, when we perform the crossover to three or more particles, the term multipartite entanglement is convenient. Throughout this thesis we use the term entanglement for both cases, since what matters is that this correlation is present in the desired states. With this type of correlation on more than just two particles, protocols between more than two parties materialise, taking advantage of the possible multipartite quantum states utilised.

1.2 State of the Art

The upcoming Quantum internet development relies on connecting two points through a quantum link, relying on bipartite entanglement which has been in particular focus lately. All in all, multipartite entanglement, which is able to connect more than two points in a quantum internet, comes as the natural extension when regarding the types of states that we might consider. Moreover, some applications like quantum sensor networks [8–11], some quantum communication protocols [12–14] and different forms of performing distributed quantum computation [15, 16] all require the distribution of multipartite entangled states across a quantum network. Finding the optimal way to distribute this multipartite states is therefore necessary for the applications built on top of it, specially considering that some parameters of the final state can render the state useless, *e.g* the fidelity of a quantum state usually has a threshold that guarantees the presence of entanglement in the state [17]. Furthermore, understanding how the noise of the network, present in each individual quantum link, affects the final state is key to finding the optimal way to distribute such state.

For the bipartite case, the protocols for extending the range of entanglement are well established [18], and so is the characterisation of the parameters that affect the final quality of the entangled pair and its distribution. The problem of routing when it comes to bipartite entanglement is regarded in [19–23], mostly by maximising the probability and rate of success using varying schemes, including on-demand generation of entanglement and also considering continuous models, with continuous entanglement generation in the background.

In [18], a number of protocols for different stages of technology of the quantum repeaters - the fundamental device to realise a quantum internet, as they are the devices that are capable of extending the range of the entanglement through specific protocols - are demonstrated, as well as their base technology. Protocols for distributing entanglement between two quantum repeaters are examplified: the DCLZ entanglement distribution scheme in which entanglement is produced between an atomic ensemble and a photon via laser pulses inducing potential Raman transitions of the atomic ensemble. The protocols for extending the range of entanglement and purification of the entangled state are also described, together with the implications on the quality of the final state.

While in [18] only the necessary protocols are described, in [19] and [20] the optimal routing problem is the main goal, as to find the best way to distribute, using these protocols, entanglement between two end-nodes in a quantum network. To do this, the concept of routing metrics are introduced (which in this project we will introduce as a product of a more abstract description using algebraic theory in the context

of the routing problem, already studied in classical routing [24]) and the rate for generating entanglement is maximised between two end-nodes, taking into account communication times, decoherence times as the maximum time before the link is no longer suitable and probability of success of the different stochastic processes involved in distributing end-to-end entanglement.

When considering the case of multipartite entanglement distribution, some schemes for distribution have already been studied [25–27]. Despite this, the question of figuring what is the optimal way of distributing the desired states across a quantum network, given a distribution scheme, is still in its early stages with most work performing some simplifications. This is why a methodology capable of considering different parameters is important and allows for a more systematic approach of the problem.

In [25] a scheme of distribution of a special class of multipartite entangled states is introduced, as well as boundaries for the number of operations needed, which main goal is to minimise. However, only pure states are considered providing that the error propagation from bipartite entanglement in the intermediate steps is not taken into account, or any other type of parameter like the rate or probability of success. In other words, the routing problem is not a concern of this paper. In [26] different types of distribution schemes are presented and compared with each other, taking into account purification protocols for multipartite states, namely multipartite hashing which is detailed in [28–30].

In [31] and in [32] the waiting time and probability of success are analysed in chains while also hinting about the extension to multipartite distribution. However, in both works, an homogeneous network is considered and therefore the routing problem is absent.

All works previously mentioned focused on finding distribution schemes more than implementing them on a quantum network and finding the best way to distribute given such scheme, which introduces one of the main goals of this present work.

1.3 Objectives

The objectives of this work, as also detailed in the paper that result from this project [33], is to find the optimal way, *i.e* the way that optimizes the important parameters that define entanglement distribution (*e.g* fidelity, probability of success,...), to distribute multipartite entanglement in quantum networks, tackling the current shortcomings posed by assuming homogeneous networks, pure states and deterministic generation of entanglement.

Given the developments in the schemes for distributing multipartite entanglement, this should be done by creating a systematic approach, *i.e* a framework, to distribute optimally multipartite states across a network, taking into account a given scheme for distribution. We should be able to provide an algorithm that takes into account heterogeneous networks and the possibility of mixed states arising from errors in the network protocols, like entanglement generation and entanglement swapping. This is done by introducing different parameters that affect the quality of the final entanglement, namely the fidelity (a measure of quality of a quantum state), probability of success, communication times and quantum memory decoherence factor.

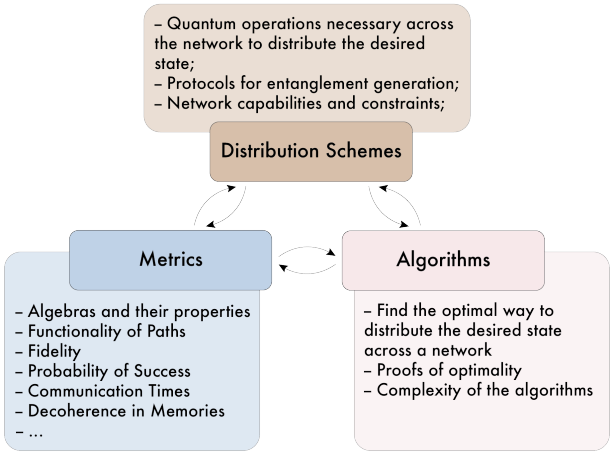


Figure 1.1: Overview of our work and the involved concepts.

More than just including this parameters, to make the framework complete, the method should leave space and considerations in case more parameters are needed or some constraints in the quantum network exist. This is crucial for a developing quantum internet with changing underlying technologies.

1.4 Thesis Structure

This thesis is organised in the following way: in *Chapter 2* we will spend sometime to go in detail through the background necessary to give context for the rest of the work. We start with the quantum background in *Section 2.1* describing qubits, the possible operations we can do with qubits, the possible states a set of qubits can share and the characteristics of some important classes of states, passing through a small introduction about networks and graph theory in *Section 2.2* and followed by the routing background in *Section 2.3* in which we introduce routing on networks with enough detail for the rest of the work. In *Chapter 3* we explain how to describe the parameters in a quantum network and adapt these parameters to utilise the routing algorithms. In *Chapter 4* we showcase the algorithms created to find the best way of distributing multipartite states, analyse the scaling problem in a network, present the simulations on a model of quantum network and finally derive expression for the complexity of our algorithms. Finally, in *Chapter 5* we present some concluding remarks and possible future work.

Chapter 2

Background

2.1 Quantum Background

2.1.1 Qubits

While classical information is coded in bits - two-level systems in which each level corresponds to either a zero or a one - quantum information is encoded in quantum bits, or qubits. These quantum bits are also two-level systems (if there were three levels it would be called a qutrit, four levels a qudit,...) but the underlying physical system behaves in a quantum manner, subject to quantum mechanics laws, and therefore can be in a state of superposition of the two possible levels. If we denote by "0" and "1" the two possible states of the qubit, then the qubit state can be described by:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{2.1}$$

where α and β are complex numbers that verify the state normalisation $|\alpha|^2 + |\beta|^2 = 1$. The bra-ket notation, or equivalently Dirac notation, was introduced in 1939 by Paul Dirac, as the name suggests, and denotes a vector in an abstract vector space with a scalar product antilinear on the first argument. This space, the Hilbert space, is where the possible states exist. Another way to visualize the state space, by performing a change of coordinates (see *Equation 2.2*), is the Bloch sphere (see *Figure 2.1*), which is a geometrical representation named after the physicist Felix Bloch.

$$|\psi\rangle = cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\cdot sin\left(\frac{\theta}{2}\right)|1\rangle \tag{2.2}$$

2.1.2 Operations on Qubits

There are a number of operations available for qubits that we can separate in two distinct categories - unitary and non-unitary. The ones we are most interested in are the unitary operations since they are reversible. Any one-qubit unitary operation has a corresponding transformation on the Bloch sphere correspondent to a rotation around some axis. The most common unitary one-qubit operations, equivalently one-qubit quantum gates, are:

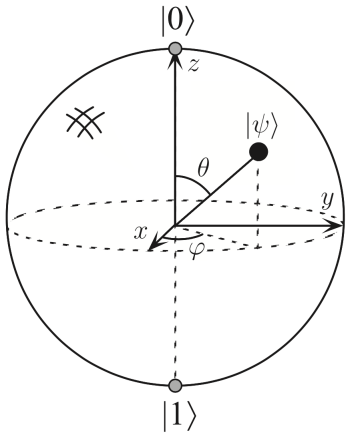


Figure 2.1: Representation of a qubit in the Bloch Sphere [34].

- Pauli gates - perform a π rotation around the corresponding axis

$$\hat{\sigma}_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \hat{\sigma}_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \hat{\sigma}_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.3}$$

  For example:

$$\hat{\sigma}_x |0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \tag{2.4}$$

- Hadamard gate - creates a state of superposition

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.5}$$

  For example:

$$\hat{H} |0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle \tag{2.6}$$

- Phase shift gates - introduce a relative phase

$$\hat{R}_\phi = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} \tag{2.7}$$

  For example:

$$\hat{R}_{\pi/2} |+\rangle = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix} = \frac{|0\rangle + i\,|1\rangle}{\sqrt{2}} = |y_+\rangle \tag{2.8}$$

Moreover, there also exist two-qubit gates. A few important examples are:

- Controlled unitary gates - perform a unitary gate dependent on the state of the control qubit

$$C(\hat{U}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{pmatrix} \tag{2.9}$$

- Swap gate - swap the two qubits states

$$SW\hat{A}P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.10}$$

The most used two-qubit gates are the $CNOT$ gate, which is just a controlled-X rotation ($\equiv C(X)$), and the controlled-Z rotation ($\equiv C(Z)$). It is know that with the Pauli gates and a two-qubit gate we can perform any quantum gate on any set of qubits (see Solovay–Kitaev theorem in [34]).

2.1.3 Density Matrix

The density matrix formalism is used to describe a state of a system of one or as many particles as necessary. Given an ensemble of possible pure states and associated probabilities $\{p_i, |\psi_i\rangle\}$, the density matrix can be written as *Equation 2.11* and will play an important role in the description of protocols, since the state of the system is fully described by this operator.

$$\hat{\rho} = \sum_i p_i \, |\psi_i\rangle \, \langle\psi_i| \tag{2.11}$$

In order to fully describe a quantum system state, this operator must follow a few conditions, namely:

1. **Trace Condition -** the trace must be equal to one (sum of probabilities of finding the system in different states must be unit)

2. **Positivity Condition -** the operator must be positive (all of the eigenvalues must be non-negative, since they represent probabilities)

The evolution of the states in this formalism will be described by unitary matrix and given by *Equation 2.12a*. A generalised measurement will be described by *positive operator valued measurement* (POVM) and the measurement probabilities and states after measurement will be given by *Equation 2.12b* and *2.12c* respectively.

$$\hat{\rho}' = U\hat{\rho}U^{+} \qquad p(m) = \mathrm{Tr}\left[M_m^{+} M_m \hat{\rho}\right] \qquad p(m) = \frac{M_m \hat{\rho} M_m^{+}}{\mathrm{Tr}\left[M_m^{+} M_m \hat{\rho}\right]} \tag{2.12}$$

One important concept, that is ultimately one of the key points of this project, is the existence of mixed states. By definition, a pure state is a state that can be represented by $\hat{\rho} = |\psi\rangle \langle\psi|$ where $|\psi\rangle$

is the state of the system. When noise is introduced in a system or it is allowed to interact with the surroundings, this initial pure state can decohere into a sum of different possible states, each with an associated probability and *Equation 2.11* is recovered. That new state is then called mixed state.

Fidelity Measure

Due to the existence of mixed states in our work, the definition of how close two quantum states are is necessary. This measure of closeness is called fidelity and is an important parameter to consider when trying to distribute entanglement, wether it is bipartite or multipartite. The formal definition [34] is, given two density operators $\hat{\rho}$ and $\hat{\sigma}$:

$$\begin{aligned} F(\hat{\rho},\hat{\sigma}) &= \mathrm{tr}\sqrt{\hat{\rho}^{1/2}\hat{\sigma}\hat{\rho}^{1/2}} \\ &= \sqrt{|\langle\psi_\rho|\psi_\sigma\rangle|^2} \end{aligned} \tag{2.13}$$

Throughout this project, we use another alternate equivalent parameter, which we denote by fidelity as well, similar to what is found in [35]:

$$f(\hat{\rho},|\psi\rangle) = \langle\psi|\,\hat{\rho}\,|\psi\rangle \tag{2.14}$$

where $|\psi\rangle$ is the pure state pretended and $\hat{\rho}$ is the density matrix correspondent to the real state. It is easy to verify that $F(\hat{\rho},|\psi\rangle\langle\psi|)^2 = f(\hat{\rho},|\psi\rangle)$. It is also important to notice that this measure of closeness is invariant up to LU, *i.e*:

$$\begin{aligned} f(\hat{O}\hat{\rho}\,\hat{O}^\dagger,\hat{O}\,|\psi\rangle) &= \langle\psi|\,\hat{O}^\dagger\hat{O}\hat{\rho}\,\hat{O}^\dagger\hat{O}\,|\psi\rangle \\ &= \langle\psi|\,\hat{\rho}\,|\psi\rangle \\ &= f(\hat{\rho},|\psi\rangle)\ , \qquad \forall\,\hat{O}\in LU \end{aligned} \tag{2.15}$$

2.1.4 Entanglement

As explained in the introduction, entanglement is a special type of correlation, intrinsically quantum, that allows for a number of protocols and is one of the main pillars of this project. Starting with a quantum system composed by two qubits, and a basis for the first qubit state space $\{|0\rangle_a, |1\rangle_a\}$ and for the second qubit state space $\{|0\rangle_b, |1\rangle_b\}$, performing the tensorial product of the Hilbert spaces, we arrive at the state space of the composite system $\{|0_a0_b\rangle, |0_a1_b\rangle, |1_a0_b\rangle, |1_a1_b\rangle\}$. From this, two systems are said to be entangled if they joint state is not separable in a tensorial product of some individual states of each system. For two qubits, the most relevant basis of maximally entangled states is the Bell basis:

$$\begin{aligned} |\Phi^+\rangle &= \frac{|0_a0_b\rangle + |1_a1_b\rangle}{\sqrt{2}} \quad |\Phi^-\rangle = \frac{|0_a0_b\rangle - |1_a1_b\rangle}{\sqrt{2}} \\ |\Psi^+\rangle &= \frac{|0_a1_b\rangle + |1_a0_b\rangle}{\sqrt{2}} \quad |\Psi^-\rangle = \frac{|0_a1_b\rangle - |1_a0_b\rangle}{\sqrt{2}} \end{aligned} \tag{2.16}$$

It is easy to verify that performing a measurement on one of the subsystems results in knowing the quantum state of the other system without requiring a second measurement. This is the type of correlation that both systems share by being entangled.

It is also worth noticing that any partial trace of one of the subsystems leaves the other in a state denoted by maximally mixed state, correspondent to a multiple of the identity matrix. The partial trace operation is physically correspondent to measuring one subsystem without gathering any information about the measurement. Algebraically it consists of finding an orthonormal basis for the desired subsystem and tracing the density matrix times the correspondent measurement:

$$\begin{aligned} \hat{\rho}_a &= \mathrm{tr}_b \hat{\rho}_{ab} \\ &= \mathrm{tr}(\hat{\rho}_{ab} \cdot M) \end{aligned} \tag{2.17}$$

where $M = \sum_i \mathbb{1}_a \otimes |i\rangle_b \langle i|_b$, for example, $M = \mathbb{1}_a \otimes |0\rangle_b \langle 0|_b + \mathbb{1}_a \otimes |1\rangle_b \langle 1|_b$.

Multipartite Entanglement

When the system is composed by more than two qubits - multipartite - the definition of entanglement becomes more complex. While for two qubits there is one class of entanglement and for three qubits there are two distinct classes [36], for more than three qubits it is still uncharted territory. The way to define these classes is by equivalence relations, in this case, if two states can be related under stochastic local operations assisted with classical communication (SLOCC). This is imposed by the fact that after distributing the many qubits, only local operations assisted with classical communication (LOCC) can be performed, particularly local unitary (LU) transformations, enlargement of Hilbert spaces and classical communications. It has been proven [37] that LU is equivalent to LOCC. If this LOCC equivalence is not deterministic, but stochastic instead, *i.e* there is a non-vanishing probability of conversion of the states under LOCC, it becomes a stochastic equivalence SLOCC which is exactly the way the classes of entanglement appear.

For two qubits, the Bell basis state $|\Phi^+\rangle$ belongs to the only class of entanglement for two qubits, which means that every entangled state can be transformed to $|\Phi^+\rangle$ under SLOCC. When three qubits are considered, two different classes emerge. For any tripartite entangled state, it can be shown [36] that such state can be converted, by means of SLOCC, to either the Greenberger–Horne–Zeilinger (GHZ) state:

$$|GHZ\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}} \tag{2.18}$$

Or the W state:

$$|W\rangle = \frac{|001\rangle + |010\rangle + |100\rangle}{\sqrt{3}} \tag{2.19}$$

This existence of these two classes inequivalent under SLOCC means that if we have a state $|\psi\rangle$ related under SLOCC with $|GHZ\rangle$ and a state $|\phi\rangle$ related under SLOCC with $|W\rangle$, then $|\psi\rangle$ and $|\phi\rangle$ are not related under SLOCC.

The GHZ state is considered the maximally entangled state for three qubits, but when tracing any of the qubits the remaining system looses any form of entanglement. The W state however, after tracing out any of the possible qubits, the remaining system retains the maximum possible amount of entanglement. These properties make this states important for multi-party protocols like quantum information

splitting [38] and quantum multipartite key distribution [14], or more commonly denoted by conference key agreement.

Moreover, a generalisation of these states can be constructed for more than just three qubits. The generalisation for m-qubit GHZ states, also denoted by m-Cat in [37] in honour of Schrödinger's cat, is given by:

$$|GHZ_m\rangle = \frac{|0\rangle^{\otimes m} + |1\rangle^{\otimes m}}{\sqrt{2}} \tag{2.20}$$

On the other hand, the generalisation of the W state for m-qubits is given by:

$$\begin{aligned} |W_m\rangle &= \frac{|m-1,1\rangle}{\sqrt{m}} \\ &= \frac{|0...01\rangle + |0...10\rangle + ... + |10...00\rangle}{\sqrt{m}} \end{aligned} \tag{2.21}$$

where $|m-1,1\rangle$ denotes the totally symmetric state including m -1 zeros and 1 ones.

Graph States

More than different classes of entanglement under the SLOCC relation, there is also an important category of entangled multipartite states - graph states. Graph states have important known applications [39], such as quantum error correcting codes [40] and one way quantum computer (a measurement based form of performing quantum logic) [16]. It is important to notice that the GHZ state and its generalisation for any number of qubits is LU equivalent to a graph state, namely the star graph (every vertice connected only to one center vertice), which is also LU equivalent to a complete graph (every vertice connected to every other vertice).

A graph state has an equivalent representation, as the name suggests, to a graph. With this equivalent representation, a variety of tools from graph theory [41] become instantaneously available and are proven to be very useful when trying to construct protocols for distributing entanglement over an arbitrary network, that can be represented by a graph. A graph can be described by a collection of vertices, V, and a description of which vertices are connected by an edge, E. A graph is then a pair $G = (V, E)$ of a finite set $V \subset \mathbb{N}$ and a set $E \subset [V]^2$, the elements of which are subsets of V with two elements each. A graph state associated with a simple graph $G = (V, E)$ is defined [25] as the following:

$$|G\rangle = \prod_{(a,b)\in E} CZ_{a,b} \, |+\rangle^V \text{ ,where } |+\rangle^V := \bigotimes_{a\in V} |+\rangle^a \tag{2.22}$$

with $|+\rangle^a = \frac{1}{\sqrt{2}}\Big(|0\rangle + |1\rangle \Big)$ and $CZ_{a,b}$ being a controlled-Z operation on qubits a and b.

Using these graph states, three elementary graph operations can be introduced as physical operations on graph states. They are as follows:

1. **Vertex Deletion -** This operation removes one vertex and all the associated edges from the graph. Physically, it is implemented by the Pauli measurement of the relevant qubit in the Z basis.

2. **Local Complementation on a vertex -** This graph operation inverts the subgraph induced by the neighbourhood N_a of the concerned vertex a (see *Figure 2* of [25]). It is implemented

by applying the relevant operation to the qubits of $a \cup N_a$, described by the quantum operator $U_a^\tau := e^{-\frac{i}{4}\cdot\hat{X}_a} \bigotimes_{b\in N_a} e^{-\frac{i}{4}\cdot\hat{Z}_b}$ acting on $|G\rangle$.

3. **Edge addition (deletion) -** By applying a controlled-Z operation between two qubits belonging to the same node, an edge between two nonadjacent (adjacent) vertices is created (deleted).

2.1.5 Quantum Operations

A quantum operation is a formalism used to describe the evolution of quantum systems in various situations. Any quantum gate has a description within this formalism, and so does any stochastic or deterministic interaction of the quantum system with its surroundings, which is crucial for developing methods of analysing the impact of decoherence while trying to distribute entanglement across a quantum network.

A quantum operation is described in terms of the density matrix. It is a map $\mathcal{E}(\cdot)$ that acts on a density matrix and transforms it into another one:

$$\rho' = \mathcal{E}(\rho) \tag{2.23}$$

The most common representation used for quantum operations is called operator-sum representation and consists of finding a set of operators that describe the interaction or process the system evolves through:

$$\mathcal{E}(\rho) = \sum_k \hat{E}_k \rho \hat{E}_k^\dagger \tag{2.24}$$

Using this description, it is easy to verify that any quantum gate $\hat{U}$ is a quantum operation with $\{\hat{E}_k\} = \{\hat{U}\}$ and so is any measurement described by a set $\{\hat{M}_k\}$. Furthermore, while not all quantum operations are trace preserving, the ones most important to us will be trace preserving (some authors call this special set of quantum operations quantum channels). This characteristic implies a restraint on the set of operators that describe the quantum operation, namely:

$$\sum_k \hat{E}_k \hat{E}_k^\dagger = \mathbb{1} \tag{2.25}$$

Equivalence of Quantum Operations

One important feature of this description of quantum operations is that some quantum operations are equivalent, *i.e* they produce the same outcome on any state subject to it. As we see in *Theorem* 6.30 of [42]:

Theorem 2.1.1. ***(Theorem for Krauss representations)*** *Let $(A_n)_{n\in\mathcal{N}}$ and $(B_m)_{m\in\mathcal{M}}$ be families of bounded operators on $\mathcal{H}$ such that:*

$$\sum_{n\in\mathcal{N}} A_n^\dagger A_n = \sum_{m\in\mathcal{M}} B_m^\dagger B_m = I$$

Consider the quantum channels:

$$\rho \mapsto \mathcal{L}_1(\rho) = \sum_{n\in\mathcal{N}} A_n \rho A_n^\dagger$$
$$\rho \mapsto \mathcal{L}_2(\rho) = \sum_{m\in\mathcal{M}} B_m \rho B_m^\dagger$$

If there exists a complex unitary matrix (eventually of infinite size) $(u_{nm})_{n,m\in\mathcal{N}}$ *(Assume that they are indexed by the same set* $\mathcal{N}$ *by adding eventually 0's to the smallest list) such that:*

$$A_n = \sum_{m\in\mathcal{N}} u_{nm} B_m$$

for all n*, then* $\mathcal{L}_1$ *and* $\mathcal{L}_2$ *define the same quantum channel.*

This theorem defines the necessary relation between sets of operators that translate in equivalent quantum operations.

Depolarisation Channel

The depolarisation channel is a quantum channel that is capable of modelling noise or decoherence in bipartite entanglement, due to an equivalence present, which we will further detail. Considering a system of only one qubit, the action of the depolarising channel is given by:

$$\mathcal{D}(\rho, p) = p\rho + \frac{1-p}{3} \cdot (\hat{X}\rho\hat{X} + \hat{Y}\rho\hat{Y} + \hat{Z}\rho\hat{Z}) \tag{2.26}$$

However, we require a definition of the depolarising channel capable of acting on specific qubits of a larger system of qubits. Such depolarising channel is a map acting on a set of qubits $\mathcal{N}$, performing the action of 2.26 on qubit i:

$$\mathcal{D}(\rho, p, i) \equiv \mathcal{D}_i(\rho, p) = p\rho + \frac{1-p}{3} \cdot (\hat{X}_i\rho\hat{X}_i + \hat{Y}_i\rho\hat{Y}_i + \hat{Z}_i\rho\hat{Z}_i) \tag{2.27}$$

where for each Pauli gate $\hat{\sigma}_i \equiv \hat{\sigma}_i \otimes \mathbb{1}^{\otimes\mathcal{N}\setminus i}$ and $\mathcal{N} \setminus i$ is the set of all qubits except qubit i.

The meaning of this transformation, as the name suggests, is a depolarisation of the qubit state from the original state applying a small amount (controlled by p) of error in each possible direction (X,Y and Z), which do not actually work as directions, as much as of types of possible errors (X constitutes a bit-flip error, Z a phase-flip error and Y both a phase-flip and bit-flip error). Considering we have a Bell State of the form $|\phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$, applying this transformation on either one of the qubits will result in a Werner State, as demonstrated by the following equation:

$$\mathcal{D}_2(|\phi^+\rangle\langle\phi^+|, F) = F|\phi^+\rangle\langle\phi^+| + \frac{1-F}{3} \cdot (\hat{X}_2|\phi^+\rangle\langle\phi^+|\hat{X}_2 + \hat{Y}_2|\phi^+\rangle\langle\phi^+|\hat{Y}_2 + \hat{Z}_2|\phi^+\rangle\langle\phi^+|\hat{Z}_2) \tag{2.28}$$
$$= F|\phi^+\rangle\langle\phi^+| + \frac{1-F}{3} \cdot (|\psi^+\rangle\langle\psi^+| + |\psi^-\rangle\langle\psi^-| + |\phi^-\rangle\langle\phi^-|) \tag{2.29}$$

using for every Pauli gate $\hat{\sigma}_2 \equiv \mathbb{1}_1 \otimes \hat{\sigma}_2$ again, for simplicity.

Another alternative representation of the action of the depolarising channel of density matrices exists. Consider we have some state $|\phi\rangle$ represented by the matrix $\rho \in \mathfrak{D}_n = \{A \in Mat_{2^n}(\mathbb{C}) : A = A^\dagger, Tr(A) = 1\}$. The action of a depolarisation channel on particle i of such element is also given by the following expression:

$$\begin{aligned} \mathcal{D}_i : \mathfrak{D}_n \times [0,1] &\longrightarrow \mathfrak{D}_n \\ (\rho_i \otimes \rho_{\mathcal{N}\setminus i}, F) = (\rho, F) &\longmapsto \frac{1+2F}{3}\rho + \frac{2(1-F)}{3}(\hat{Y}\rho_i\hat{Y})^T \otimes \rho_{\mathcal{N}\setminus i} \end{aligned}$$

Given that the action on the last element of the previous expression is linear, we can further simplify and arrive at the following result:

$$\mathcal{D}_i(\rho, F) = \frac{1+2F}{3}\rho + \frac{2(1-F)}{3}\ \Lambda_i(\hat{Y}_i\rho\hat{Y}_i) \tag{2.30}$$

where Λ_i denotes the partial transposition with respect to particle i and $\hat{Y}_i$ is, just as before, $\mathbb{1}_1 \otimes \mathbb{1}_2 \otimes ... \otimes \hat{Y}_i \otimes ... \otimes \mathbb{1}_n$. Using this form it is easier to understand some of the properties of this map, namely:

1. $\mathcal{D}_i(\alpha\rho_1 + \beta\rho_2, F) = \alpha\mathcal{D}_i(\rho_1, F) + \beta\mathcal{D}_i(\rho_2, F)$ (Linearity on the first argument)
2. $\mathcal{D}_i(\mathcal{D}_j(\rho, F_j), F_i) = \mathcal{D}_j(\mathcal{D}_i(\rho, F_i), F_j)$ (Commutativity on the indice of the qubit)
3. $\mathcal{D}_i(\rho_1 \otimes \rho_2, F_i) = \mathcal{D}_i(\rho_1, F_i) \otimes \rho_2$ with $i \in \mathcal{N}_1$ [1]
4. $U_i\mathcal{D}_j(\rho, F)U_i^\dagger = \mathcal{D}_j(U_i\rho U_i^\dagger, F)$ for any U_i unitary single qubit operation acting on any qubit i
5. $CZ_{i,j}\mathcal{D}_k(\rho, F)CZ_{i,j}^\dagger = \mathcal{D}_k(CZ_{i,j}\rho CZ_{i,j}^\dagger, F)$ for any set of indices $i, j \neq k \in \{1, 2, ..., n\}$ [2]
6. $\mathcal{L}_i(\mathcal{D}_j(\rho, F)) = \mathcal{D}_j(\mathcal{L}_i(\rho), F)$ for any quantum channel $\mathcal{L}_i(\cdot)$ acting on qubit $i \neq j$

2.1.6 Distributing Entanglement

Entanglement is the basis for many quantum communication protocols, as we have discussed earlier, making its distribution a pivotal study in quantum networks. The main goal of a quantum network is to enable communication between two or more terminals that can be far apart from each other. However, due to losses in the channels of communications (air, optical fiber,...) and given that a qubit ca not be copied or amplified due to quantum mechanics laws, a solution for guaranteeing communication between any points must be created - a quantum repeater. A quantum repeater is a physical device capable of executing a set of protocols to enlarge the range of the entanglement [18]. Its physical implementation depends on the stage of functionality of the quantum network [6], a way of characterising the different stages in the development of a full-fledged quantum internet, ranging from quantum trusted repeaters to quantum repeaters with memories.

[1] $\mathcal{N}_1$ is the set of qubits described by ρ_1

[2] $CZ_{i,j}$ is the controlled-Z gate with control on qubit i and target on qubit j (or vice-versa since it is symmetric on the indices $i.e\ CZ_{i,j} = CZ_{j,i}$)

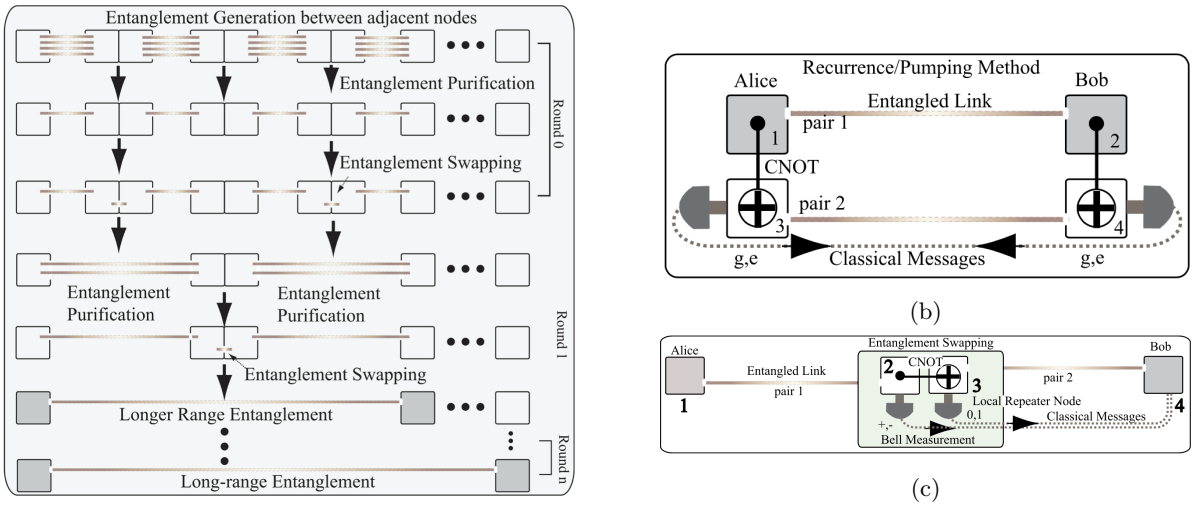


Figure 2.2: From [18] (*a*) Representation of the protocol the quantum repeater has to implement. (*b*) Representation of the protocol for entanglement purification. (*c*) Representation of the protocol for entanglement swapping.

There are three primary protocols required to create the long-range entanglement that can be used for quantum communication tasks, namely entanglement generation, entanglement purification and entanglement swapping. The representation of the type of protocol that a quantum repeater must be able to do can be found in *Figure 2.2a*.

Entanglement generation

Entanglement generation is the process for creating entangled links between network nodes. At this step, entanglement is only created between adjacent nodes. Using some physical system that can encode quantum information, for instance, using single atoms (or artificial atoms) within cavities or ensembles of atoms in a vapor. To transmit the information, photons are a good tool for the job. After interacting with the qubit holding the information, they can carry an entangled state (between them and the qubit), which after reaching another node of the network, is able to create entanglement between the two qubits in each node, creating a way of distributing entanglement between adjacent nodes.

One example of protocol for entanglement generation is the one introduced in [18], called DCLZ scheme, named after Duan, Lukin, Cirac and Zoller [43], consisting of using atomic ensembles and different levels of excitation as qubits and photons that excite the atomic ensembles to distribute entanglement with another atomic ensemble.

Entanglement Purification

One of the biggest setbacks with quantum technologies nowadays is that quantum states suffer from decoherence, this is, quantum states quickly transform themselves into other states, therefore loosing the information they had. In order to counteract this problem, entanglement purification protocols have been designed. The aim of the process is to create a higher quality entangled state from a number of lower

quality ones. Consider the Bell state $|\phi^+\rangle\langle\phi^+|$, if a dephasing error or other types of errors associated with imperfect local operation occurs, the altered state can be expressed as in *Equation 2.31*, which is called a Werner state [37].

$$\begin{aligned}\hat{\rho}_w &= F\,|\phi^+\rangle\langle\phi^+| + \frac{1-F}{3}\Big(\,|\phi^-\rangle\langle\phi^-| + |\psi^+\rangle\langle\psi^+| + |\psi^-\rangle\langle\psi^-|\,\Big) \\ &= \frac{1-F}{3}\mathbb{1}_4 + \frac{4F-1}{3}\,|\phi^+\rangle\langle\phi^+|\end{aligned} \tag{2.31}$$

with $|\phi^\pm\rangle = 1/\sqrt{2}(|00\rangle \pm |11\rangle)$ and $|\psi^\pm\rangle = 1/\sqrt{2}(|01\rangle \pm |10\rangle)$ corresponding to the Bell basis. Considering this Werner state is shared between qubits 1&2 and 3&4, the recurrence method protocol for entanglement purification, represented in *Figure 2.2b*, requires application of $CNOT$ gates between qubits (1&3) and (2&4) followed by a measurement on qubits (3&4) in the computational basis, taking only the outcomes when both qubits are in the same state. The resulting state will also be a Werner state with a new higher fidelity given by *Equation 2.32*. The case considering different pair fidelities can also be seen in [18].

$$F_p = \frac{F^2 + 1/9\cdot(1-F)^2}{F^2 + 2/3\cdot F(1-F) + 5/9\cdot(1-F)^2} \tag{2.32}$$

Entanglement Swapping

Entanglement swapping is the protocol that enlarges the range of the entanglement by performing a set of operations at the quantum repeater node that translate in sharing an entangled state between the terminal nodes.

Assuming that there is an entangled link between qubits 1&2 and 3&4 as in *Figure 2.2c*, performing a $CNOT$ operation between qubits 2&3 and then measuring them in the Bell basis will project the state of qubits 1&4 in $|\phi^+\rangle$, up to a local unitary correction, $\{\mathbb{1}, \hat{X}, \hat{Z}, \hat{Z}\hat{X}\}$, depending on the measurement outcome. This operation decreases the fidelity of the final state shared between the terminals, which is the main source for the problem of finding the best form in a quantum network to distribute bipartite entanglement.

If the initial entangled pairs can be modelled as a Werner states with fidelities F_1 and F_2, which is in fact equivalent to placing a depolarising channel over whichever qubit of each Bell State ϕ^+ (see *Equation 2.29*), the final fidelity of the new Werner state shared between the terminals will be given by:

$$F' = F_1F_2 + \frac{(1-F_1)(1-F_2)}{3} \tag{2.33}$$

By performing a change of variables that takes advantage of the multiple representations of the Werner state in *Equation 2.31*, $\gamma = {}^{4F-1}/_{3}$, the equations become:

$$\hat{\rho}_w = \frac{1-\gamma}{4}\mathbb{1}_4 + \gamma\,|\phi^+\rangle\langle\phi^+| \tag{2.34}$$

$$\gamma' = \gamma_1\cdot\gamma_2 \tag{2.35}$$

This simplifies the problem and provides a way of characterising how each link in a path of a quantum network will affect the fidelity of the final state after performing the successive entanglement swaps across each path. Moreover, this protocol will prove to be insightful when trying to find and characterise the

fidelity in a protocol that considers not only of bipartite entanglement, but also multipartite entanglement, which is one of the goals throughout this project.

2.2 Networks and Graphs

Networks and graph theory are closely related since graphs are essentially what networks are. A graph [41] is a mathematical object that can be described by a set of vertices and a set of edges connecting those same vertices. It can be completely defined by an adjacency matrix A with each component $a_{i,j}$ defining the weight of the edge connecting vertices i and j, which is an alternative formulation to the one we already presented when considering graph states in *Section 2.1.4*. From this, it is obvious that weights can not be negative. If they are zero, such edge does not exist and if the matrix is symmetrical, then the graphs is undirected, *i.e*, the weight of going in one direction of an edge is the same as going on the opposite direction.

A random network is a statistical ensemble, where each member—a particular configuration of vertices and edges—is realised with some prescribed probability (statistical weights). One example of these types of networks are Erdös-Rényi networks, with Poisson distribution for the number of neighbours each node has. Complex networks can model an handful of current real-world networks, including the World Wide Web, the Internet, basic cellular networks, and many others. In this work, we will closely follow the descriptions of Erdös-Rényi networks and square cyclical lattices. We chose this two models, since they are both well studied models and capture some of the properties of a possible quantum internet, namely the small world property present in Erdös-Rényi networks and the added complexity for square cyclical lattices.

It is also important to distinguish between structural and functional connectivity in a network. This difference comes form the network construction and edge associated parameters, for once, if there is a parameter threshold for some form of connection between two nodes, while there may exist an structural path to it, *i.e* a set of edges connecting both nodes, this connection might not be functional in the sense that such parameter falls below the threshold and the connection is not functionally possible. In this chapter we only deal with structural connectivity in networks and leave the functional to the next chapter where we introduce a set of tools to best describe these parameters and everything associated.

2.2.1 Erdös-Rényi Networks

The most important parameter that characterises an Erdös-Rényi (ER) is its degree distribution. The degree is the number of edges connected to any particular vertex, which for ER networks follows a poissonian distribution with average value λ:

$$P(\text{degree } = z) = Pois(\lambda; z) = \frac{\lambda^z e^{-\lambda}}{z!} \tag{2.36}$$

From the definition of Poisson distribution, it is easy to verify that λ is in fact the average degree. Besides the average degree, another important parameter for the network that can be derived from its

degree distribution, using some results and properties of generating functions [41, 44], is the maximum path length of a graph, or diameter of the graph. For a ER network the diameter is:

$$d_{ER} = \frac{\ln N}{\ln \lambda} \tag{2.37}$$

With N being the number of nodes in the graph.

2.2.2 Square Cyclical Lattice Networks

Another network which we consider is the square cyclical lattice (SCL), which is constituted by an $2D$ array of nodes arranged in a square lattice, every node connected to the closest four nodes. The cyclical means that at the edges of the lattice they connected with the opposite edge neighbours. Topologically, it can be perceived as a torus. The degree distribution of this network is 4 for every node and the diameter of this network is:

$$d_{SCL} \sim \sqrt{N} \tag{2.38}$$

Where N is the number of nodes of the network.

2.3 Routing Background

Routing is solving the problem of finding the best path in some network or graph composed by nodes and edges with associated parameters. There have been numerous advances in routing throughout the birth and development of the internet, that will show to be convenient when trying to solve the problem of routing in a quantum network. For once, the fidelity has an associated cost-function, equivalently metric, that distinguishes different paths, with higher fidelities reducing the amount of error created by less than perfect entanglement and imposing a restriction on the minimum end-to-end fidelity, since for fidelities inferior to $1/2$ the amount of entanglement present vanishes. In addition, there will be other metrics that will be proven relevant to describe the entanglement distribution associated with parameters of each edge.

While most work on routing undertakes a more practical and pragmatic approach, there is also a more fundamental theory of algebraic routing [24], which will be the starting point in this approach.

2.3.1 Routing Algebra

A network can be described as an undirected graph comprised of nodes $\{u_i\}$ with cardinality N and a set of links between nodes $\{(u_i, u_j)\}$ with cardinality E. A routing algebra is a special tool that allows for standardisation of the way a routing problem is solved. While there are different ways of defining it, even among the same author, they all end up being the same in less or more compact descriptions [24, 45, 46]. In this project, we use the definition present in [24], which allows for a better introduction into routing algebras:

Definition 2.3.1. Algebra for Routing is an ordered septet $(W, \preceq, L, \Sigma, \phi, \oplus, f)$ comprised as follows: W a set of weights, $\preceq$ a total order, L a set of labels, Σ a set of signatures, ϕ a special signature, $\oplus$ a binary operation that maps pairs of labels and signatures into a signature and a function f that maps signatures into weights.

From this definition, we can establish the parallelism with a quantum network. The set of labels is the set of edges associated parameters, the set of signatures are the parameters of the paths in the quantum network, the set of weights are the cost that this paths can have under the function f, which maps the paths parameters into a cost (or weight), the total order defines an ordering on the costs, *i.e* if we want to minimise then $\preceq = \leq$, the special signature is the signature used to discard paths (equivalently the no-possible-path signature) and the binary operation $\oplus$ is the way a path is extended with an edge to a bigger path.

With this algebra, a set of important characteristics can be defined and will be fundamental when finding the necessary properties for optimal convergence to the solution in the shortest-path algorithms.

The condition of monotonicity means that every time we extend the path, its weight will follow the same trend, defined by the total order of the algebra. This condition for an algebra for routing guarantees the convergence to a path under an appropriate algorithm, even though this convergence is not necessarily to the optimal path.

Definition 2.3.2. (Monotonicity) an algebra for routing is called monotone if:

$$\forall\, l \in L, \alpha \in \Sigma : f(\alpha) \preceq f(\alpha \oplus l)$$

A strict monotonicity can also be considered with the relation $\prec$ defined such that $a \prec b$ if $a \preceq b$ and $a \neq b$.

Definition 2.3.3. (Strict Monotonicity) an algebra for routing is called strictly monotone if:

$$\forall\, l \in L, \alpha \in \Sigma : f(\alpha) \prec f(\alpha \oplus l)$$

It is easy to verify that, by definition of neutral element of $(\Sigma, \oplus)$ which we will call e_Σ, that for an algebra to be strictly monotone, the set of labels must not contain e_Σ. As for the isotonicity condition, it states that if the optimal path is extended, it will continue being the optimal path. From this property, together with the monotonicity, the algebra for routing will converge always to the optimal path [24] under an appropriate algorithm. There are two cases of isotonicity, depending on the argument of the binary operation considered, namely left and right-isotonicity [47]. An algebra is called isotonic is it is both left and right-isotonic.

Definition 2.3.4. (Right-Isotonicity) an algebra for routing is called right-isotone if:

$$\forall\, l \in L, \alpha, \beta \in \Sigma : f(\alpha) \preceq f(\beta) \Rightarrow f(\alpha \oplus l) \preceq f(\beta \oplus l)$$

Definition 2.3.5. (Left-Isotonicity) an algebra for routing is called left-isotone if:

$$\forall\ l \in L, \alpha, \beta \in \Sigma : f(\alpha) \preceq f(\beta) \Rightarrow f(l \oplus \alpha) \preceq f(l \oplus \beta)$$

2.3.2 Algebras for Multi-Objective Routing

Not always a shortest problem path can be described with only one parameter. While for more than one parameter, from the abstract definition of routing algebra, such descriptions are made possible, usually they do not verify the required conditions for optimal convergence of the shortest-path algorithms, usually failing the isotonicity property, *e.g* the shortest-widest algebra [45]. One possible way to overcome this problem is to separate the different parameters into different algebras, independently monotonic and isotonic and use a multi-objective approach, introduced in 1984 by Martins [48], defining additional relations and a new definition of optimal paths, required when dealing with multiple objectives for routing.

Definition 2.3.6. Set of Algebras for k-Multi-Objective Routing is a set of ordered septets [3] $\{(W^i, \preceq^i, L^i, \Sigma^i, \phi^i, \oplus^i, f^i)\}$ with $i = 1, ..., k$. Each member of each septet is comprised as follows: W^i a set of weights, $\preceq^i$ a total order, L^i a set of labels, Σ^i a set of signatures, ϕ^i a special signature, $\oplus^i$ a binary operation that maps pairs of labels and signatures into a signature and a function f^i that maps signatures into weights.

In the case for $k = 1$, we return at the usual definition of algebra for routing considering only one objective. The important relation to consider, introduced in [48], that glues all algebras and defines the order between the different paths is now the dominance relation:

Definition 2.3.7. (Dominance) let ω and ν be two different signatures in $\{\Sigma^i\}$. ω dominates ν, ω D ν if $f^j(\omega^j) \preceq^j f^j(\nu^j)\ \forall j \in \{1, ..., k\}$ and the strict order holds at least once.

This relation states that given two different paths in a set of algebras for routing, unless every parameter of one path is better or equal than the corresponding parameter of the other (with the strictly better ordering happening at least once), then neither path is better than the other, *i.e* neither path dominates the other. From the definition of total order, which requires that $\forall a, b$ either $a \preceq b$ or $b \preceq a$, it is clear to see that the dominance relation is not a total ordering, justifying why we can not attack the problem only considering one algebra describing every parameter.

Proposition 2.3.1. *For the set of algebras for multi-objective routing, if the individual algebras are monotone and isotone, then the dominance relation is inherited, i.e, ω, ν are two distinct signatures such that $\omega\ D\ \nu$, then:*

$$\forall\ l \in L \equiv \{L^i\} : (\omega \oplus l)\ D\ (\nu \oplus l)$$

[3] every set $\{\chi^i\} \equiv \{\chi^1, ..., \chi^k\}$

Proof. From the definition of dominance, we get that if ω D ν then $f^j(\omega^j) \preceq^j f^j(\nu^j)\ \forall j \in \{1, ..., k\}$ (with the strict order happening once). From isotonicity we get that $\forall\, i, \forall\, \omega, \nu \in W : f^i(\omega^i) \preceq^i f^i(\nu^i) \Rightarrow f^i(\omega^i \oplus l^i) \preceq^i f^i(\nu^i \oplus l^i), \forall\, l \in L$. For the case of the strict order, it is important to notice that isotonicity implies that the order also maintains for the strict order[4]. Because of this isotonocity property, then we get that $(\omega \oplus l)$ D $(\nu \oplus l)$, and therefore we say that the dominance relation is inherited. □

Remark. For the set of algebras for multi-objective routing, if the individual algebras are isotone, let ω, ν are two distinct signatures such that $\omega \not\mathrel{D} \nu$, then:

$$\forall\, l \in L \equiv \{L^i\} : (\omega \oplus l) \not\mathrel{D} (\nu \oplus l)$$

Proof. Identical to *Proposition 2.3.1*. □

This proposition is one of the most important and ultimately justifies the optimality of our algorithms. By defining each algebra independently and requiring them to be individually isotonic, then we do not need to worry about possible interferences of parameters. If there are these interferences, then the algebras are not isotonic and we can find another formulation of the objectives that results in individual isotonic algebras, usually by separating different parameters, *i.e* consider that we have two paths represented each by a vector of two signatures:

$$\begin{aligned} \alpha(t) = (\alpha_1(t), \alpha_2(t)) \text{ and } \beta(t) = (\beta_1(t), \beta_2(t)) \\ \exists t_1, t_2 : \alpha(t_1) \text{ D } \beta(t_1), \alpha(t_2) \not\mathrel{D} \beta(t_2) \end{aligned} \tag{2.39}$$

This would imply that the underlying algebras are not isotonic for all t since we would be able to find a t_1 and a t_2 for which the ordering of one of the algebras would switch.

Given the dominance relation, the optimal solution considering the k-criteria becomes a more complex problem. To find the solution that minimises all of the criteria is not always possible. Nevertheless, there exists a privileged set of paths that are non-dominated by other paths and which can be considered the set of optimal paths for the problem. This set of paths is called the set of Pareto Optimal (PO) paths. Another important definition is the definition of possible path, which is just a path such that every parameter is not a special signature of the correspondent algebra.

Definition 2.3.8. (Pareto optimal paths) let X denote the set of signatures correspondent to the paths connecting node i and node j. Now let the set $X_d = \{\forall x \in X, \exists y \in X : y \text{ D } x\}$. Then the set $X_p = X \setminus X_d$ is the set of non-dominated signatures connecting node i and node j.

Definition 2.3.9. (Possible Path) let ω be a signature in $\{W^i\}$. We say that ω is a viable signature (correspondent to a possible path) if $\forall\ i = 1, ..., k : \omega^i \neq \phi^i$.

Moreover, an optimal path is always necessarily a possible path. All these definitions and propositions will reveal very useful when defining and proving the optimality of the algorithms.

[4] If for every two different signatures, that are not the special signature, the value of their weight is different. If this is not true then the algebras must be strict-isotone.

2.3.3 Algorithms for Multi-Objective Routing

While work related with multi-objective routing can be traced back to simpler cases up to two different objectives in the decade of 1970 by Vincke [49] and Hansen [50], the first general approach for any number of objectives and systematic approach, as seen on the previous subsection, was made by Martins [48], also introducing an algorithm capable of solving the shortest-path problem. This algorithm has ever since been studied, optimised in [51] and relaxed for some specific cases of objectives [52]. The algorithm used throughout this work is identical to the one in [51], slightly modified using a different data structure as described in *Algorithm 1*.

Algorithm 1 Multi-Objective Routing

procedure SHORTEST-PATH(*source*) ▷ Finds the shortest path to every node from the source
    $Nodes$:= Set of nodes of the network, each with underlying list of paths $Paths_u$ initialised as empty;
    A := Set of visited nodes of the network initialised as empty;
    B := Set of nodes to visit ordered as a priority queue data structure, with priority defined by the dominance relation;
    Initialise *source* $\leftarrow \{e_{\Sigma_i}\}$; ▷ $\{e_{\Sigma_i}\}$ are the neutral elements of $(\Sigma_i, \oplus_i)$
    Add *source* to B;
    while $B \neq$ empty **do**
        $node \leftarrow$ Top(B)
        Remove *node* from B and add to A;
        for $v \in$ neighbours($node$) **do**
            $Paths^{add} \leftarrow$ possible paths from $\{Paths^{(i)}_{node} \oplus Edge(node, v)\}$;
            if $Paths_v =$ empty **then**
                $Paths_v \leftarrow Paths^{add}$;
                Add v to B;
            if $Paths_v \neq$ empty **then**
                $Paths^p \leftarrow$ non-dominated paths of $Paths^{add} \cup Paths_v$;
                if $Paths^p \neq Paths_v$ **then**
                    $Paths_v \leftarrow Paths^p$
                    if $v \in$ A **then**
                        Add v to B and remove from A; ▷ needs to be revisited
                    if $v \notin$ A **then**
                        Update v in B;

Since some modifications have been made from the initial algorithm, it is important to prove the optimality, in the Pareto sense, of this algorithm under the monotonicity and isotonicity conditions.

The behaviour of this algorithm is the following: the source node is initialised with the neutral path and then visited; every time a node is visited, it concatenates its paths with the edges connecting it to each neighbour and writes those concatenated paths in its neighbours, as long as they do not dominate any previously existent paths, and adds the nodes to the queue, which is sorted to optimise the number of times a node is visited.

Theorem 2.3.1. *Monotonicity and isotonicity of every algebra for k-Multi-Objective Routing are sufficient conditions so the solutions found for each node of the network by Algorithm 1 (SA) constitute the set of Pareto Optimal Solutions (POS) for the shortest-path problem.*

Proof. Consider a graph G comprised by a set of $V = \{u_i\}$ vertices, or nodes, and a set of egdes $E = \{(u_i, u_j)\}$ connecting those vertices.

1. First, let us prove that there is no cycle in any shortest path. This is pretty straight forward as, for a cycle to be present in a shortest path it would have to be dominated by the neutral elements of $(\Sigma_i, \oplus_i)$, which is the same as not moving from the vertice we are in. Representing an edge by its associated label $l(u,v) = \{l^1(u,v), l^2(u,v), ..., l^k(u,v)\} \in L \equiv \{L^i\}$, considering that any cycle in a graph can be described by a sequence of edges $l(u_{i_1}, u_{i_2}) \oplus l(u_{i_2}, u_{i_3}) \oplus ... \oplus l(u_{i_{k-1}}, u_{i_k}) = \omega_{cycle} \in \Sigma$ and considering that $\forall\ (u,v) \in E(G) : l(u,v) \neq \{e_{W_i}\}$ (if not possible, we can always implement a simple way to prevent the same edge being added twice to a path), then $f^i(\omega^i) = f^i(l^i(u_{i_1}, u_{i_2}) \oplus^i l^i(u_{i_2}, u_{i_3}) \oplus^i ... \oplus^i l^i(u_{i_{k-1}}, u_{i_k})) \succ^i$ [5] $f^i(e_{\Sigma_i} \oplus^i ... \oplus^i e_{\Sigma_i}) = f^i(e_{\Sigma_i})\ \forall\ i = 1, ..., k$ which is the same as not moving from the vertice where we stand. The presence of no cycles will guarantee the convergence to some solution by the algorithm, since every time the algorithm adds the same edge twice (simple cycle) on one path, the dominance conditions imposed would prevent it from being a solution.

2. Second, let us check if the algorithm solutions in any vertice are necessarily solutions for the shortest-path problem ($SA \subseteq POS$). We do this proof by contradiction: consider a solution of the algorithm, z, that does not belong to the set of POS, $\exists\ y \in SA : y\ \mathrm{D}\ z$. There are two options for this:

   (1) either paths were added from *step* 12 of the algorithm, from the monotonicity and isotonicity of the algebras we realise that the previous vertice also contained at least one element that was not part of the solution of the previous vertice ($y\ \mathrm{D}\ z \implies \exists\ \tilde{y}, \tilde{z} \in W; l \in L : \tilde{y} \oplus l = y, \tilde{z} \oplus l = z$ such that $\tilde{y}\ \mathrm{D}\ \tilde{z}$)

   (2) the paths were added over successive iterations from *step* 17 of the algorithm after the first when the set was empty.

   From the conditions to add the paths to the set of solutions in *step* 15, we realise that only dominant solutions can be added, which implies that every time a wrong solution is added, it must be due to (1) and not (2). Following recursively to the beginning of the algorithm where the source is initialised to the neutral elements of the signatures, then, the first paths added to the neighbours of the source would have to be non-dominant and not part of the POS. Since the algebra is monotone, this is impossible, so the first solutions are indeed part of POS and so are the successive following solutions.

3. Third, let us check if the set of solutions found by the algorithm are in fact the only possible solutions ($SA \supseteq POS$). Consider there is a solution of POS, z, that is not in SA, $\forall\ y \in SA : z \not\mathrm{D}\ y$. This might have happened because:

   (1) it was not added when the node solutions were empty

   (2) it was not added before visiting the first time

   (3) it was not added after visiting the first time

[5] the relation $\prec$ is defined by $a \prec b$ if $a \preceq b$ and $a \neq b$ and $\succ$ is defined by $a \succ b$ if $b \prec a$

The first case (1) is obviously impossible from the dominance inheritance presented in *Theorem 2.3.1*. The second case (2) is also impossible since if it was a non-dominated path, then *step* 15 would imply its addition to SA. The third case is more complicated since a node can be visited a finite number of times (no cycles in any path). This number of times is in fact bounded by the number of neighbours due to the priority queue being defined by the dominance relation. If we have dominant paths coming from every neighbour of our vertex, the algorithm will ensure that only the dominant ones will be included in SA, and, if z is in fact a solution, it would be in SA. As previously, recursively going back to the source, then if z was indeed a member of POS, then it would have to be in SA.

From these three conditions, we get the proof of the theorem, and, therefore, optimality of the algorithm under individually monotone and isotone algebras. □

2.4 End-of-Chapter Remarks

In this first chapter we went over a lot of the background we will use throughout the rest of this thesis. The multipartite states introduced will be our main focus, specially the GHZ state, which will be the subject of many of our derivations. The fidelity measure is a crucial component of this thesis, as it is one of our main objectives to optimise. The abstract algebras introduced, together with the properties are very important to keep in mind. They are how we mathematically describe the important parameters which we want to optimise and, while their description might be too technical, they encapsulate the behaviour of choosing different paths or equivalently using different entangled pairs.

Chapter 3

Distribution Metrics for Quantum Networks

Like previously mentioned in *Section 2.3*, even though rather superficially, a metric is the cost-function associated to a parameter or set of parameters that attributes a cost, a weight, to a path in a graph. This will help define which path is better and allow for the resolution of the shortest-path problem. In *Section 2.3.1*, we also introduced a more fundamental way [24] of describing the theory behind the shortest-paths, the algebras for routing that hide the metric in themselves, namely through the binary operation $\oplus$ and the function f that maps signatures into weights. Because of this, from now on, we will refer to metric as the function that takes signatures (or paths) and labels (or edges) and maps them into weights - $f(\alpha \oplus l)$. The monotonicity and isotonicity properties introduced in *Definitions 2.3.2 and 2.3.5* continue to be well-defined and tractable when dealing with metrics.

One critical aspect when dealing with metrics for distributing multipartite (and bipartite) states is the scheme, or protocol, of distribution. This is the sequence of steps that must be undertaken so that the final state distributed among the terminals (the end-nodes) is the desired state, wether it is a GHZ state, a W state or any other entangled state. Therefore, the metric is always closely related with the scheme of distribution, and implicitly, with the state distributed.

In the bipartite case, the scheme of distribution is simply the concatenation of different links resulting in a path between the end-nodes. This concatenation is made through entanglement swapping protocols, which have well studied effect on the fidelity (see *Equations 2.33 and 2.35*), which form a monotone and isotone metric on the fidelity, as we will see in the next section. In this chapter, we will first go over metrics in the case of distributing bipartite entanglement and then consider two different distribution schemes for multipartite entanglement and describe some metrics, keeping an high level of detail for the fidelity metrics, which is one of the main goals of this project.

3.1 Distributing Bipartite Entanglement

Before jumping into distributing multipartite entanglement, since most schemes of distribution start from constructing paths and merging those paths, it is important to first introduce the metrics for distributing end-to-end bipartite entanglement. As previously described in *Section 2.1.6*, there are already protocols for distributing bipartite entanglement across a chain of entangled pairs: entanglement generation and entanglement swapping. The effect of these protocols on the fidelity of the state is known (*see Equation 2.35*). However, more than the fidelity of state, there are other crucial parameters for deciding the best way to distribute a given quantum state:

1. *Waiting time* - the time it takes between starting the protocol and signalling its completion. Since this is not always deterministic, this time must be considered part of a metric that takes also into account the probability of success and becomes the waiting time for each try.

2. *Quantum memories* - it is known [6] that in some of the early stages of the quantum internet, the capabilities of each node to preserve a given quantum state will depend on the quantum memories used. This memories, while being useful for storing the qubits for longer periods of time, will introduce an error that can be quantised and therefore minded across a network, given a distribution scheme.

3. *Probability of Success* - since some of the steps in distributing entanglement are not necessarily deterministic (entanglement generation, entanglement swapping,...), it is important to introduce a metric for the probability of success in generating end-to-end entanglement. This metric will depend on the nodes characteristics.

Some approaches have started to be studied recently in [31, 32, 53, 54]. While in [31, 32], the routing problem vanishes from the problem construction, assuming an homogeneous network, it provides good context and characterisation of the different processes involved in distributing bipartite entanglement. In [53, 54], the routing problem is defined in an heterogeneous network, but the fidelity metric acts as only a constraint with a minimum fidelity and not as a routing objective.

Using the characterisation of the probabilistic behaviour and the effect of memory times described in [31] we are able to construct the metrics that caracterise the entanglement distribution and will be used later when creating an algorithm that solves the routing problem. In this section and throughout the rest of this project we refer to entanglement generation as creating entanglement between two neighbouring nodes and end-to-end entanglement generation as creating entanglement between two end nodes across a chain using the entanglement swapping protocol.

3.1.1 Communication Time

Since both generation of entanglement and entanglement swapping require an heralding signal to verify if the processes were successful, then classical communication times are required to be taken into account. These times depend mostly on two different things: distance and the speed of light in the medium. For

simplicity let us consider the velocity to be always c which is the speed of light, even though if the velocity is different the approach is the same, since the parameter to consider is the time and not the distance. Then, the communication time will be given by $L_{a:b}/c$ where $L_{a:b}$ is the distance between nodes a and b.

3.1.2 Memory Times

At different stages of the quantum internet, the underlying physical system will impact the routing problem, specially considering that decoherence is an important factor to take into account when distributing entanglement between two nodes in a network. If the quality of the link (fidelity of the state) decreases bellow a certain threshold, then the link might not be usable for some protocols, so it is important to take into consideration the decoherence with time. This will depend on the quantum memories times that can vary from each physical device used.

Consider each node has a memory time given by τ_a. This means that the fidelity of the state will decrease as shown on *Equation 3.1* by $e^{\Delta t/\tau_a}$ after Δt time that the qubit is held on the quantum memory in node a [31].

$$\gamma' = \gamma \cdot e^{\Delta t/\tau_a} \tag{3.1}$$

Moreover, if two qubits share a state and each of them is stored in a different physical device with different memory times, then their joint fidelity is decreased by:

$$\begin{aligned} e^{\Delta t/\tau_a} \cdot e^{\Delta t/\tau_b} &= e^{\Delta t(1/\tau_a + 1/\tau_b)} = e^{\Delta t/\tau_{a:b}} \\ \tau_{a:b} &= \left[\frac{1}{\tau_a} + \frac{1}{\tau_b}\right]^{-1} \end{aligned} \tag{3.2}$$

3.1.3 Probability of Success

The probability of success will depend on two main protocols when distributing bipartite entanglement: entanglement generation and entanglement swapping. Each of them can be modelled by a geometric distribution as follows:

Entanglement Generation

This probabilistic behaviour will depend on parameters of each link, *i.e*, the connection between two different quantum repeaters. Assuming *w.l.o.g* a baseline time t_0, then each link probability for generating entanglement can be described by a time distribution, *i.e* a geometric distribution with a parameter depending on the link (where t comes in multiples of the elementary time t_0):

$$P(T^{a:b} = t) = p_{a:b}(1 - p_{a:b})^{t-1} \quad , \; t = \{1, 2, 3, ...\} \tag{3.3}$$

where $T^{a:b}$ is the random variable representing the time of generating entanglement between nodes a and b and $p_{a:b}$ is the probability of generating entanglement between a and b within t_0. In cases where the entanglement generation is near deterministic, this parameter can be discarded.

Entanglement Swapping

Since entanglement swapping is a probabilistic operation which happens at each node (or quantum repeater), the parameters are going to be node properties, instead of link as before. Its behaviour is similar to entanglement generation but instead of time, we have number of tries before the first success - therefore a geometric distribution is the best option [31].

$$P(K^a = n) = k_a(1 - k_a)^{n-1} \quad , \; n = \{1, 2, 3, ...\} \tag{3.4}$$

where K^a is the random variable representing the number of times before the swapping is successful of node a and k_a is the probability of the swapping being successful at the first time on node a.

3.1.4 Metric for Fidelity from Entanglement Swapping

As stated in *Equation 2.33*, the fidelity after the entanglement swapping protocol is given, in the γ change of variables, by:

$$\gamma' = \gamma_1 \cdot \gamma_2 \tag{3.5}$$

Because if the fidelity drops below the threshold of $1/2$ (which in γs corresponds to $1/3$), the entanglement present vanishes and the state is discarded, the correspondent algebra is $\gamma : \Big([1/3; 1) \cup \{0\}, \geq, (1/3; 1), (0; 1), 0, \oplus_\gamma, g\Big)$ where $\oplus_\gamma$ is the following binary operation:

$$\begin{aligned} \oplus_\gamma : (0,1) \times (1/3; 1) &\longrightarrow (0; 1) \\ (\gamma_{i:j}, \gamma_{j:k}) &\longmapsto \gamma_{i:j} \cdot \gamma_{j:k} \end{aligned} \tag{3.6}$$

And $g(\cdot)$ is the following function:

$$g(\gamma) = \begin{cases} \gamma & \gamma \geq 1/3 \\ 0 & \gamma < 1/3 \end{cases} \tag{3.7}$$

In *Chapter 4* this type of algebras with a truncation value gain a formal definition and some advantages are presented. Nonetheless, this algebra is both monotone and isotone and the proofs are presented in *Section B.1*.

3.1.5 Metric for Quantum Memories Decoherence

Starting from a chain of nodes capable of generating entanglement between the neighbours, there are two possible ways of achieving end-to-end entanglement:

1. in a pyramid-like scheme (*see [31]*) where if any process in extending the range of entanglement fails, only the previous steps of that link must be repeated from the beginning and the non-mutual entangled pairs wait its completion.

2. in a two-step scheme where all the entangled pairs between nodes are generated and then all the entanglement swapping (which are measurements in all intermediary nodes proceeded by an heralding signal) happen at the same time. If any of the individual processes fails, the scheme starts from the beginning.

It is easy to verify that the first scheme maximises the probability of generating end-to-end entanglement, but complicates the analytic description of the waiting time and probability while also not providing a limit for the memory decoherence. The second scheme however has an easier description and provides a fixed limit on the memory decoherence time since, if the process repeats, no qubit is stored in memory, so the maximum memory time will be the heralding of the two signals (entanglement generation and entanglement swapping). In [31], the first scheme is well studied under the homogeneous constraint for 2^n chains of entangled pairs. In our case we take advantage of the simpler description of the second scheme.

Under the two-step scheme, the generation of entanglement must be communicated before doing the entanglement swapping at each intermediary node. This means that, after t_0 time, we need to communicate throughout the whole chain the OK-signal to proceed. This will take:

$$t_{wait}^{(1)} = \sum_{i=1}^{n} \frac{L_i}{c} \tag{3.8}$$

Since at this point, the entanglement is being held between each two neighbouring nodes in the chain, then the contribution to the decoherence factor will be:

$$\sigma_d^{(1)} = \sum_{i=1}^{n-1} \frac{t_{wait}^{(1)}}{\tau_{i:i+1}} = \sum_{i=2}^{n-1} \frac{2t_{wait}^{(1)}}{\tau_i} + \frac{t_{wait}^{(1)}}{\tau_1} + \frac{t_{wait}^{(1)}}{\tau_n} \tag{3.9}$$

Afterwards, the heralding signal for the entanglement generation is received and the swapping starts. In the end, an heralding signal for the swapping must also be sent and will take again, at most:

$$t_{wait}^{(2)} = \sum_{i=1}^{n} \frac{L_i}{c} = t_{wait}^{(1)} \tag{3.10}$$

Which will translate in a decoherence factor of:

$$\sigma_d^{(2)} = \frac{t_{wait}^{(2)}}{\tau_{1:n}} = \frac{t_{wait}^{(2)}}{\tau_1} + \frac{t_{wait}^{(2)}}{\tau_n} \tag{3.11}$$

since if the swapping is successful, the link will be shared between the first and last node memories. This will result in a combined communication time and decoherence factor given by:

$$t_{wait} = t_{wait}^{(1)} + t_{wait}^{(2)} = 2\sum_{i=1}^{n} \frac{L_i}{c} \tag{3.12}$$

$$\sigma_d = \sigma_d^{(1)} + \sigma_d^{(2)} = \sum_{i=1}^{n} \frac{t_{wait}}{\tau_i} \tag{3.13}$$

This will translate in the following algebras for the waiting time t_{wait} : $\left(\mathbb{R}^+ \cup \infty, \leq, \mathbb{R}^+, \mathbb{R}^+, \infty, \oplus_t, \mathrm{id}_{\mathbb{R}^+}\right)$ and the memory decoherence time and τ : $\left(\mathbb{R}_0^+, \geq, \mathbb{R}^+, \mathbb{R}^+, 0, \oplus_\tau, \mathrm{id}_{\mathbb{R}^+}\right)$, where the binary operations $\oplus_t$ and $\oplus_\sigma$ are defined as follows:

$$\begin{aligned} \oplus_t : \mathbb{R}^+ \times \mathbb{R}^+ &\longrightarrow \mathbb{R}^+ \\ (t_{i:j}, L_{j:k}/c) &\longmapsto t_{i:j} + L_{j:k}/c \end{aligned} \tag{3.14}$$

$$\begin{aligned} \oplus_\tau : \mathbb{R}^+ \times \mathbb{R}^+ &\longrightarrow \mathbb{R}^+ \\ (\tau_{i:j}, \tau_{j:k}) &\longmapsto \left[\frac{1}{\tau_{i:j}} + \frac{1}{\tau_{j:k}}\right]^{-1} \end{aligned} \tag{3.15}$$

Both metrics are monotone and isotone. The formal proof of this is presented in *Section B.2 and B.3*.

3.1.6 Metric for Probability of Success

Given the model presented in *Section 3.1.3* and the scheme of distribution being the two-step scheme presented in *Section 3.1.5*, it is important to analyse how the probabilities come out to play. Since every time something fails, everything repeats from the beginning, then the final distribution of the probability of success will also be a geometric distribution with the parameter of the distribution being the multiplication of all parameters. This comes from the fact that having either simultaneous (*Figure 3.1a*) or sequential (*Figure 3.1c*) geometric processes result in a geometric process:

1. Consider m simultaneous geometric distributions with parameters $a_1, a_2, ..., a_m$ (see *Figure 3.1a*). The probability of at least one element of that list failing one time is given by:

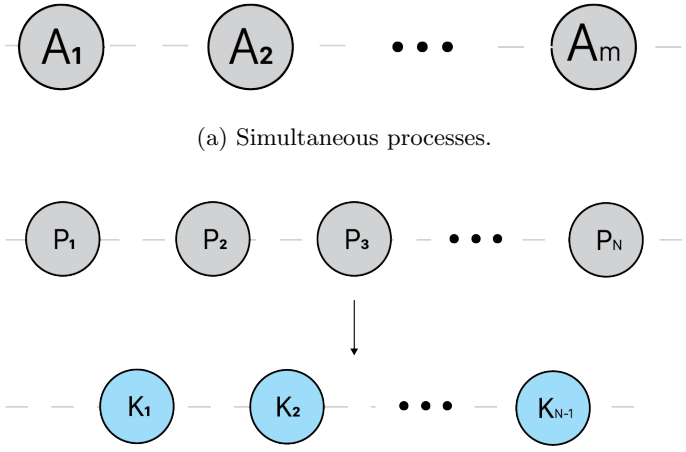
$$\begin{aligned} P(\text{failing one time}) &= (1-a_1)a_2...a_m + a_1(1-a_2)...a_m + a_1a_2...(1-a_m) + \\ &\quad + (1-a_1)(1-a_2)...a_m + ... + a_1(1-a_2)...(1-a_m) + ... \\ &= \sum_{j=1}^{2^m-1} \prod_{k=1}^{m} a_k^{\overline{j_k}}(1-a_k)^{j_k} \\ &= \sum_{j=0}^{2^m-1} \prod_{k=1}^{m} a_k^{\overline{j_k}}(1-a_k)^{j_k} - \prod_{k=1}^{m} a_k \\ &= \prod_{k=1}^{m}(a_k + (1-a_k)) - \prod_{k=1}^{m} a_k \\ &= 1 - \prod_{k=1}^{m} a_k \end{aligned} \tag{3.16}$$

   where $j = j_m j_{m-1} ... j_2 j_1$ is the binary decomposition of j and $\overline{j_k} = 0$ if $j_k = 1$ and $\overline{j_k} = 1$ if $j_k = 0$.

2. Consider n sequential geometric distributions with parameters $b_1, b_2, ..., b_n$ (see *Figure 3.1c*). The probability of one of them failing and having to re-do all the steps from the beginning is:

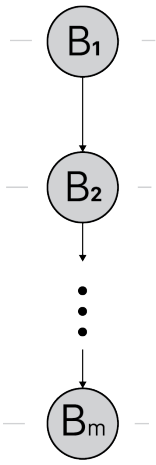
$$\begin{aligned} P(\text{failing one time}) &= (1-b_1) + b_1(1-b_2) + b_1b_2(1-b_3) + ... + b_1b_2...b_{n-1}(1-b_n) \\ &= 1 - \prod_{k=1}^{n} b_k \end{aligned} \tag{3.17}$$

Using theses results, the final probability distribution (see *Figure 3.1b*) will be:



(a) Simultaneous processes.

(b) Two-step scheme with a first round of entanglement generation and a second round of entanglement swapping.



(c) Sequential processes.

Figure 3.1: Picture description of how the different processes are considered.

$$\begin{aligned} P(\text{end-to-end at } t) &= \tilde{p}\cdot\tilde{k}(1-\tilde{p}+\tilde{p}(1-\tilde{k}))^{t-1} = \tilde{p}\cdot\tilde{k}(1-\tilde{p}\cdot\tilde{k})^{t-1} \\ P(\text{end-to-end at } t) &\sim Geom(\tilde{p}\cdot\tilde{k}) \\ P(\text{end-to-end within } t) &= 1-(1-\tilde{p}\cdot\tilde{k})^{t} \end{aligned} \tag{3.18}$$

with $\tilde{p} = \prod_{i=1}^{n} p_i$ and $\tilde{k} = \prod_{i=1}^{m} k_i$ where n is the number of links within a chain and m is the number of intermediary nodes in the chain (in a regular chain, $m = n-1$).

Using the results of *Equation 3.18*, we can fix t to be some value of time, and verify that the properties of the correspondent algebra are maintained (this will be verified in *Section B.4*. The algebra for $t = 1$ is given by $p_{suc} : \Big([0;1), \geq, (0;1)\times(0;1), [0;1), 0, \oplus_p, \mathrm{id}_{[0;1]}\Big)$ where $\oplus_p$ is the following binary operation:

$$\begin{aligned} \oplus_p : [0;1)\times(0;1)^2 &\longrightarrow [0;1) \\ (p_{i:j}, (p_{j:k}, k_j)) &\longmapsto p_{i:j}\cdot p_{j:k}\cdot k_j \end{aligned} \tag{3.19}$$

3.1.7 Overall Metrics for Bipartite Entanglement

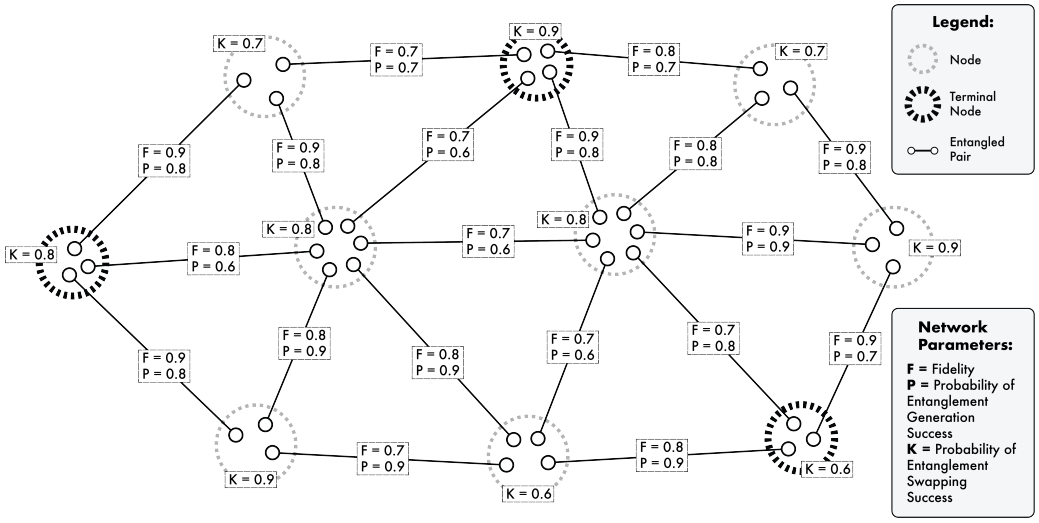
Given the previous sections, let us organize all the found metrics into one simpler vectorial description. Consider all the previously defined algebras:

1. Fidelity (in γ values) - $\gamma : \Big([1/3;1)\cup\{0\}, \geq, (1/3;1), (0;1), 0, \oplus_\gamma, g\Big)$
2. Waiting time - $t_{wait} : \Big(\mathbb{R}^+\cup\infty, \leq, \mathbb{R}^+, \mathbb{R}^+, \infty, \oplus_t, \mathrm{id}_{\mathbb{R}^+}\Big)$
3. Memory decoherence time - $\tau : \Big(\mathbb{R}_0^+, \geq, \mathbb{R}^+, \mathbb{R}^+, 0, \oplus_\tau, \mathrm{id}_{\mathbb{R}^+}\Big)$
4. Probability of success - $p_{suc} : \Big([0;1), \geq, (0;1)\times(0;1), [0;1), 0, \oplus_p, \mathrm{id}_{[0;1]}\Big)$

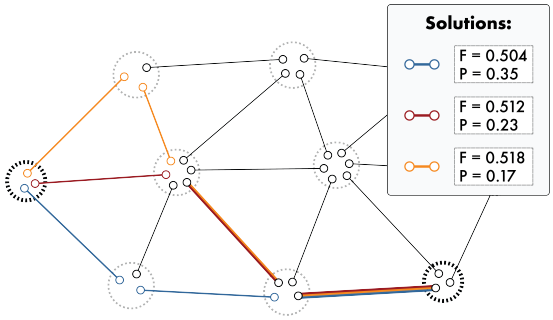
Using this, our vector description of the set of algebras would become: $(\gamma, t_{wait}, \tau, p_{suc})$. Reminding the dominance relation introduced in *Definition 2.3.7*, one path a corresponding to a vector $(\gamma^{(a)}, t_{wait}^{(a)}, \tau^{(a)}, p_{suc}^{(a)})$ dominates a path b corresponding to a vector $(\gamma^{(b)}, t_{wait}^{(b)}, \tau^{(b)}, p_{suc}^{(b)})$ if and only if:

$$
\begin{aligned}
\gamma^{(a)} &\geq \gamma^{(b)} \\
t_{wait}^{(a)} &\leq t_{wait}^{(b)} \\
\tau^{(a)} &\geq \tau^{(b)} \\
p_{suc}^{(a)} &\geq p_{suc}^{(b)}
\end{aligned}
\tag{3.20}
$$

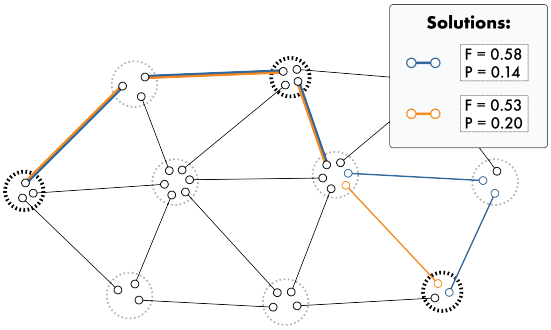
With the strict inequality happening at least for one of these inequalities. Since these algebras do not depend on each other by construction, then each path vector has, in this case, four individually independent weights that do not interfere. In *Figure 3.2a* we present an example of a quantum network where each entangled pair (or network link) is parametrised by a value of fidelity and a probability of entanglement generation success and each node has an associated value for the probability of successful entanglement swapping. In *Figure 3.2b* we present the solution for the multi-objective shortest-path problem taking fidelity and probability of success into account, described by the metrics introduced in the previous sections. Finally, in *Figure 3.2c*, we present the solution for the multi-objective shortest-tree problem, using the same objectives and metrics for the shortest-paths and the 3-GHZ metrics described in the next sections.



(a) Example of a simple quantum network with correspondent parameters. The final path fidelity depends on the independent entangled pairs fidelity and the probability of success depends on the probability of entanglement generation success and of entanglement swapping success.



(b) Solutions for the optimal paths problem



(c) Solution for the shortest tree connecting three terminals, equivalent to the distribution of 3-GHZ state

Figure 3.2: Example of network with parameters correspondent to the fidelity and probability of success

3.2 Distributing GHZ States

When distributing GHZ states and graph states, which can be obtained from GHZ states through LOCC, several protocols have been under recent focus in [25, 26, 55]. In this project, we considered the scheme of distribution present in [25] as the starting base, even though it only considers, as the many counterparts, pure states and deterministic generation of entanglement. We take advantage of the description of the depolarising channel as a way to model the imperfect entanglement, equivalent to using Werner states as we have seen before in *Equation 2.29*, and calculate the effect of the distribution scheme with the depolarising channels on the fidelity of the final GHZ state distributed among the terminals.

3.2.1 Scheme for distribution

The protocol in [25] starts by first solving the Steiner Tree[1] (see *Figure 3.3*) problem using the end-nodes as the terminal nodes. After the Steiner tree connecting the end-nodes is found, the protocol follows a set of sequential LOCC operations to create a star graph across the desired nodes - star expansion protocol (see [25] for a more detailed description). This star graph is LU equivalent to a GHZ state, as seen in the beginning. The problem then becomes finding the best-tree such that, after the star expansion, results in the state that maximises or minimises the imposed metrics. In this section we start by deriving the metric for the fidelity of the final state after the star expansion.

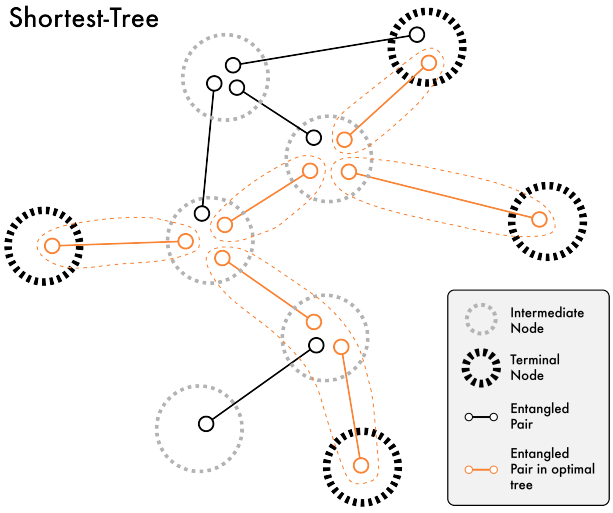


Figure 3.3: Example of a shortest tree (highlighted in orange) connecting a set of 4 terminals in a network. This corresponds to the optimal solution for distributing a 4-GHZ, minimizing the number of entangled pairs used.

The star expansion protocol consists of a set of steps that result in extending entanglement in a star-like form. To perform the necessary calculations in a simpler manner, we adapted this protocol to an identical one that can be understood as merging star-graph states (or GHZ states) across a tree. Since every tree can be modified through LOCC to set of star-graph states (see *Figure 3.6a*) the protocol

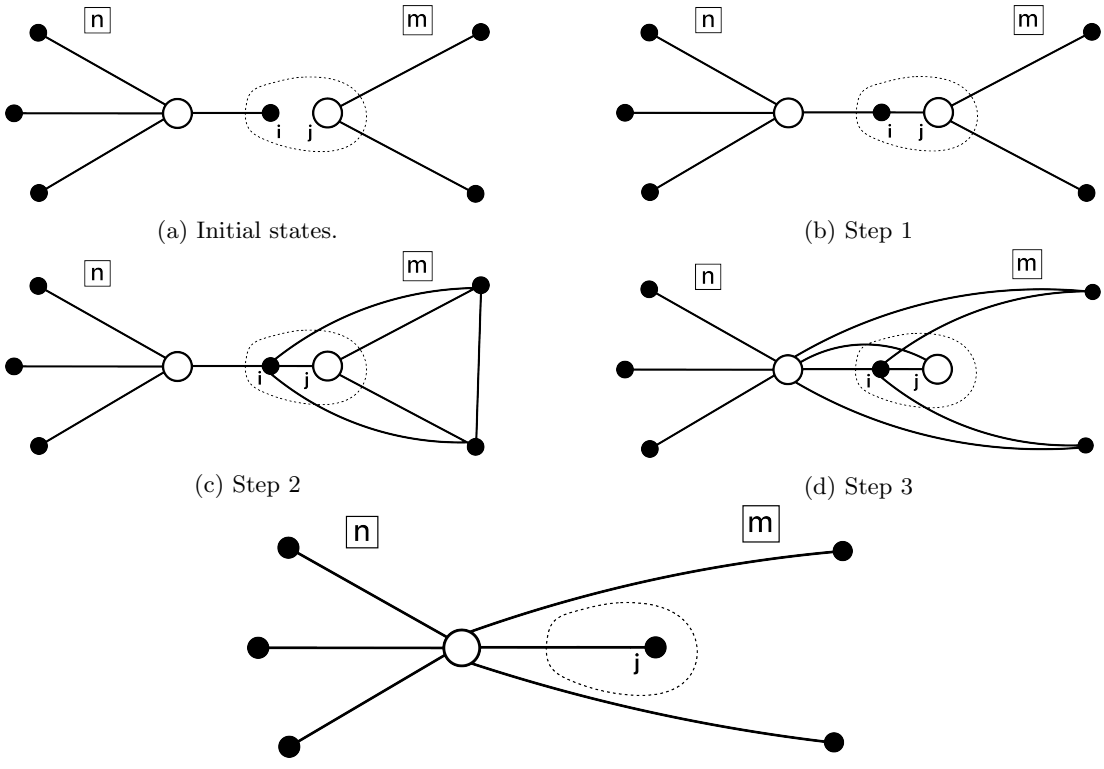
[1] The Steiner tree is the shortest (under some metric) tree like graph connecting a set of nodes called the terminal nodes. A tree graph is a special type of graph that contains no cycle. For example suppose that we start with one node which has n links to n nodes and those n nodes have each some links to other nodes (never connecting to previous nodes, which would be a cycle) and so on, the type of graph this generates is always a tree graph.

merges every GHZ state into another GHZ state connecting every node.

The merge of two GHZ states, not necessarily with the same number of qubits, starting from one n-GHZ[2] state and a m-GHZ state with qubits i and j in the same node (necessary to apply two-qubit gates), goes as follows (see *Figure 3.4*):

1. Perform a controlled-Z operation between qubits i and j (*Figure 3.4b*);
2. Perform the local complementation operation on qubit j (*Figure 3.4c*);
3. Perform the local complementation operation on qubit i (*Figure 3.4d*);
4. Z-measure qubit i (*Figure 3.4e*).

A comparison between the two possible distribution schemes, which are equivalent up to LU, is made in *Figures 3.5 and 3.6*. After performing this set of operations, the final state is also a star-graph state, making it easier to understand why after repeating this protocol throughout the tree will always result in a star-graph state. Some intermediary nodes might not be terminal nodes (these are called Steiner nodes) and while in the star expansion protocol in [25], those nodes are measured before stepping on another node, in this protocol, only for calculation purposes, we only measure them at the end even though the result is the same.



(a) Initial states.

(b) Step 1

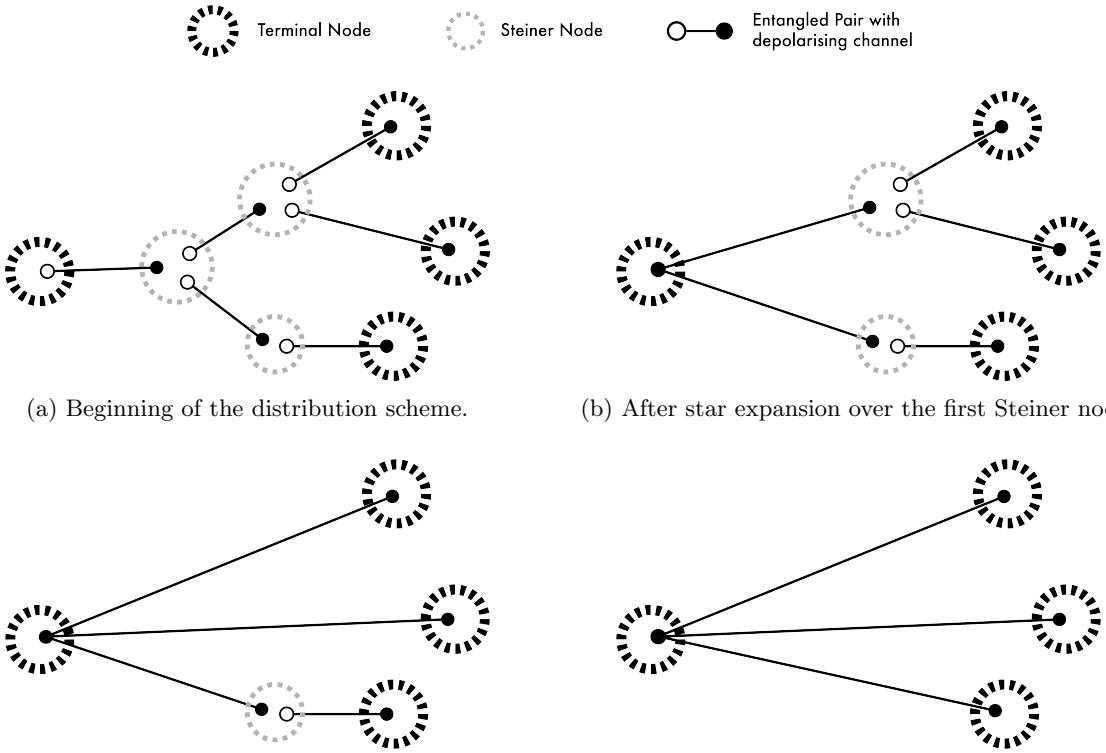
(c) Step 2

(d) Step 3

(e) Step 4

Figure 3.4: Merging steps of a n-GHZ state with a m-GHZ state. Each dot is a qubit, each line represents a relation between the two qubits connected. Since they are graph states, each qubit is in fact in the $|+\rangle$ state and each line represents a CZ gate between both qubits (recall *Section 2.1.4*, namely the graph states subsection). The qubits that are filled represent the ones affected by the depolarising channel.

[2] n-GHZ stands for the GHZ state with n qubits, or, as introduced in the first chapter, the n-Cat state.



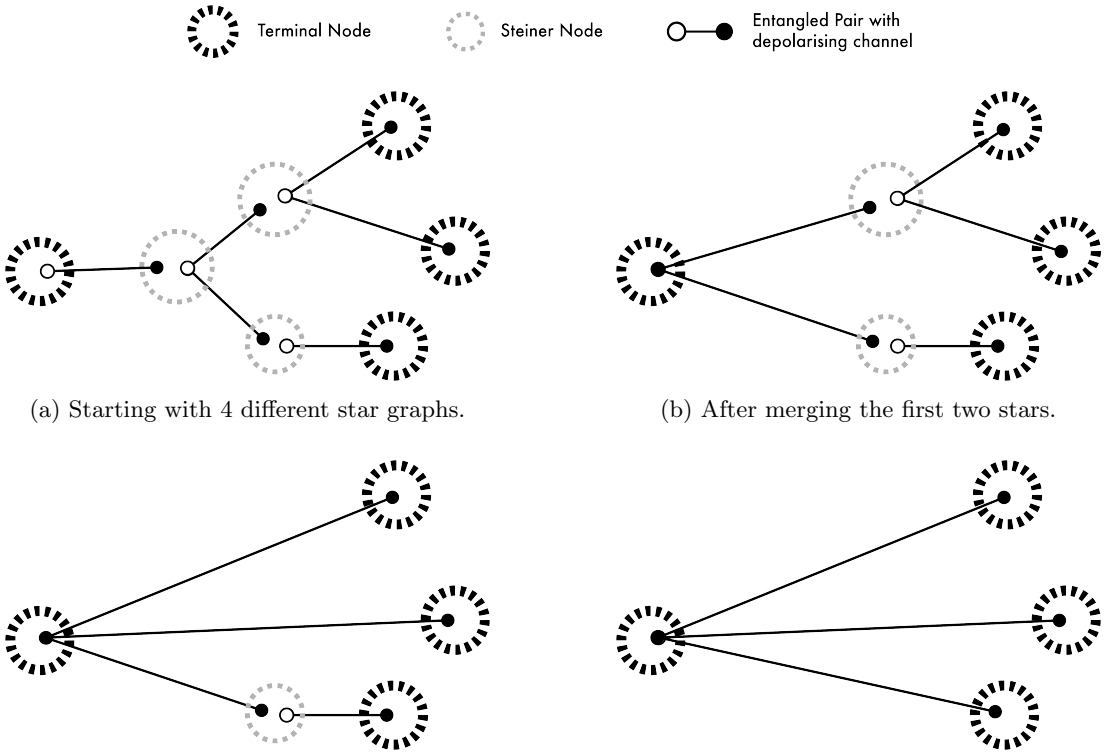
(a) Beginning of the distribution scheme.

(b) After star expansion over the first Steiner node.

(c) After star expansion over the second Steiner node.

(d) After star expansion over the third Steiner node.

Figure 3.5: Usual steps of distribution scheme presented in [25] for a distribution of a 4-GHZ state, consisting of successive applications of the star expansion protocol over the Steiner nodes.



(a) Starting with 4 different star graphs.

(b) After merging the first two stars.

(c) After merging with the third star.

(d) After merging with the fourth star.

Figure 3.6: Steps for distribution considering successive star-graph merges introduced in this section as a simpler way of calculating the effects on the final fidelity.

3.2.2 Mixed States and Fidelity

Finding a metric for the final state fidelity is equivalent to verifying how the star expansion affects the initial state, using depolarising channels to model noise in each entangled pair. Using the approach introduced in the previous section, it is easier to calculate the final form of the GHZ state. Because the controlled-Z operations applied in step 1 do not commute with the action of the depolarising channel, the final form will be slightly different than just having a depolarising channel in each qubit of the GHZ state. In fact, the only terms that change are the non-GHZ terms, *i.e*, if we consider the density matrix to be in one orthonormal basis with GHZ being one of the basis states[3] the density matrix entry correspondent to the GHZ state would always be the same, and in the end the one necessary to calculate the fidelity.

The necessary calculations for the fidelity (and the notation used in this section) are performed in *Section A.1* and result in the final fidelity of the state given by:

$$f = \frac{E(\mathbf{F}_{\mathcal{S}}, \#S) \cdot \prod_{F_{t_i} \in \mathbf{F}_{\mathcal{T}}} \frac{1+2F_{t_i}}{3} + O(\mathbf{F}_{\mathcal{S}}, \#S) \cdot \prod_{F_{t_i} \in \mathbf{F}_{\mathcal{T}}} \frac{2(1-F_{t_i})}{3} + \prod_{F_i \in \mathbf{F}_{\mathcal{T}} \cup \mathbf{F}_{\mathcal{S}}} \frac{4F_i-1}{3}}{2} \tag{3.21}$$

Even though this expression is somewhat complicated, it is important to consider that for the 3-GHZ state, there are no Steiner nodes and the final form is identical to applying just one depolarising channel in each qubit, making *Equation 3.21* become:

$$f = \frac{\prod_{F_{t_i} \in \mathbf{F}_{\mathcal{T}}} \frac{1+2F_{t_i}}{3} + \prod_{F_{t_i} \in \mathbf{F}_{\mathcal{T}}} \frac{2(1-F_{t_i})}{3} + \prod_{F_i \in \mathbf{F}_{\mathcal{T}}} \frac{4F_i-1}{3}}{2} \tag{3.22}$$

3.3 Distributing Arbitrary States

Unlike [25], where all states considered for distribution have necessarily a correspondent graph state (up to LU), we want to generalise for any state possible. For this we take advantage of properties of bipartite entanglement, namely the symmetry in the $|\phi^+\rangle$ pair. We start by proposing a protocol for pure states, equivalently a perfect network, and then analyse the possibility of mixed states.

3.3.1 Arbitrary n-Qubit State

Let us start by defining an arbitrary state. We choose the usual orthonormal basis $\{|0\rangle, |1\rangle\}$ for each qubit. Any n-qubit state can be written as:

$$|\psi\rangle = \alpha_0 |00...00\rangle + \alpha_1 |00...01\rangle + \alpha_2 |00...10\rangle + ... + \alpha_{2^n-1} |11...11\rangle \tag{3.23}$$

subject to $\sqrt{\sum_i |\alpha_i|^2} = 1$ for proper normalisation. These englobes all possible states for n qubits, entangled or not. For simplification purposes, let us rewrite this state in a more compact way:

$$|\psi\rangle = \sum_{m=0}^{2^n-1} \alpha_m |m_{n-1}...m_2 m_1 m_0\rangle \tag{3.24}$$

where $m_{n-1}...m_2 m_1 m_0$ is the binary decomposition with size n of m.

[3] this generalisation is made by applying all combinations of $\{\mathbb{1}, \hat{X}, \hat{Z}, \hat{X}\hat{Z}\}$ to each qubit of the GHZ state but the first

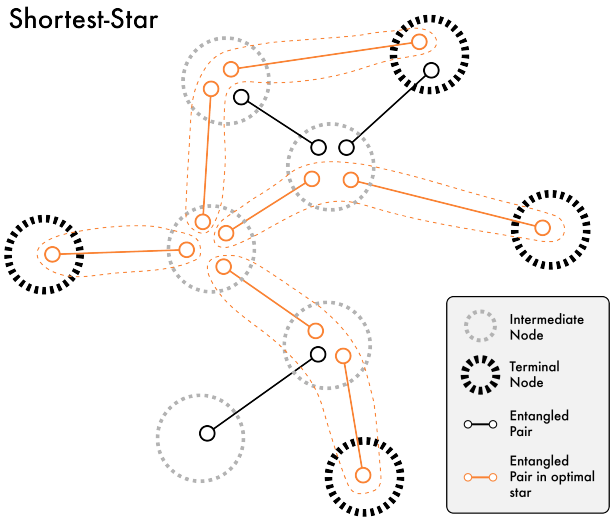


Figure 3.7: Example of a shortest star (highlighted in orange) connecting a set of 4 terminals in a network. This corresponds to the optimal solution for distributing a 4 qubit arbitrary state, minimizing the number of entangled pairs used.

3.3.2 Scheme for Distribution

Consider we want to distribute a n qubit state through a set of n terminals across an arbitrary network. From entanglement generation and entanglement swapping protocols, it is possible to generate a link between any two nodes in a network, so let us consider that our starting point is a star graph (see *Figure 3.7*) connecting the n terminals and a center node (for once, if the center node is one of the terminals, create a virtual node inside the center node and an ancilla[4] qubit). This state can be described as follows:

$$|S\rangle = \bigotimes_{k=1}^{n} |\phi^+\rangle_{k,a_k} \tag{3.25}$$

where a_k are the qubits in the center node, part of the pair shared with the kth terminal. Moreover, we can rewrite this state in a form that separates the qubits present in the center node and the qubits present in the terminals. Because of the symmetry of the $|\phi^+\rangle$ state, this is very simple to do:

$$\begin{aligned} |S\rangle &= \bigotimes_{k=1}^{n} |\phi^+\rangle_{k,a_k} \\ &= \frac{1}{\sqrt{2^n}} \Big[|00...00\rangle_{1,2,..,n} \otimes |00...00\rangle_{a_1,a_2,...,a_n} + \\ &\qquad |00...01\rangle_{1,2,..,n} \otimes |00...01\rangle_{a_1,a_2,...,a_n} + \\ &\qquad\qquad ... \\ &\qquad |11...11\rangle_{1,2,..,n} \otimes |11...11\rangle_{a_1,a_2,...,a_n} \Big] \\ &= \frac{1}{\sqrt{2^n}} \sum_{m=0}^{2^n-1} |m_{n-1}...m_1 m_0\rangle_{1,2,..,n} \otimes |m_{n-1}...m_1 m_0\rangle_{a_1,a_2,...,a_n} \end{aligned} \tag{3.26}$$

Since the only qubits that are in the same node are the ones on the center node, we can do whichever operation we want in the center node, affecting whichever qubits necessary. From the forms of *Equations*

[4] An ancilla qubit is usually used in quantum related things to express an auxiliary qubit that is necessary to make some operations.

3.24 and 3.26, it is simple to see that a projection of the center node qubits in an arbitrary state would create that same arbitrary state throughout the terminal nodes, *i.e*:

$$\begin{aligned}
\langle\psi|_{\mathcal{A}} |S\rangle \langle S| |\psi\rangle_{\mathcal{A}} &= \sum_{m,m'=0}^{2^n-1} \sum_{l,l'=0}^{2^n-1} \frac{\alpha_m^* \cdot \alpha_{m'}}{2^n} \langle m|_{\mathcal{A}} \Big(|l\rangle_{\mathcal{N}} \otimes |l\rangle_{\mathcal{A}} \langle l'|_{\mathcal{N}} \otimes \langle l'|_{\mathcal{A}} \Big) |m'\rangle_{\mathcal{A}} \\
&= \sum_{m,m'=0}^{2^n-1} \sum_{l,l'=0}^{2^n-1} \frac{\alpha_m^* \cdot \alpha_{m'}}{2^n} |l\rangle_{\mathcal{N}} \langle l'|_{\mathcal{N}} \delta_{l,m} \delta_{l',m'} \\
&= \sum_{l=0}^{2^n-1} \sum_{l'=0}^{2^n-1} \frac{\alpha_l^* \cdot \alpha_{l'}}{2^n} |l\rangle_{\mathcal{N}} \langle l'|_{\mathcal{N}} \\
&= \frac{1}{2^n} |\psi\rangle_{\mathcal{N}} \langle\psi|_{\mathcal{N}}
\end{aligned} \tag{3.27}$$

where $\mathcal{N} = 1, 2, ...n$ and $\mathcal{A} = a_1, a_2, ..., a_n$ correspond to the qubits in each terminal and inside the center node, respectively, and for every $k \in \mathbb{N}, |k\rangle = |k_{n-1}...k_1k_0\rangle$ where $k_{n-1}...k_1k_0$ is the binary decomposition of lenght n of k.

If the $|\psi\rangle$ state is part of an orthonormal basis, then the probability of getting the final state across the terminal can be set to 1 by performing corrections depending on the measurement obtained in the center node. This can be regarded as a generalisation of quantum teleportation for n qubits.

3.3.3 Mixed States and Fidelity

Starting from the state in *Equation 3.25*, we can introduce the Werner states by applying depolarising channels on the qubits outside the center node [5], *i.e* the terminal nodes:

$$\mathcal{D}_{\mathcal{N}}\Big(|S\rangle \langle S| , \{F_i\}_{i\in\mathcal{N}} \Big) = \mathcal{D}_{\mathcal{N}}\Big(\frac{1}{2^n} \sum_{m,m'=0}^{2^n-1} |m\rangle_{\mathcal{N}} \otimes |m\rangle_{\mathcal{A}} \langle m'|_{\mathcal{N}} \otimes \langle m'|_{\mathcal{A}} , \{F_i\}_{i\in\mathcal{N}} \Big) \tag{3.28}$$

Using the properties of the depolarising channel we get that:

$$\mathcal{D}_{\mathcal{N}}\Big(|S\rangle \langle S| , \{F_i\}_{i\in\mathcal{N}} \Big) = \frac{1}{2^n} \sum_{m,m'=0}^{2^n-1} \mathcal{D}_{\mathcal{N}}\Big(|m\rangle_{\mathcal{N}} \langle m'|_{\mathcal{N}} , \{F_i\}_{i\in\mathcal{N}} \Big) \otimes |m\rangle_{\mathcal{A}} \langle m'|_{\mathcal{A}} \tag{3.29}$$

From here, projecting the qubits inside the center node in the desired state, following the scheme of distribution presented in the previous section, will result in the desired state distributed amongst the terminal nodes with a depolarising channel applied to each pair applied over the correspondent qubit:

$$\begin{aligned}
\langle\psi|_{\mathcal{A}} \mathcal{D}_{\mathcal{N}}\Big(|S\rangle \langle S| , \{F_i\}_{i\in\mathcal{N}} \Big) |\psi\rangle_{\mathcal{A}} &= \frac{1}{2^n} \sum_{m,m'=0}^{2^n-1} \mathcal{D}_{\mathcal{N}}\Big(|m\rangle_{\mathcal{N}} \langle m'|_{\mathcal{N}} , \{F_i\}_{i\in\mathcal{N}} \Big) \otimes \langle\psi|_{\mathcal{A}} |m\rangle_{\mathcal{A}} \langle m'|_{\mathcal{A}} |\psi\rangle_{\mathcal{A}} \\
&= \frac{1}{2^n} \mathcal{D}_{\mathcal{N}}\Big(|\psi\rangle_{\mathcal{N}} \langle\psi|_{\mathcal{N}} , \{F_i\}_{i\in\mathcal{N}} \Big)
\end{aligned} \tag{3.30}$$

[5] Notice that each of this entangled pairs connecting the center node and each terminal node is not necessarily only one link, but can be a chain of links. However, given the entanglement swapping protocol, we can transform the chain of links into an entangled pair connecting each center node to each terminal node. Even after modelling each individual link as a Werner state, the final entangled pair can also be modelled as a Werner state with a fidelity that will depend on each of its link fidelities. If the change of variable to γs is considered, then the γ value of the entangled pair connecting the center node to the terminal is just the multiplication of all the γ values of each link constituting the chain connecting the two nodes.

To find the metric translates to the simpler problem of finding the completely mixed state [6] using the value of the network fidelity for each link and calculating its fidelity. In the case of a m-qubit GHZ state, the final fidelity is calculated in *Section A.2, Equation A.19* to be:

$$\begin{aligned} f &= \frac{\prod_{i=1}^{m} \frac{1+2F_i}{3} + \prod_{i=1}^{m} \frac{2(1-F_i)}{3} + \prod_{i=1}^{m} \frac{4F_i-1}{3}}{2} \\ f &= \frac{\prod_{i=1}^{m} \frac{1+\gamma_i}{2} + \prod_{i=1}^{m} \frac{1-\gamma_i}{2} + \prod_{i=1}^{m} \gamma_i}{2} \end{aligned} \tag{3.31}$$

The correspondent algebra for trees (as we will introduce in *Chapter 4*, but for now regard it as an usual algebra for routing) will be given by $f_{GHZ} : \left([1/2; 1) \cup \{0\}, \geq, (1/2; 1), (0; 1)^3, 0, \oplus_{GHZ}, h\right)$ where $\oplus_{GHZ}$ is the following binary operation:

$$\begin{aligned} \oplus_{GHZ} : (0;1)^3 \times (1/2;1) &\longrightarrow (0;1)^3 \\ (\{a,b,c\}, f_{i:j}) &\longmapsto \{a \cdot \frac{1+2f_{i:j}}{3}, b \cdot \frac{2(1-f_{i:j})}{3}, c \cdot \frac{4f_{i:j}-1}{3}\} \end{aligned} \tag{3.32}$$

And $h(\cdot)$ is the following function:

$$h(\{a,b,c\}) = \begin{cases} \frac{a+b+c}{2} & , \text{ if } \frac{a+b+c}{2} \geq 1/2 \\ 0 & , \text{ if } \frac{a+b+c}{2} < 1/2 \end{cases} \tag{3.33}$$

This algebra is monotone, but not isotonic. However, as we will see in *Chapter 4*, there exists a special property for this algebras that is present - path-isotonicity. The proof of all these properties can be found in *Section B.5*.

Memory Decoherence Factor

In *Section 3.1.5*, the decoherence factor was introduced, depending on two different metrics: the memory decoherence time and the waiting time. This factor, despite depending on these two metrics, will only affect the fidelity of the state. As seen in *Equation 3.1*, it will appear as an additional factor to the γ values and therefore can be incorporated in the above mentioned fidelity metric for distributing arbitrary states. This means that after finding the non-dominated set of solutions for a path: $X_d = \{(\gamma_1, t_1, \tau_1), ...(\gamma_n, t_n, \tau_n)\}$, the solution can be restrained to only one value, which is:

$$\max_{\nu \in X_d} \gamma_\nu \cdot e^{-\frac{t_\nu}{\tau_\nu}} = \max_{i=1,\dots,n} \gamma_i \cdot e^{-\frac{t_i}{\tau_i}} \tag{3.34}$$

This reduction of objectives can be crucial for reducing the complexity of our star-algorithm, as we will see when we introduce the algorithm in *Chapter 4*. It is important to notice that this reduction can only be made under the following assumption (or approximation): the center node coordinates each path, taking into account its waiting time, so the bipartite entanglement distribution of every path finishes at the same time, allowing the arbitrary state to be distributed as soon as all entangled pairs between the center node and the terminals are generated. If this is not possible, then the metric will depend on the difference between the biggest waiting time and all the others and the reduction can not be made for

[6] We say completely because every qubit is in a mixed state with a given fidelity.

the waiting time metric, *i.e* the metric must remain separate from the fidelity. These two situation are equivalent to the following reductions, respectively:

$$\begin{aligned}(\gamma, t_{wait}, \tau, p_{suc}) &\longmapsto (\gamma \cdot e^{-\frac{t_{wait}}{\tau}}, p_{suc}) \\ (\gamma, t_{wait}, \tau, p_{suc}) &\longmapsto (\gamma \cdot e^{-\frac{t_{wait}}{\tau}}, t_{wait}, p_{suc})\end{aligned} \tag{3.35}$$

3.3.4 Probability of Success Metric for Distributing Arbitrary States

Identically to what has been done in *Section 3.1.6*, we can consider that each path has geometric distribution for the probability of success in achieving end-to-end bipartite entanglement and model the probability of the final projection succeeding when distributing using the star scheme as another geometric distribution. As before, if everything is distributed geometrically, then the joint distribution is also geometric ruled by the product of every individual parameter. This guarantees the same exact properties as before, adding a new parameter depending on the center node that models the final projection (which can be 1 if the final projection is deterministic, *i.e* there exists a set of corrections that can always retrieve the desired state).

3.4 End-of-Chapter Remarks

In this chapter we first went over characterising the routing objectives for bipartite entanglement distribution to enable a differentiation between paths and trees (which we will clarify in the next chapter). We found four main objectives and described in depth their correspondent algebras (equivalently metrics), proving some important properties. Concluding the bipartite entanglement, we introduced one scheme for distributing GHZ states (and eventually graph states) from the literature and introduced a new trivial scheme for distributing arbitrary states. For each of this schemes we analysed how each individual entangled pair used to distribute the multipartite state would affect the fidelity of the final state, which is crucial to define a correspondent metric for the fidelity and is one the main goals of this project. We analysed the GHZ state, and furthermore provide a description compatible with more states, which is also described in [33].

Chapter 4

Algorithms for Optimal Distribution of Multipartite Entanglement

In the previous chapter both the protocols for creating the desired states and some corresponding metrics have been explored. In this chapter, the best way to distribute the desired states is found by implementing an algorithm and finding the properties that guarantee that the algorithm provides the set of optimal solutions (in the sense of Pareto optimality, as introduced in *Definition 2.3.8*). This approach can be expanded to more parameters that define entanglement distribution if necessary, as long as the properties seen throughout this chapter are verified, making this approach a systematic approach for the problem of finding the best way of distributing some type of multipartite entanglement.

We do this by first introducing a new concept, in line with the algebra for routing introduced in *Section 2.3*, called algebra for trees. As the algebra for routing defined the framework for each parameter when trying to find the best path, the algebra for trees will have a similar role when trying to find the best tree.

After defining this new algebra for trees and some new interesting properties and conditions, we introduce two different algorithms, one based on the GHZ distribution scheme from *Section 3.2* and another one based on the arbitrary state distribution from *Section 3.3*. These algorithms take advantage of the properties of these algebras, as well as of the algorithm for the multi-objective shortest-path problem (MOSP) from *Chapter 2*.

4.1 Algebra for Trees

Given the definition of algebra for routing introduced in *Chapter 2*, an extension for trees can be made in a trivial manner, where instead of edges (described by labels) we have paths and instead of paths (described by signatures) we have trees. Just as every edge is a path, we have that every path is also a tree, the set of inclusions is Edges $\subset$ Paths $\subset$ Trees. The definition comes as follows:

Definition 4.1.1. Algebra for Trees is an ordered septet $(W, \preceq, \Sigma, \Xi, \phi, \oplus, f)$ comprised as follows: W a set of weights, $\preceq$ a total order, Σ a set of labels, Ξ a set of signatures, ϕ a special signature, $\oplus$ a binary

operation that maps pairs of labels and signatures into a signature and a function f that maps signatures into weights.

This allows to take advantage of the definitions of monotonicity, isotonicity, multi-objective algebras and dominance exactly as before, but now applied to the new algebras for trees. Some extra definitions are also of special interest for our problem, namely to take into account the fidelity constraints of a minimum fidelity and a proper extension from protocol to distribute end-to-end bipartite entanglement to protocol to distribute multipartite entanglement.

Another important definition to introduce is the definition of truncated algebras, which we have already came across in *Section 3.1.4*. A truncated algebra is helpful for defining algebras that have thresholds, the reason behind why the algebra for the fidelity is a truncated algebra, and making speed-ups in intermediary steps of the algorithm.

Definition 4.1.2. (Truncated Algebra for Routing) Considering the regular algebra for routing $(W, \preceq, L, \Sigma, \phi, \oplus, f)$, the truncated algebra for routing is a septet $(W, \preceq, L, \Sigma, \phi, \oplus, \tilde{f})$ where, for $\omega \in \Sigma$:

$$\tilde{f}(\omega) := \begin{cases} f(\omega) & \text{for } f(\omega) \preceq \sigma_{trunc} \\ \phi & \text{for } f(\omega) \succ \sigma_{trunc} \end{cases} \tag{4.1}$$

As one can see from the definition, it is just a modification on the function that attributes a weight to each signature, wether it is a path or a tree.

To realise the distribution of multipartite entanglement, some schemes might use different protocols to produce bipartite entanglement in intermediary steps than the ones used to produce multipartite entanglement, as we will see further ahead. For this reason, a new form of isotonicity is necessary to guarantee that the two protocols are compatible with each other. This new form of isotonicity is what we call path-isotonicity and is given by the following definition:

Definition 4.1.3. (Path-Isotonicity) An algebra for trees $(W, \preceq, \Sigma, \Xi, \phi, \oplus, f)$ is said to be path-isotone if:

$$\forall\ \sigma_1, \sigma_2 \in \Sigma, t \in \Xi : \sigma_1 \preceq \sigma_2 \Rightarrow f(t \oplus \sigma_1) \preceq f(t \oplus \sigma_2)$$

One additional definition, for the sake of coherence, is that of possible tree, in the same way as the possible path is defined in *Definition 2.3.9*.

Definition 4.1.4. (Possible Tree) let T be a tree represented by $t \in \{\Xi_i\}$. Similarly to a possible path, t is said to be a possible tree if $\forall\ i = 1, ..., k : t^i \neq \phi^i$.

From the top, given the definition of these new algebras for trees and the usual definitions of monotonicity and isotonicity of *Section 2.3.1*, existence conditions for a shortest-tree or shortest-star can be made. More than that, some truncations, once possible solutions are found, can be made to refine the search space, which results in algorithm speed-ups. In *Theorem 4.1.1*, upper-bounds (in the sense of

the best possible tree) for the shortest-tree are defined, which comes with the consequence of finding a necessary (not sufficient though) condition for the existence of the shortest-tree in *Lemma 4.1.2*.

Theorem 4.1.1. *Under a monotone algebra for trees, any tree* $t \in \Xi$ *connecting a set of* $T \geq 3$ *terminals will follow that,* $\forall\ u \neq v \in terminals : f(\omega(u,v)) \preceq f(t)$ *with* $\omega(u,v) \in \Sigma$ *being the shortest-path connecting the two terminals* u *and* v.

Proof. W.l.o.g consider a tree t with 3 terminals, $a \neq b \neq c$. Now, assume we find the shortest path between a and b, $path_{a:b}$. There are two possible cases for the tree from this point: (1) either $path_{a:b} \in t$ or (2) $path_{a:b} \notin t$. For the first case, since $a \neq b \neq c$, then there is necessarily another path connecting some vertice in $path_{a:b}$ to the vertice c. From the monoticity we get that $f(path_{a:b}) \preceq f(t)$. The second case would imply that $\exists\ \overline{path}_{a:b}$, part of the minimum tree, but in that case $f(path_{ab}) \preceq f(\overline{path}_{a:b}) \preceq f(t)$. Doing this for every path connecting two nodes of the tree, the proof is completed for the 3-tree. For the more general tree, the same can be done considering that now instead of adding paths, we add trees. □

Lemma 4.1.2. *Under a monotone algebra, a necessary condition for the existence of a shortest-tree, distributed across a graph* $G(V,E)$ *with terminals* $\in V$ *is that:* $\forall\ u \neq v \in terminals : f(\omega(u,v)) \neq \phi$.

Proof. Follows from *Theorem 4.1.1*. □

Remark. Theorem 4.1.1 and *Lemma 4.1.2* also apply for the multi-objective set of algebras for trees, using individual monotonicity, the dominance relation and the set of Pareto Optimal paths.

Because of *Theorem 4.1.1*, some modifications can be made in intermediary steps of the algorithm that take advantage of the truncated algebras, further refining the possible solutions and discarding solutions worse than the ones already found. Consider that, while an algorithm is finding the shortest-tree, a set of non-dominated solutions is found $\mathcal{A} = \{T_1, T_2, ..., T_p\}$. Because of this, the algebra for finding solutions can be truncated using the maximum/minimum values for the objectives of the trees in $\mathcal{A}$, meaning that we are not interested in finding solutions that are worse than every possible solution already found. The maximum/minimum is decided based on the total order $\preceq$ of each objective. If each tree T_j is described by a set of signatures $\theta_j = \{\theta_j^1, ..., \theta_j^k\}$ where k is the number of objectives, then for each objective i, there is a ordering:

$$f^i(\theta^i_{j_1}) \preceq^i f^i(\theta^i_{j_2}) \preceq^i ... \preceq^i f^i(\theta^i_{j_p}) \tag{4.2}$$

Then, the value for $trunc^i = \{\max/\min_{tree \in C} f^i(\theta^i)$ is the right side value of *Equation 4.2*, *i.e* the worst possible solution. This way, when finding a tree that is worse than the truncation tupple $trunc = \{trunc^i\}$, there exists necessarily a tree in the set of solutions $\mathcal{A}$ that already dominates such tree.

4.2 Steiner Tree Algorithm

Finding the multi-objective Steiner tree connecting a set of terminals is the solution of the first scheme presented in *Section 3.2*. The Steiner tree solves the problem of finding the smallest weight tree graph that spans a set of terminals, including Steiner nodes which are the intermediary necessary nodes to

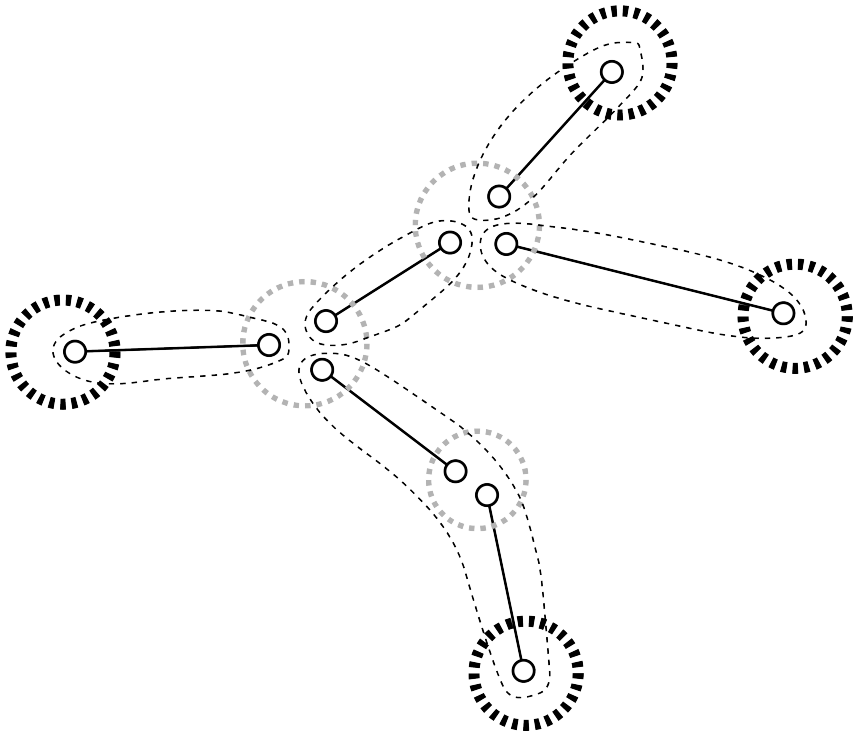


Figure 4.1: Decomposition of a tree in several branches.

connect the terminals. This problem exact solution is generally NP-Hard [56, 57], and our particular case, with the multi-objective problem, only creates more difficulties.

To create an algorithm that is capable of finding the shortest-tree connecting a set of terminals, because of the natural extension between algebra for paths and algebra for trees, we tried to adapt the initial multi-objective shortest-path *Algorithm 1* in *Section 2.3.3* to find the multi-objective shortest-tree. The key point this time is to define which trees are comparable and which are not, which also takes some intuition from the dominance relation. We arrived at the conclusion that only trees connecting the same set of terminals are comparable, *i.e* while visiting some node, only trees connecting the same set of terminals can be compared, like in the dominance relation only trees that are worse for every objective can be discarded.

This way, the dominance relationship is slightly modified in *Algorithm 2*, by imposing that trees not connecting the same set of terminals can not dominate each other. Since the problem itself is computationally extensive, a approximated approach can be made imposing a stop condition as soon as some tree connecting all terminals is found. We used a computational structure to describe the trees separating them by the different branches they are constituted, *i.e* the paths connecting points where the paths divide (an example of these branches can be found in *Figure 4.1*). This structure allows the verification of comparable trees to be easier while allowing to distinguish different protocols for creating paths, which might be necessary depending on the distribution schemes.

Algorithm 2 (k)-Multi-Objective Steiner Tree

procedure STEINER-TREE(*terminal*) ▷ Finds the set of non-dominated trees connecting terminal
    $Nodes$:= Set of nodes u of the network, each with underlying list of trees $Trees_u$ initialised as empty;
    A := Set of visited nodes of the network initialised as empty;
    B := Set of nodes to visit ordered as a priority queue data structure, with priority defined by the dominance relation;
    C := Set of non-dominated trees connecting all terminals;
    for $node \in terminal$ **do**
        Initialise $node \leftarrow \{e_{W_i}\}$; ▷ $\{e_{W_i}\}$ are the neutral elements of $(\Xi_i, \oplus_i)$
        Initialise number of terminals connected $node \leftarrow 1$;
        Add $node$ to B;
    while $B \neq$ empty **do**
        $node \leftarrow$ Top(B) and Remove $node$ from A and B;
        for $v \in$ neighbours($node$) **do**
            $Trees^{add} \leftarrow$ possible trees from $\{Trees^{(i)}_{node} \oplus Edge(node, v)\}$;
            if $Trees_v =$ empty **then**
                $Trees_v \leftarrow Trees^{add}$;
                Add v to B;
            if $Trees_v \neq$ empty **then**
                $Trees^p \leftarrow$ Possible non-dominated trees from $\{Trees^{add^{(i)}} \cup Trees^{(j)}_v\}_{i,j}$; ▷ Must be trees!
                if $\exists$ $\{Trees^k\}$ in $Trees^p$ that connect all terminals **then**
                    Add $\{Trees^k\}$ to C and remove dominated ones from C;
                    $trunc = \{\max/\min_{tree \in C} f^i(\omega^i)\}$; ▷ the max/min is defined by the order
                    Define $trunc$ has the truncation weights;
                if $Trees^p \neq Trees_v$ **then**
                    $Trees_v \leftarrow Trees^p$
                    if $v \in$ A **then** ▷ To visit
                        Update v in B;
                    if $v \notin$ A **then** ▷ To Revisit
                        Add v to A and B;

4.3 Star Algorithm

In *Section 3.3* we introduced a distribution scheme for arbitrary multipartite states and the correspondent algebras. Using that, an algorithm that finds the best way to distribute the desired state in a quantum network can be implemented. Since a star graph is always a tree, we can use the definition of algebra for trees to find the necessary properties of the algebra that guarantee the optimal solutions of the algorithm.

First, let us prove that if the algebra for trees containing the metric correspondent to the star scheme distribution is path-isotonic, then the shortest-star (the star corresponding to the best way of distribution) contains all the shortest-paths between the center node and the terminal nodes.

Proposition 4.3.1. *For the shortest-star with n terminals, the paths connecting the center node and the terminals must be the shortest-paths, if the underlying algebra for trees is path-isotone.*

Proof. Consider that the center node is connected by n paths (this does not happen if the center node is one of the terminals, but for that case consider the star composed of $n-1$ paths), indexed by a number between 1 and n: $path_1, path_2, ...path_n \in \Sigma$. Each path is connected to one of the n terminals. Fix all

paths but $path_1$. Let $t \in \Xi$ be correspondent to the tree formed by $path_2 \cup path_3 \cup ... \cup path_n$. Now consider there $\exists\ \overline{path_1} : \overline{path_1} \preceq path_1$, due to path-isotonicity of the algebra for trees, then if $\overline{path_1} \preceq path_1 \Rightarrow f(t \oplus \overline{path_1}) \preceq f(t \oplus path_1)$ and the shortest tree would be $\overline{path_1} \cup path_2 \cup path_3 \cup ... \cup path_n$. Doing this for every other path, we get that the shortest-star is the one with every branch being the shortest-path between the center node and the terminals. □

Remark. The same applies for the multi-objective shortest-star problem, with the paths being the set of Pareto-Optimal paths and requiring that every algebra for trees is path-isotone.

Proof. Consider that $path_1 \notin X_1$ where X_1 is the set of Pareto-Optimal paths between the node 1 and the center node. Then, $\exists\ \overline{path_1}$ such that $\overline{path_1}$ D $path_1$ and from here the star containing $\overline{path_1}$ is better than the one containing $path_1$. The rest of the proof is identical to the proof of *Theorem 4.3.1*. □

The star-algorithm is very easy to understand: first it finds the shortest-path between each terminal and every other node using *Algorithm 1* and then creates all possible stars, only choosing the non-dominated ones for the set of solutions and constantly updating the set of solutions if any new solution is found and a previous one must be discarded. Some speed-ups are performed in several stages of the algorithm, derived from properties of the algebras.

Algorithm 3 Exact Algorithm for Star Graph

```
1:  procedure T-Star Exact(terminal)
2:      A := Set of possible star graphs, ordered as a priority queue, with priority defined by the dominance relation;
3:      trunc = {φ_i};
4:      for node ∈ terminal do
5:          procedure Shortest-Path(node) subject to trunc;
6:          for Tree T = ∪_{i,j} Path^j(node, terminal_i) do
7:              if T is possible and non-dominated by any tree in A then
8:                  Add T to A;
9:          trunc = {max/min_{tree∈A} f^i(ω^i)};          ▷ the max/min is defined by the order
10:         Erase all paths from node bigger than trunc;
11:     Nodes_reach ← Choose the set of nodes reachable from every terminal;
12:     for node ∈ Nodes_reach do
13:         for Tree T = ∪_{i,j} Path^j(node, terminal_i) do
14:             if T is possible and non-dominated by any tree in A then
15:                 Add T to A;
```

This algorithm searches the possible shortest-stars connecting the set of terminals after finding the best path between each terminal and all other terminals. Some key advantages of this algorithm are:

1. If there is no connection between some pair of terminals, it is noticed early in the algorithm and, because of *Lemma 4.1.2*, there is no possible solution which terminates the algorithm;

2. As soon as the algorithm starts finding paths, it creates lower bounds on the possible trees. If there is some solution in which the terminal node is the center node, it may refine the upcoming path searches, by excluding paths that would result in worse trees;

3. If there are physical constraints such as no link can be used twice, they are easy to implement by discarding such options. However, if this is only done *a posteriori*, for a number of terminals bigger than three, then we can only retrieve at least a part of the optimal solution and divise a strategy that outputs a flag saying if there might exist more solutions or not, by keeping in a separate list the solutions where each link might be used twice;

This algorithm is exact, in the sense that finds the set of Pareto optimal stars. A proof of this is performed in the following proposition.

Proposition 4.3.2. *Under a monotone and isotone algebras for shortest-paths and monotone and path-isotone algebras for trees,* Algorithm 3 *will converge to the set of Pareto optimal stars.*

Proof. Every tree analysed by the algorithm will be constituted by a center node and one of the Pareto optimal paths from the center node to each terminal. From *Proposition 4.3.1* we get that, for each placement of the center node, the trees considered are the optimal trees. As for the truncation used in step 5 of the algorithm, we need to prove that doing this prevents any optimal tree from being discarded.

The first time Shortest-Path algorithm runs, there is no truncation ($trunc = \{\phi_i\} \equiv$ no truncation), so everything is as before. For the second time, a value for the truncation weights is calculated from the max/min values of each objective, making every tree dominated by the truncation weights also dominated by all the stars found so far (see *Equation 4.2*). Now consider we run the algorithm again starting from another node, but subject to the truncation weights. This would imply that any discarded paths would be dominated by the truncation weights and consequently dominated by the shortest trees found so far. As stated in *Theorem 4.1.1*, if there is a path z such that $\forall\ i, trunc^i \preceq^i f^i(z^i)$, than, every tree $\tilde{t}$ with that path would have that $\forall\ i, trunc^i \preceq^i f^i(z^i) \preceq^i f^i(\tilde{t}^i)$ and, therefore, would be dominated by the tree found before, making it not a suitable solution. By doing this, at each shortest-path search, all the trees discarded are not optimal solutions and always dominated by previously found potential solutions.

In the end, all other possible choices for the center node are considered and for each of them, using *Theorem 4.3.1*, the possible non-dominated star-graphs are chosen. □

4.4 Algorithms Comparison

There are some key differences in the algorithms that come mostly from trying to solve two different problems. In the Steiner-tree algorithm, the problem is much more complex than to find the shortest-star, although for three terminals they are the same problem, since every tree connecting three terminals is always a star (the proof of this is very intuitive, just consider all choices of the shortest-tree and realise that all possible choices are, in fact also a star-graph). While a proof of exactness is not presented for the Steiner-tree algorithm, it is at least an approximated algorithm from the construction, while for the Star-algorithm, if the necessary properties are gathered, the algorithm is exact.

There is a big advantage in the Star-algorithm that comes into play when, to conserve the isotonicity property of a complex metric depending on several parameters, the metric is divided in simpler isotone algebras (*e.g* consider the fidelity with the decoherence factor depending on the waiting time and the

quantum memory decoherence time). Every time this decomposition in isotone algebras is done, in the step where we find for every center node, the possible paths to each path, the non-dominated paths can be filtered by returning to the more complex metric and find the only best solution.

One more key distinction is, from the structure of the Steiner tree algorithm, by independently searching paths starting at each terminal, if the terminals are close together, the algorithm runtime will probably decrease significantly.

4.5 Simulations

To illustrate the star-algorithm, we considered a simple GHZ distribution that took into account the fidelity of the final state and the probability of success after all entanglement swapping and bipartite entanglement generation, and implemented the algorithms, both the multi-objective shortest-path and the star-algorithm that depended on it. No simulations can be performed without first finding how the parameters are distributed in the network, which hides in itself an important problem - scaling of the network. In this section we first describe the scaling problem and then present the results of the algorithm for the chosen scalings.

4.5.1 Scaling of the Network

In *Section 2.2* we introduced the concept of structural and functional connectivity and were only concerned with the concept of structural connectivity, from the graph definition. In the next chapter, we introduced several metrics and one of them had a threshold, the fidelity metric (*Section 3.1.4*). This threshold guarantees that entanglement is still present in the final state and if the fidelity drops below such threshold, the path is rendered useless, *i.e* it looses its functionality. When dealing with networks subject to this thresholds, to ensure that we can connect any two nodes of the network, not only structurally, but also functionally, we need to look at the distribution of the parameters across the network.

In our simulations, the only parameter with a threshold is the fidelity. Working under the γ change of variables introduced in *Equation 2.35*, let us assume that the γ value for each edge is distributed according to some probability distribution $P_D(x), x \in [0, 1]$. Assuming that the structural connectivity of the network gives us a value for the network diameter (which is related with the maximum size of the shortest paths connecting nodes in the network), then, ideally, if a network is properly scaled, its functional connectivity should be the same, *i.e* the parameters should be distributed in such a way that paths with the length of the network diameter should also be functionally connected. Alternatively, the inverse interpretation of the scaling problem is to find the size of the connected components based on the distribution of parameters.

To analise the probability distribution of the paths let $\Gamma_i \sim P_D$ correspond to a realisation of a single γ value of one edge indexed by i. Considering a concatenation of edges $\Gamma_1 \cdot \Gamma_2 \cdot ... \cdot \Gamma_n$ where $\Gamma_i \perp\!\!\!\perp \Gamma_j$, $\forall i \neq j$, their joint distribution, *i.e* the γ value distribution of the path is given by the nth product of their distributions. The average value of the distribution is:

$$\langle \Gamma_1 \cdot \Gamma_2 \cdot ... \cdot \Gamma_n \rangle = \langle \Gamma_1 \rangle \cdot \langle \Gamma_2 \rangle \cdot ... \cdot \langle \Gamma_n \rangle \tag{4.3}$$

In our case, we considered that the distribution of the γ values across each edge was a uniform distribution in $[\gamma_{min}, 1]$, with a probability distribution function given by:

$$P_U(x) = \begin{cases} \frac{1}{1-\gamma_{min}} & , \quad \text{if } x \in [\gamma_{min}, 1] \\ 0 \quad , & \text{else} \end{cases} \tag{4.4}$$

The probability of distribution of a product of n independent realisations of a random variable $X \sim U(a, b)$ is obtained in [58]:

$$f_{X^n}(x) = \begin{cases} f_X^k(x), & a^{n-k+1}b^{k-1} \leq x \leq a^{n-k}b^k \\ & k = 1, 2, \dots n \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

$$f_X^k(x) = \sum_{j=0}^{n-k} \frac{(-1)^j}{(b-a)^n(n-1)!} \begin{pmatrix} n \\ j \end{pmatrix} \left(\ln \frac{b^{n-j}a^j}{x} \right)^{n-1} \tag{4.6}$$

Using this result, making $a \to \gamma_{min}$ and $b \to 1$, we can retrieve the probability distribution for the γ value of a path composed of n independent edges. Letting the minimum value of the γ values be a power of γ_{trunc}, the value of the threshold, then we can calculate the probability of $\mathbb{P}\left(\Gamma^k > \gamma_{trunc}\right)$ for the multiple options of $\gamma_{trunc} = \gamma_{min}^n$. The values for every pair of k and n are given, in the limit where $\gamma_{min} \to 1$[1] by:

$$\mathbb{P}\left(\Gamma^k > \gamma_{min}^n\right) = \begin{cases} 1 & , k \leq n \\ 1 - \frac{T(k-1,k-n)}{k!} & , n < k < 2n \\ 1/2 & , k = 2n \\ \frac{T(k-1,n)}{k!} & , 2n < k \end{cases} \tag{4.7}$$

$$\text{where,} \quad T(n,k) = \sum_{j=0}^{k} ((-1)^j \cdot (k-j)^{n+1}) \cdot \binom{n+1}{j} \tag{4.8}$$

This sequence $T(n, k)$ (see sequence A179457 from OEIS) is the sequence that describes the number of permutation trees [2] of power n with width not exceeding k.

This probability restricted to $\mathbb{P}\left(\Gamma^k > \gamma_{min}^n\right) > 10^{-5}$ is plotted in *Figure 4.2*.

Since the value for the last branch in *Equation 4.7* , $\frac{T(k-1,n)}{k!}$ rapidly decreases when extending the path (correspondent to increasing the values of k), a good first approach in the scaling would be to consider that:

$$\gamma_{min}^{d/\alpha} = \gamma_{trunc} \tag{4.9}$$

[1] This is an approximation, validated by the fact that that in the limit of big networks ($n \to \log N \nearrow$) since γ_{trunc} is at least $1/3$, then $1/3^{1/n} \to 1$

[2] A permutation tree is a labeled rooted tree that has vertex set 0,1,2,..,n and root 0, and in which each child is larger than its parent and the children are in ascending order from the left to the right. The power of a permutation tree is the number of descendants of the root. The height of a permutation tree is the number of descendants of the root on the longest chain starting at the root and ending at a leaf. The width of a permutation tree is the number of leafs.

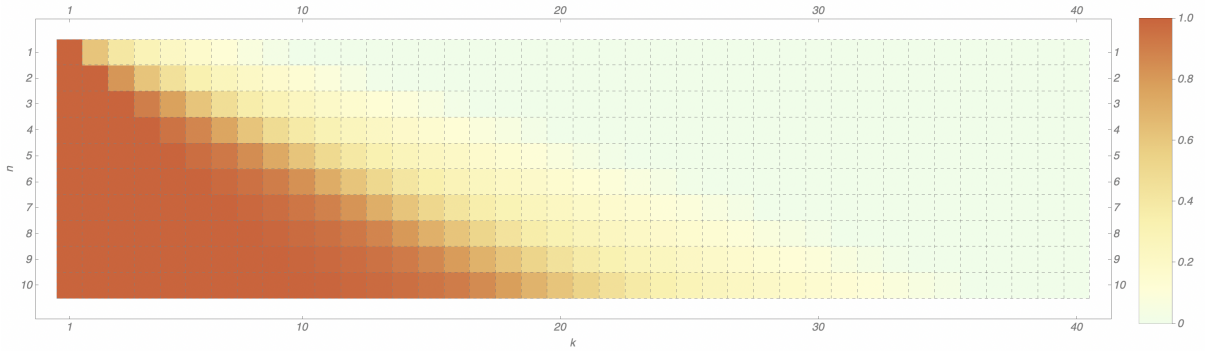


Figure 4.2: Probability of k independent realisations of Γ (*i.e* a path with k edges) having a γ value bigger than $\gamma_{trunc} = \gamma^n_{min}$.

Where d is the network diameter. Before implementing the final algorithms, simulations of the scaling for the network were performed by varying the values of α in *Equation 4.9* by small steps and measuring the number of nodes reached, normalised to the number of nodes of the network, for both an ER network with 1000 and 5000 nodes and a SCL network with 100 and 2500 nodes. The results are demonstrated in *Figure 4.3*.

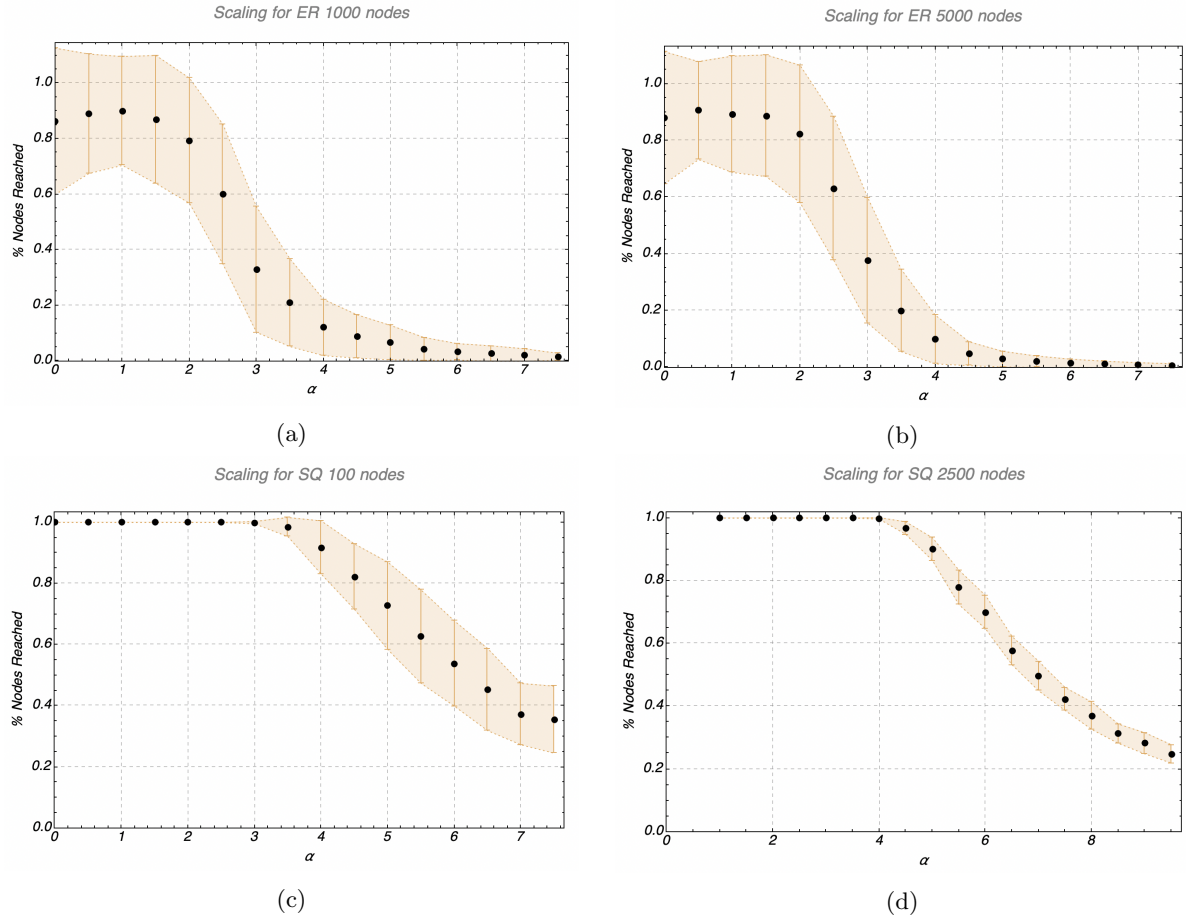


Figure 4.3: Simulations for the scaling in an ER network with (a) 1000 nodes and (b) 5000 nodes and in a SCL network with (c) 100 nodes and (d) 2500 nodes.

It is necessary to note that for the ER network scaling, the value of d considered was $\ln N$ instead of

the usual $\ln N / \ln \lambda$. This means values of α should be corrected by an amount of $\ln \lambda$, which in these simulations were $\lambda = 3$.

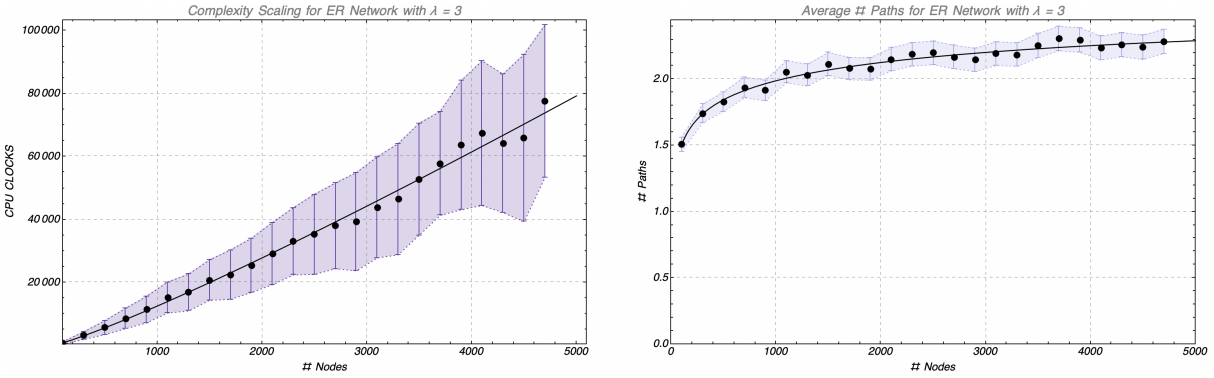
From the figures, we get that for ER networks the scaling does not depend on the number of nodes, which comes in line with the tree-approximation, but for SCL networks, as we increase the number of nodes the α for which the network becomes disconnected starts to increase, which suggests that the scaling with the diameter might not be the best suited. However, since the scaling is a whole problem on its own, we considered that $\alpha \cdot \ln \lambda = 2$ for ER networks and $\alpha = 4$ for SCL networks throughout the simulations in the next section.

4.5.2 Simulations Results

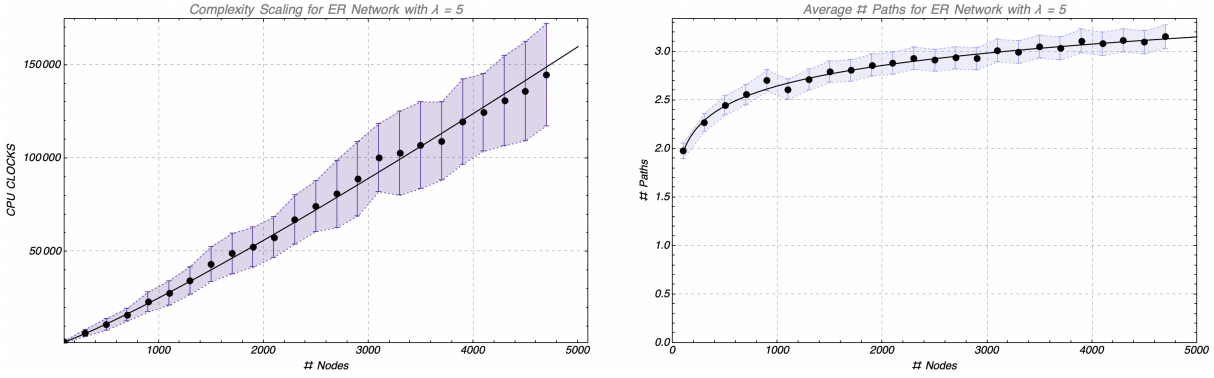
In this section we start by presenting the results for the algorithm for the multi-objective shortest-path problem with routing objectives given by the fidelity represented by the γ variables with the algebras described in *Section 3.1.4* and the probability of success with the correspondent algebras described in *Section 3.1.6*. The scaling of the network is the same described in the previous section. The important parameters necessary for constructing the networks are:

1. Type of network - ER or SCL
2. In the case of ER, the average degree λ
3. The threshold γ value γ_{trunc}
4. The minimum probability of success for entanglement generation p_{min}
5. The minimum probability of success for entanglement swapping k_{min}

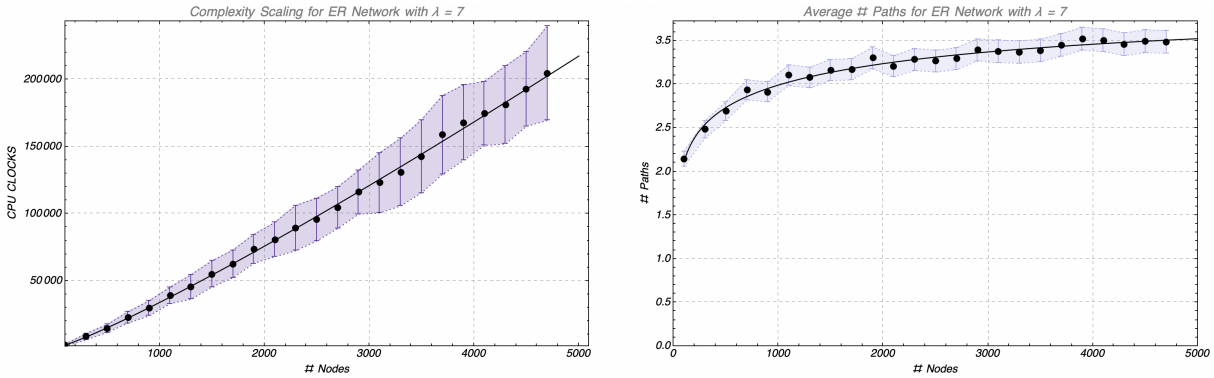
From this point, the simulation code will create 20 different networks generated with the same parameters and for each network will run the multi-objective shortest-path algorithm for 20 different randomly chosen nodes of each network, realising therefore 400 random samples of the algorithm for each set of parameters. In *Figures 4.4a to 4.4c*, the simulations for ER networks, varying in the average degree and in the number of nodes are presented for $\gamma_{trunc} = 0.9$, $p_{min} = k_{min} = 0.9$. In *Figure 4.4d*, the simulations for SCL networks, varying in the number of nodes are presented, again for the same values $\gamma_{trunc} = 0.9$, $p_{min} = k_{min} = 0.9$.



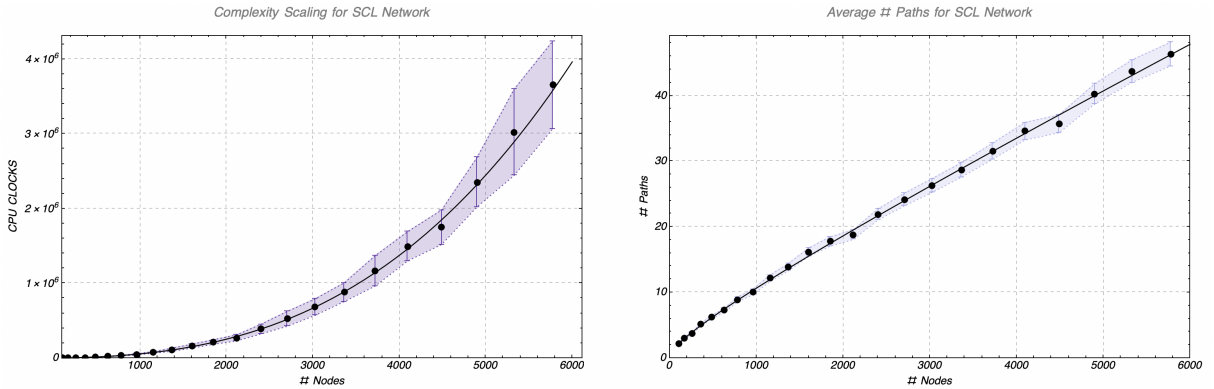
(a) Simulation of MOSP in an ER network with average degree $\lambda = 3$.



(b) Simulation of MOSP in an ER network with average degree $\lambda = 5$.



(c) Simulation of MOSP in an ER network with average degree $\lambda = 7$.



(d) Simulation of MOSP in a SCL network.

Figure 4.4: In the above figures, the left image is always correspondent to do with the complexity of the algorithm (how much time in CPU clocks it takes to run the algorithm) and the right image has to do with the h_{paths} quantity (how many optimal paths per node exist after the algorithm finishes, as it will be better described in *Section 4.6*).

For ER networks, the simulations points for the complexity were fitted to the expression 4.10 and for the SCL network, the simulations points for the complexity were fitted to the expression 4.12. In the case of the quantity h_{paths}, the expressions for the fit were 4.11 for ER networks and 4.13 for SCL networks. Our purpose in making these fits is not to study in detail the behaviour, but to present arguments that this algorithm runs in polynomial time, and almost linearly if we consider ER networks, which are capable of modelling networks in the real world. In *Section 4.6* we will present calculations on the complexity that justify this observed behaviour and find how does the complexity of this algorithm relate with the setup problem.

$$g_{ER}(N) = a_0 + a_1 N \cdot (1 + a_2 \log N)^2 \tag{4.10}$$

$$h_{ER}(N) = \tilde{a}_1 + \tilde{a}_2 \log N \tag{4.11}$$

$$g_{SCL}(N) = b_0 + b_1 N \cdot (1 + b_2 \sqrt{N} + b_3 N)^2 \tag{4.12}$$

$$h_{SCL}(N) = \tilde{b}_1 + \tilde{b}_2 \sqrt{N} + \tilde{b}_3 N \tag{4.13}$$

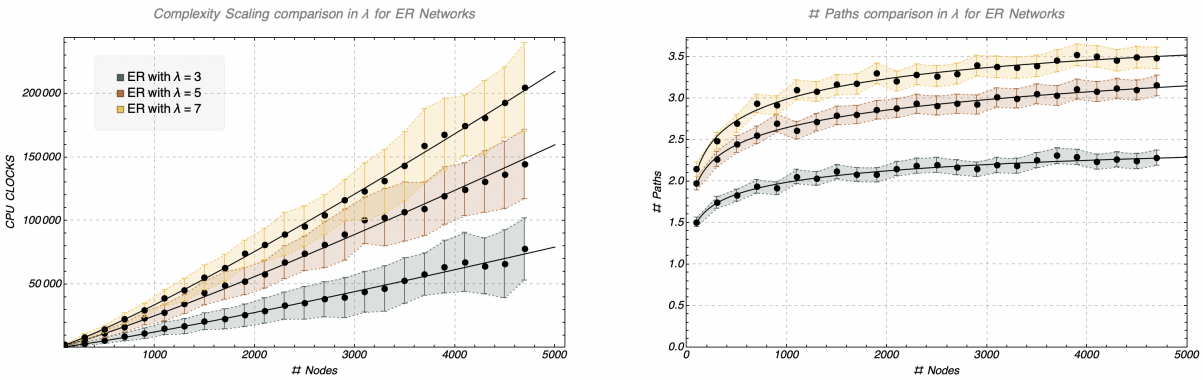


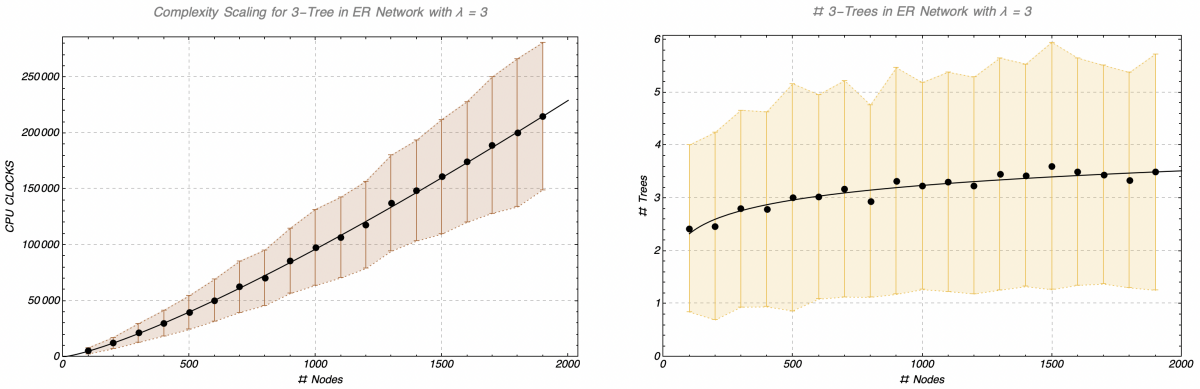
Figure 4.5: Simulations for the MOSP algorithm in an ER network: comparison for different average degrees λ. From the complexity points, we can infer a possible linear dependency on the average degree.

Moving to the Star algorithm, we tested the algorithm for finding first 3-stars, varying the number of nodes and the average degree and then varying the number of terminals. The scaling used is the same scaling as before, and so are the values for $\gamma_{trunc} = 0.9$, $p_{min} = 0.9$ and $k_{min} = 0.9$. While the simulation data from MOSP was fitted to the expressions from 4.10 to 4.13, the fits from the Star-algorithm have been modified: for the complexity in ER networks expression 4.14 was used and 4.15 in the case of SCL networks. Moreover, the multipartite fidelity for the metric used was the one present in *Equation 3.31*.

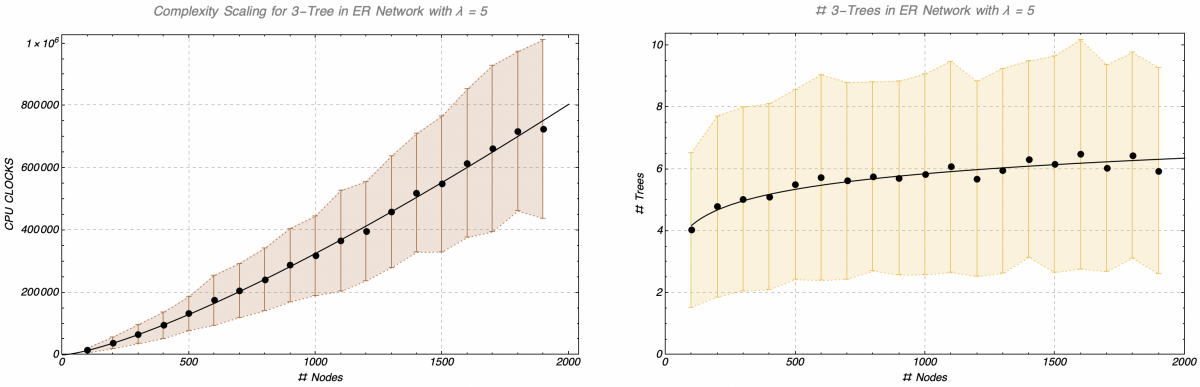
$$\tilde{g}_{ER}(N) = a_0 + a_1 N \cdot (1 + a_2 \log N)^3 \tag{4.14}$$

$$\tilde{g}_{SCL}(N) = b_0 + b_1 N \cdot (1 + b_2 \sqrt{N} + b_3 N)^3 \tag{4.15}$$

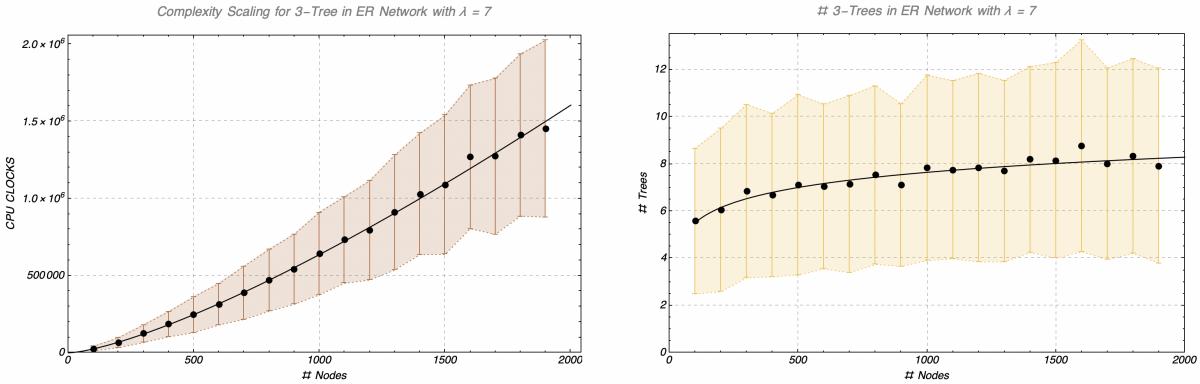
The expressions for the average number of optimal stars is the same as the expression for the average number of optimal paths.



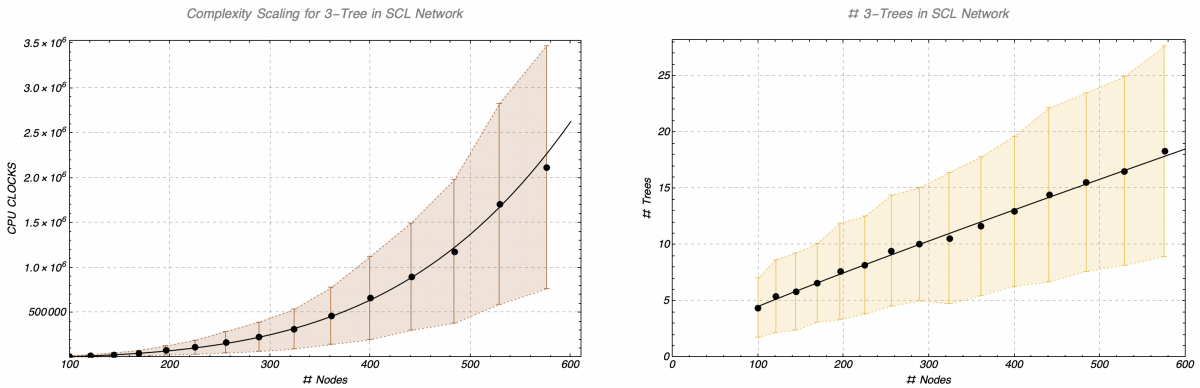
(a) Simulation of the Star algorithm for a 3-star in an ER network with average degree $\lambda = 3$.



(b) Simulation of the Star algorithm for a 3-star in an ER network with average degree $\lambda = 5$.



(c) Simulation of the Star algorithm for a 3-star in an ER network with average degree $\lambda = 7$.



(d) Simulation of the Star algorithm for a 3-star in an SCL network.

Figure 4.6: In the above figures, the left image is always correspondent to the complexity of the algorithm (how much time in CPU clocks it takes to run the algorithm) and the right image has to do with the number of optimal stars found by the algorithm.

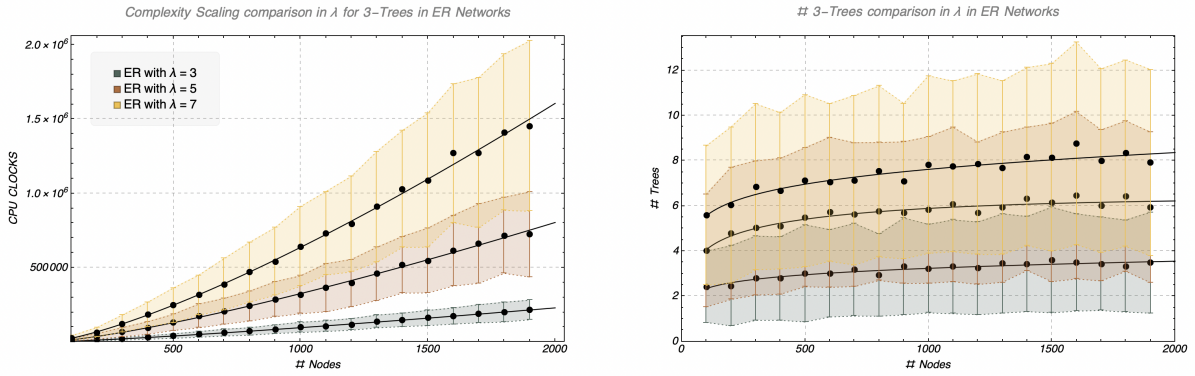
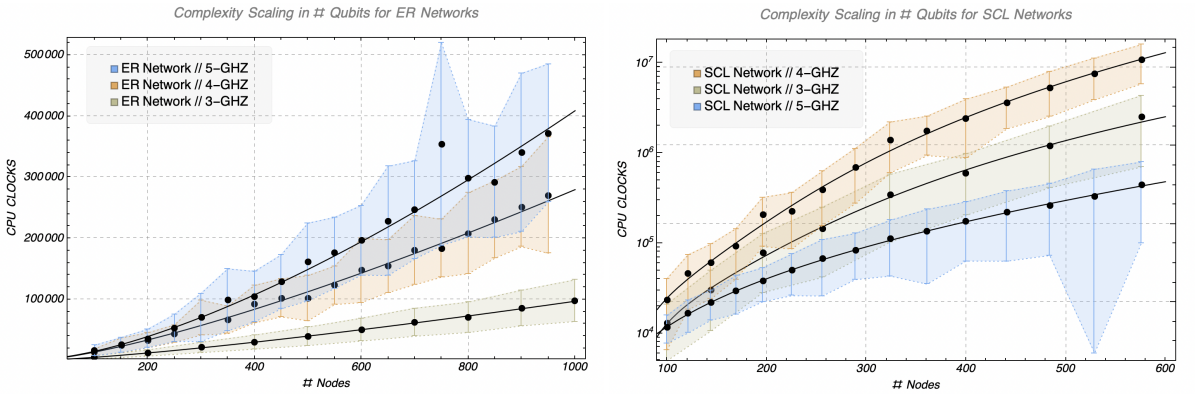


Figure 4.7: Simulations for the Star-algorithm for a 3-Tree in an ER network: comparison in the average degree λ

One important aspect of this algorithm is that, more than just finding the best 3-star (which is always equivalent to finding the best 3-tree), the algorithm is also capable of finding the shortest T-star, with $T \geq 4$. Because of this, using the same metric for multipartite GHZ distribution, we simulated for $T = 4$ and $T = 5$, for both ER networks with $\lambda = 3$ and SCL networks. The results are presented in *Figures 4.8a and 4.8b* and the fit models were again modificated to:

$$\tilde{g}_{ER}(N,T) = a_0 + a_1 N \cdot (1 + a_2 \log N)^T \tag{4.16}$$

$$\tilde{g}_{SCL}(N,T) = b_0 + b_1 N \cdot (1 + b_2\sqrt{N} + b_3 N)^T \tag{4.17}$$

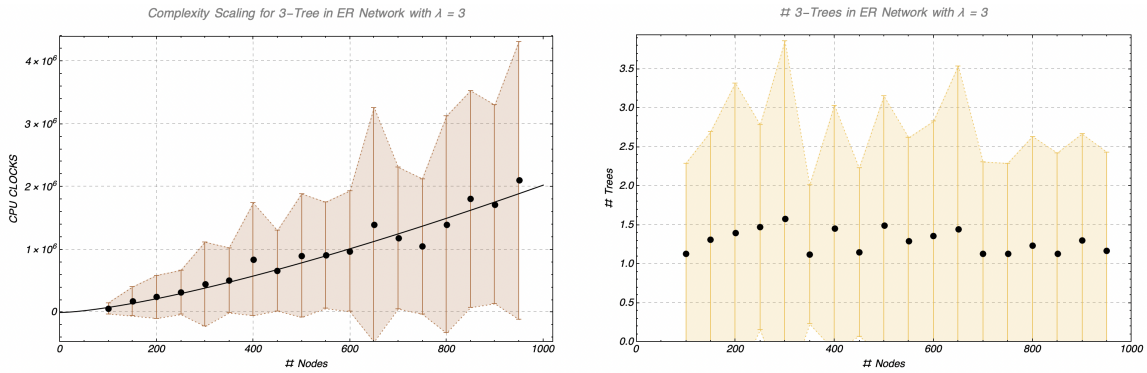


(a) (b)

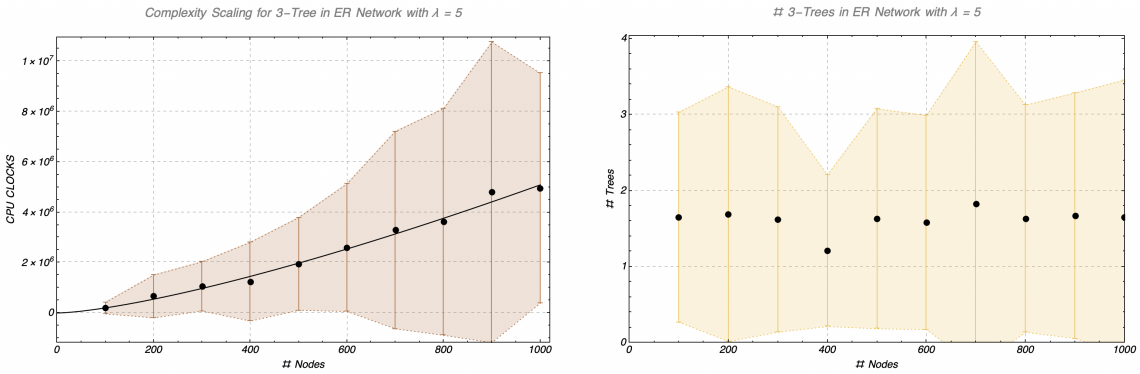
Figure 4.8: Simulations for the Star-algorithm varying the number of terminals. Notice that for SCL networks, when T=5 the complexity is inferior. This is explained by the fact that the simulations where solutions were found are mainly constituted by sets of terminals that are "closer", since when adding one more terminal resulted in sets of empty solutions which data was not considered for these simulations.

While the simulations mainly focus on the Star algorithm, which is exact for any number of terminals, we also present some simulations for the Steiner-tree algorithm presented in *Section 4.2*. There are two striking features one can observe in the simulations: the average complexity is bigger than the one of the star algorithm and the standard deviation is also much larger. The first can be explained by the fact that this algorithm is structured to solve any Steiner tree, no matter how many terminals, which is a

much harder problem than finding the shortest-star. The second can be explained from the fact that, as discussed in *Section 4.4*, if the terminals are very close, which might happen sometimes, the algorithm search space might reduce a lot and a solution can be found in much less time. Given that the terminal choice is completely arbitrary, this, together with the fact that the complexity depends on how close the terminals are, will result in a larger oscillation for the values of complexity.



(a) Simulation of the Steiner Tree algorithm for a 3-tree in an ER network with average degree $\lambda = 3$.



(b) Simulation of the Steiner Tree algorithm for a 3-tree in an ER network with average degree $\lambda = 5$.

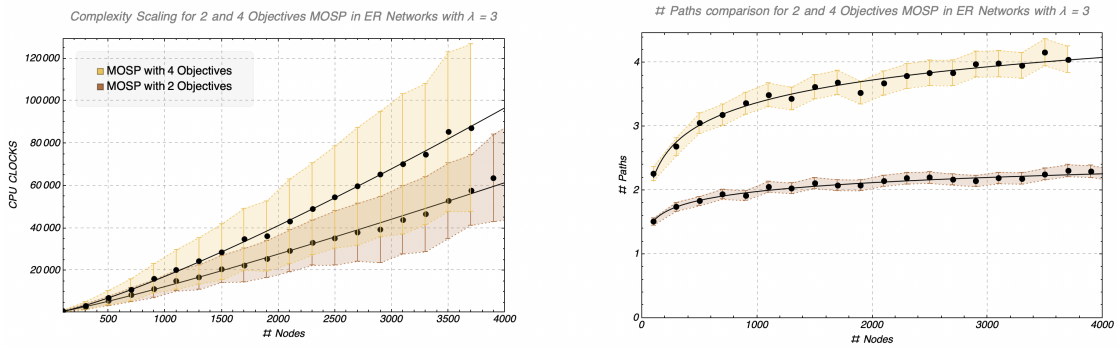
Figure 4.9: In the above figures, the left image is always correspondent to the complexity of the algorithm (how much time in CPU clocks it takes to run the algorithm) and the right image has to do with the number of optimal trees found by the algorithm.

The complexity data was fitted to the same expressions as the ones used for a 3-star in the star algorithm (see *Equation 4.14*). We also verified how the MOSP algorithm would scale by including more objectives. Taking into account our description of four objectives, correspondent to the four algebras described in *Chapter 3* : $(\gamma, t_{wait}, \tau, p_{suc})$, we simulated the algorithm using the same parameters for γ and p_{suc} introduced at the beginning of this section and for the waiting time and memory decoherence time we used the following parameters (the units are made arbitrary):

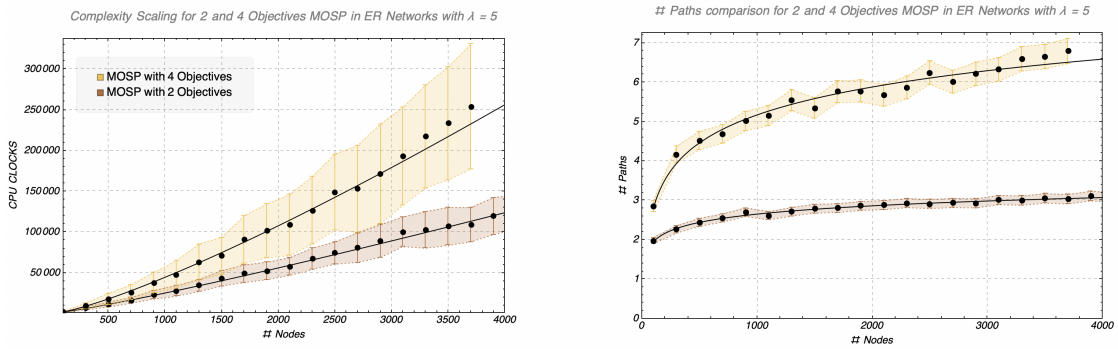
1. The individual communication times for each entangled pair in $[1, 100]$
2. The individual communication times for each entangled pair in $[1000, 10000]$

These simulations are presented in *Figure 4.10*, where we compared the previous results for only two objectives. We only present the results of simulations for the MOSP and not the Star algorithm since, as explained in *Section 3.3.3*, after finding the optimal solutions for the four objectives, a reduction of objectives could be made conjugating the fidelity, memory decoherence times and communication times

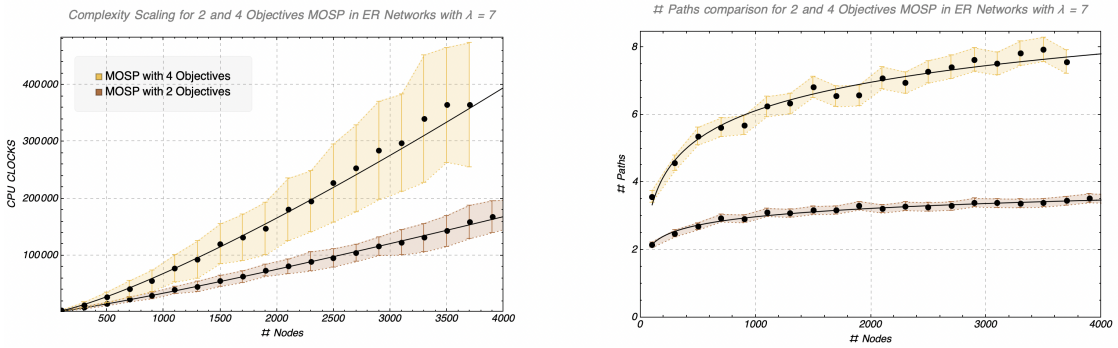
into a single one ($\{\gamma, t_{wait}, \tau\} \mapsto \gamma \cdot e^{-t_{wait}/\tau}$), resulting in the Star algorithm depending again on only two objectives.



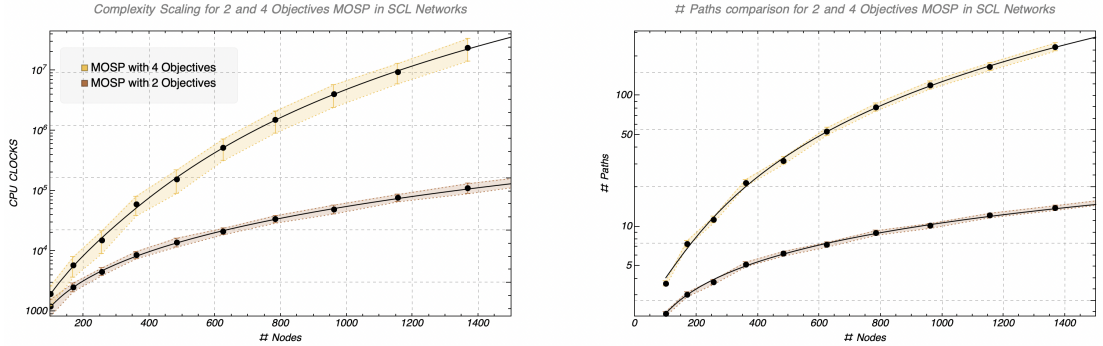
(a) Comparison between 2 and 4 objectives MOSP algorithm in ER network with average degree $\lambda = 3$.



(b) Comparison between 2 and 4 objectives MOSP algorithm in ER network with average degree $\lambda = 5$.



(c) Comparison between 2 and 4 objectives MOSP algorithm in ER network with average degree $\lambda = 7$.



(d) Comparison between 2 and 4 objectives MOSP algorithm in SCL network. (Logarithimic scale)

Figure 4.10: Comparison between 2 and 4 objectives MOSP algorithm. Notice that the scaling remains identical in form for ER networks while for SCL networks the number of optimal paths grows quadratically which results in complexity scaling with a polinomial of degree 5, instead of 3.

4.6 Complexity of Star-Algorithm

The star-algorithm needs the multi-objective shortest-path algorithm presented in *Section 2.3.3, Algorithm 1*, since the first step in finding the shortest is finding the shortest-paths from each terminal to all other nodes of the network. From the shortest-path algorithm structure, we can decompose its complexity in the following elements:

1. How many nodes are visited or revisited, which depends on the number of nodes N, and how many times each nodes is visited $h_{visit} = h_{visit}(N, \lambda, ...)$
2. Each time a node is visited, how many optimal paths $h_{paths} = h_{paths}(N, \lambda, k, ...)$ it adds to its neighbours

From these quantities we can draft a complexity for the algorithm, taking into account that before adding a path from the visited node to its neighbour, the dominance relation must be verified for, at most, all paths on the neighbour:

$$\mathcal{O}\Big(MOSP\Big) = \mathcal{O}\Big(N \cdot h_{visit}(N, \lambda)\Big) \cdot \mathcal{O}\Big(\lambda \cdot h_{paths}(N, \lambda, k)^2\Big) \tag{4.18}$$

The quantities h_{visit} and h_{paths} will rely heavily on the type of network and the parameters distribution across the network. Using the distributions utilised in *Section 4.5*, we are able to calculate the complexity of the algorithm by deriving expressions for the quantities h_{visit} and h_{paths} that depend on the network type and parameter distributions, and therefore compare these results with the simulations. These derivations can be found in *Appendix C*.

Considering that the star-algorithm structure, its complexity will depend on two different things: the complexity of the multi-objective shortest-path algorithm and the number of possible choices for trees that depend on the number of optimal paths from each terminal in each node. Therefore, the complexity of the star-algorithm is given by:

$$\mathcal{O}\Big(Star\Big) = \mathcal{O}\Big(T\Big) \cdot \mathcal{O}\Big(MOSP\Big) + \mathcal{O}\Big(N \cdot h_{paths}(N, \lambda, k)^T\Big) \tag{4.19}$$

Where T is the number of terminals.

In the case of an ER network, the quantity h_{visit} should grow with the number of neighbours, *i.e* the average degree of the network. However, due to the priority queue ordering, this quantity is minimised to an average of $h_{visit} = const$. For an SCL network, the quantity $h_{visit} = const$, since the number of neighbours is always the same, from the network construction, and also from the priority queue ordering.

As for the quantity h_{paths}, which derivations can be found in *Appendix C*, it will depend on the type of network and how the parameters are distributed. The results are the following:

1. ER networks with uniform distributions of the parameters fidelity and probability of success and scaling of fidelity parameters according to *Section 4.5.1*

$$h_{paths} = 1 + \eta \frac{\log N}{\log \langle \lambda \rangle} \tag{4.20}$$

2. SCL networks with uniform distributions of the parameters fidelity and probability of success and scaling of fidelity parameters according to *Section 4.5.1*

$$h_{paths} = \sqrt{N} + \eta N \tag{4.21}$$

Using this, the complexity of the MOSP algorithm and the Star-algorithm can be calculated to be:

1. For ER networks:

   MOSP Algorithm:

$$\begin{aligned} \mathcal{O}\Big(MOSP\Big) &= \mathcal{O}\Big(N \cdot \lambda \cdot h_{paths}(N, \lambda, k)^2\Big) \\ &= \mathcal{O}\Big(N \cdot \lambda \cdot \Big(1 + \eta \frac{\log N}{\log \lambda}\Big)^2\Big) \end{aligned} \tag{4.22}$$

   Star Algorithm:

$$\mathcal{O}\Big(Star\Big) = \mathcal{O}\Big(T \cdot N \cdot \lambda \cdot \Big(1 + \eta \frac{\log N}{\log \lambda}\Big)^2 + N \cdot \Big(1 + \eta \frac{\log N}{\log \lambda}\Big)^T\Big) \tag{4.23}$$

2. For SCL networks:

   MOSP Algorithm:

$$\begin{aligned} \mathcal{O}\Big(MOSP\Big) &= \mathcal{O}\Big(N \cdot h_{paths}(N, k)^2\Big) \\ &= \mathcal{O}\Big(N \cdot \Big(\sqrt{N} + \eta N\Big)^2\Big) \end{aligned} \tag{4.24}$$

   Star Algorithm:

$$\mathcal{O}\Big(Star\Big) = \mathcal{O}\Big(T \cdot N \cdot \Big(\sqrt{N} + \eta N\Big)^2 + N \cdot \Big(\sqrt{N} + \eta N\Big)^T\Big) \tag{4.25}$$

From these complexities for the different algorithms, we can observe that for ER networks, both the MOSP and the Star algorithm are almost linear in the number of nodes of a network, making it a suitable option for a future quantum network with a large number of quantum nodes. For the case of the SCL, the complexity is not linear in the number of nodes, but it also is not exponential, growing polynomially with the number of nodes. These derivations come in line with the numerical simulations made in the previous section, corroborating the results.

4.7 End-of-Chapter Remarks

In this chapter we started by introducing something that came as a natural extension of an algebra for routing, useful to describe distribution schemes for multipartite states. The algebras for trees go the extra step of describing the metrics for multipartite states, while also taking into account different possible metrics for bipartite states. We proceeded to introduce an algorithm that solves (at least approximately) the multi-objective Steiner tree that was inspired by the MOSP algorithm introduced in the first chapter, that is correspondent to the first multipartite scheme of distribution of the previous chapter. Afterwards,

we introduced an algorithm that solves exactly, under some properties of the metrics, the multi-objective shortest star, which we reviewed in detail, and is also present in [33]. Next we presented some simulations, as well as an important characteristic of a network, its scaling. We then corroborated our simulation results with an in-depth analysis of the complexity of our algorithms. It is important to keep in mind that while we used these algorithms in our approach, they are not unique. The approach can be formulated in other ways providing that there exists an algorithm to solve it, for example through genetic algorithms [59–61]. However, due to the properties of the underlying algebras, our algorithms take them into account structurally, resulting in almost linear scaling with the number of nodes, for the case of an ER network.

Chapter 5

Concluding remarks

In the beginning of this thesis, we proposed to introduce a framework in which we gather the several parts of the problem - the network parameters, the entanglement distribution schemes and the routing algorithms - and assemble them in order to solve the routing problem when dealing with multipartite entanglement distribution.

We did this by taking some concepts from classical routing theory - the algebras for routing and multi-objective routing - that became how we would describe each of the parameters taken into account, when finding the optimal way to distribute multipartite entanglement. We also used something very important from quantum channels and operators theory that would allow us to derive simple expressions for the fidelity of the final state. As we said in the beginning, the fidelity is a very important metric regarding the functionality of a quantum state, which is why, having a new useful description of the depolarising channel and a complete form of calculating the resulting state with depolarising channels, is of utmost importance in calculating the fidelity of the state. This was achieved in *Chapter 3*, taking into account the individual fidelities of each quantum link of a network and the distribution schemes. Moreover, we introduced from the literature a few quantum network metrics capable of modelling the entanglement distribution in the case of bipartite, such as communication times, memory decoherence times and the probability of success. This was extended for the multipartite case also in *Chapter 3*.

For both distribution schemes, the tree scheme in *Section 3.2* and the star scheme in *Section 3.3*, we implemented two new different algorithms, targeting the two different problems (shortest-tree and shortest-star). This algorithms were built on top of the foundations laid by the classical routing theory, gathering the previously calculated metrics, with their underlying properties, and the distribution schemes themselves, into a new efficient and adaptable approach on the problem of finding the optimal way of distributing multipartite entanglement in a quantum network. This was fundamental for us to ensure that our algorithm for the star scheme provided the optimal solution, which was done in *Section 4.3*.

While simulating our algorithm on random networks, we stumbled upon a dilemma that foresighted a new problem to ensure that a quantum network is connected - the scaling problem. This problem rose from the fact that the fidelity is a parameter with a threshold of functionality. We solved it by distributing the parameters across the network such that the largest distance in a network remains connected. By doing this in a statistical manner, it shed some light on the problem of deriving the complexity of our

algorithm, which we accomplished by taking into consideration the structure of the algorithm and defining some quantities that depended on the statistical distribution of the parameters. This resulted in the fact that this approach on the complexity is broader for other types of parameters and respective distributions. We presented several results from the simulations, which came in agreement with the calculated values in *Section 4.6*, namely with the star algorithm scaling almost linearly (apart from a poly-logarithmic function) with the number of nodes of the network. In detail, for the case of the ER network, the scaling was in $\mathcal{O}\Big(N \cdot (1 + \eta \log N)^T\Big)$ which is specially important for a network with an increasing number of nodes. For the case of the SCL network, the scaling was in $\mathcal{O}\Big(N \cdot (\sqrt{N} + \eta N)^T\Big)$.

While this framework is used to solve the problem taking into account these two distinct distribution schemes, it is adaptable to new schemes and other parameters capable of characterising the entanglement distribution, for example the rate of distribution [33].

Some notes on possible paths for future work: the main caveat of this framework is that the properties necessary for the exactness of the algorithm solutions are not always verified. In some cases of metrics, while non-isotonic, through some modifications on the algorithms, they can still be implemented and its exactness proved [62]. This would extent the framework allowing more metrics and distribution schemes to be analysed.

Although in this project, the effect of the distributions errors that affected the fidelity was only studied for GHZ states, it would also be important to discuss for other types of states, eventually for W states, which are known to form, for only three qubits, the only two classes of non-equivalent entanglement. More than just studying the effect on different groups of entangled states, which could probably be done by means of quantum operators theory and possible symmetries of the entangled states, the extra step of considering different models of noise would also be meaningful. This extension comes naturally, given that considering the same amount of error in bit-flip, phase-flip and bit-phase-flip errors, *i.e* the standard depolarisation channel, might not always correspond to the physical reality.

So far, we only investigated a scheme that considered no purification protocols that might happen in intermediary steps of the distribution protocols, *i.e* either there was an initial round of purification which guaranteed high fidelities on the initial states or rounds of purification with multiple copies of the distributed state through multipartite quantum purification protocols like the ones introduced in [28–30, 63]. Some distribution schemes present in [54] optimize the distribution schemes for bipartite entanglement by considering purification happening in intermediary steps. It would be important to realise the compatibility with this framework for multipartite entanglement distribution. It would also be interesting to consider the possibility of continuous distribution of entanglement, by continuous generation of bipartite entanglement in the network which would affect the metrics for the network parameters and eventually lead to possible optimisations in the distribution scheme.

While bipartite entanglement distribution has been around for quite sometime now, as applications that rely on multipartite entanglement appear, multipartite entanglement distribution is gradually starting to become not only a side-note of entanglement distribution literature, but a crucial area of investigation. With this project, I hope to have demonstrated that optimally distributing these types of states in a network merges routing problems with distributions schemes.

Bibliography

[1] C. H. Bennett and G. Brassard, *Quantum cryptography: Public key distribution and coin tossing*, Theoretical Computer Science, 560 (2014), pp. 7–11.

[2] A. I. Nurhadi and N. R. Syambas, *Quantum Key Distribution (QKD) Protocols: A Survey*, Proceeding of 2018 4th International Conference on Wireless and Telematics, ICWT 2018, (2018), pp. 18–22.

[3] A. Broadbent, J. Fitzsimons, and E. Kashefi, *Universal blind quantum computation*, Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS, (2009), pp. 517–526.

[4] D. Gottesman, T. Jennewein, and S. Croke, *Longer-baseline telescopes using quantum repeaters*, Physical Review Letters, 109 (2012), pp. 1–5.

[5] P. Kómár, E. M. Kessler, M. Bishof, L. Jiang, A. S. Sørensen, J. Ye, and M. D. Lukin, *A quantum network of clocks*, Nature Physics, 10 (2014), pp. 582–587.

[6] S. Wehner, D. Elkouss, and R. Hanson, *Quantum internet: A vision for the road ahead*, Science, 362 (2018).

[7] A. Pirker and W. Dür, *A quantum network stack and protocols for reliable entanglement-based networks*, New Journal of Physics, 21 (2018).

[8] C. Ren and H. F. Hofmann, *Clock synchronization using maximal multipartite entanglement*, Physical Review A - Atomic, Molecular, and Optical Physics, 86 (2012), pp. 1–4.

[9] E. T. Khabiboulline, J. Borregaard, K. De Greve, and M. D. Lukin, *Quantum-assisted telescope arrays*, Physical Review A, 100 (2019), pp. 1–10.

[10] Z. Eldredge, M. Foss-Feig, J. A. Gross, S. L. Rolston, and A. V. Gorshkov, *Optimal and secure measurement protocols for quantum sensor networks*, Physical Review A, 97 (2018), p. 042337.

[11] T. Qian, J. Bringewatt, I. Boettcher, P. Bienias, and A. V. Gorshkov, *Optimal Measurement of Field Properties with Quantum Sensor Networks*, (2020), pp. 1–6.

[12] M. Hillery, V. Bužek, and A. Berthiaume, *Quantum secret sharing*, Physical Review A - Atomic, Molecular, and Optical Physics, 59 (1999), pp. 1829–1834.

[13] C. Zhu, F. Xu, and C. Pei, *W-state Analyzer and Multi-party Measurement-device-independent Quantum Key Distribution*, Scientific Reports, 5 (2015), pp. 1–10.

[14] G. Murta, F. Grasselli, H. Kampermann, and D. Bruss, *Quantum Conference Key Agreement: A Review*, 2 (2020), pp. 1–15.

[15] J. Fitzsimons, *Private quantum computation: An introduction to blind quantum computing and related protocols*, npj Quantum Information, 3 (2016).

[16] R. Raussendorf and H. J. Briegel, *A one-way quantum computer*, Physical Review Letters, 86 (2001), pp. 5188–5191.

[17] J. Siewert and C. Eltschka, *Quantifying tripartite entanglement of three-Qubit generalized Werner States*, Physical Review Letters, 108 (2012), pp. 1–5.

[18] W. Munro, K. Azuma, K. Tamaki, and K. Nemoto, *Inside quantum repeaters*, IEEE Journal of Selected Topics in Quantum Electronics, 21 (2015), pp. 1–13.

[19] M. Caleffi, *Optimal Routing for Quantum Networks*, IEEE Access, 5 (2017), pp. 22299–22312.

[20] K. Chakraborty, F. Rozpedek, A. Dahlberg, and S. Wehner, *Distributed Routing in a Quantum Internet*, (2019).

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

[22] S. Shi and C. Qian, *Modeling and Designing Routing Protocols in Quantum Networks*, (2019).

[23] C. Li, T. Li, Y.-X. Liu, and P. Cappellaro, *Effective routing design for remote entanglement generation on quantum networks*, (2020).

[24] J. L. Sobrinho, *An algebraic theory of dynamic network routing*, IEEE/ACM Transactions on Networking, 13 (2005), pp. 1160–1173.

[25] C. Meignant, D. Markham, and F. Grosshans, *Distributing Graph States Over Arbitrary Quantum Networks*, Physical Review A - Atomic, Molecular, and Optical Physics, 052333 (2018), pp. 1–6.

[26] J. Wallnöfer, A. Pirker, M. Zwerger, and W. Dür, *Multipartite state generation in quantum networks with optimal scaling*, Scientific Reports, 9 (2019), pp. 1–18.

[27] F. Hahn, A. Pappa, and J. Eisert, *Quantum network routing and local complementation*, npj Quantum Information, 5 (2019), p. 76.

[28] W. Dür, H. Aschauer, and H. J. Briegel, *Multiparticle Entanglement Purification for Graph States*, Physical Review Letters, 91 (2003), pp. 3–6.

[29] H. Aschauer, W. Dür, and H. J. Briegel, *Multiparticle entanglement purification for two-colorable graph states*, Physical Review A - Atomic, Molecular, and Optical Physics, 71 (2005), pp. 1–20.

[30] C. Kruszynska, A. Miyake, H. J. Briegel, and W. Dür, *Entanglement purification protocols for all graph states*, Physical Review A - Atomic, Molecular, and Optical Physics, 74 (2006), pp. 1–9.

[31] S. Brand, T. Coopmans, and D. Elkouss, *Efficient computation of the waiting time and fidelity in quantum repeater chains*, IEEE Journal on Selected Areas in Communications, 38 (2020), pp. 619–639.

[32] W. Dai, T. Peng, and M. Z. Win, *Optimal Remote Entanglement Distribution*, IEEE Journal on Selected Areas in Communications, 38 (2020), pp. 540–556.

[33] L. Bugalho, B. Coutinho, and Y. Omar, *Distribution Multipartite Entanglement over Noisy Quantum Networks*, in preparation, (2021).

[34] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, Cambridge University Press, New York, NY, USA, 10th ed., 2011.

[35] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, *Purification of noisy entanglement and faithful teleportation via noisy channels*, Physical Review Letters, 76 (1996), pp. 722–725.

[36] W. Dur, G. Vidal, and J. I. Cirac, *Three qubits can be entangled in two inequivalent ways*, Physical Review A - Atomic, Molecular, and Optical Physics, 62 (2000), pp. 062314–062311.

[37] C. H. Bennett, S. Popescu, D. Rohrlich, J. A. Smolin, and A. V. Thapliyal, *Exact and asymptotic measures of multipartite pure-state entanglement*, Physical Review A - Atomic, Molecular, and Optical Physics, 63 (2001), pp. 012307–012301.

[38] S. B. Zheng, *Splitting quantum information via W states*, Physical Review A - Atomic, Molecular, and Optical Physics, 74 (2006), pp. 2–5.

[39] M. Hein, W. Dür, J. Eisert, R. Raussendorf, M. Van Den Nest, and H. J. Briegel, *Entanglement in graph states and its applications*, 162 (2006), pp. 115–218.

[40] A. Raina, P. J. Nadkarni, and S. S. Garani, *Recovery of quantum information from a node failure in a graph*, Quantum Information Processing, 19 (2020).

[41] A.-L. Barabási and M. Pósfai, *Network science*, Cambridge University Press, Cambridge, 2016.

[42] W.-H. Steeb and Y. Hardy, *Quantum Channels*, Problems and Solutions in Quantum Computing and Quantum Information, (2018), pp. 369–384.

[43] L. M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, *Long-distance quantum communication with atomic ensembles and linear optics*, Nature, 414 (2001), pp. 413–418.

[44] S. N. Dorogovtsev, A. V. Goltsev, and J. F. Mendes, *Critical phenomena in complex networks*, Reviews of Modern Physics, 80 (2008), pp. 1275–1335.

[45] J. L. Sobrinho, *Network Routing with Path Vector Protocols: Theory and Applications*, in Computer Communication Review, vol. 33, 2003, pp. 49–60.

[46] J. L. Sobrinho, *Fundamental Differences Among Vectoring Routing Protocols on Non-Isotonic Metrics*, IEEE Networking Letters, 1 (2019), pp. 95–98.

[47] Y. Yaling and W. Jun, *Design guidelines for routing metrics in multihop wireless networks*, Proceedings - IEEE INFOCOM, (2008), pp. 2288–2296.

[48] E. Q. V. Martins, *On a multicriteria shortest path problem*, European Journal of Operational Research, 16 (1984), pp. 236–245.

[49] P. Vincke, *Problèmes multicriteres.*, Cah. Cent. Étud. Rech. Opér., 16 (1974), pp. 425–439.

[50] Hansen P., *Bicriterion Path Problems*, Multiple Criteria Decision Making Theory and Application. Lecture Notes in Economics and Mathematical Systems, 177 (1980).

[51] S. Demeyer, J. Goedgebeur, P. Audenaert, M. Pickavet, and P. Demeester, *Speeding up Martins' algorithm for multiple objective shortest path problems*, 4or, 11 (2013), pp. 323–348.

[52] X. Gandibleux, F. Beugnies, and S. Randriamasy, *Martins' algorithm revisited for multi-objective shortest path problems with a MaxMin cost function*, 4or, 4 (2006), pp. 47–59.

[53] K. Chakraborty, D. Elkouss, B. Rijsman, and S. Wehner, *Entanglement Distribution in a Quantum Network, a Multi-Commodity Flow-Based Approach*, (2020), pp. 1–18.

[54] K. Goodenough, D. Elkouss, and S. Wehner, *Optimising repeater schemes for the quantum internet*, (2020), pp. 1–39.

[55] V. V. Kuzmin, D. V. Vasilyev, N. Sangouard, W. Dür, and C. A. Muschik, *Scalable repeater architectures for multi-party states*, npj Quantum Information, 5 (2019), pp. 1–6.

[56] X. Wang, *Exact algorithms for Steiner tree problem*, 2008.

[57] G. Robins and A. Zelikovsky, *Tighter bounds for graph steiner tree approximation*, 19 (2005), pp. 122–134.

[58] C. P. Dettmann and O. Georgiou, *Product of n independent uniform random variables*, Statistics and Probability Letters, 79 (2009), pp. 2501–2503.

[59] P. Leesutthipornchai, C. Charnsripinyo, and N. Wattanapongsakorn, *Solving multi-objective routing and wavelength assignment in WDM network using hybrid evolutionary computation approach*, Computer Communications, 33 (2010), pp. 2246–2259.

[60] H. Yetgin, K. T. K. Cheung, and L. Hanzo, *Multi-objective routing optimization using evolutionary algorithms*, IEEE Wireless Communications and Networking Conference, WCNC, (2012), pp. 3030–3034.

[61] N. Magaia, N. Horta, R. Neves, P. R. Pereira, and M. Correia, *A multi-objective routing algorithm for Wireless Multimedia Sensor Networks*, Applied Soft Computing Journal, 30 (2015), pp. 104–112.

[62] M. Saad, *Non-isotonic routing metrics solvable to optimality via shortest path*, Computer Networks, 145 (2018), pp. 89–95.

[63] W. Dür and H. J. Briegel, *Entanglement purification and quantum error correction*, Reports on Progress in Physics, 70 (2007), pp. 1381–1424.

Appendix A

Calculations of Fidelity

For these calculations, using the second description of the depolarising channel expressed in *Equation A.1* will prove to be more useful:

$$\mathcal{D}_i(\rho, F) = \frac{1+2F}{3}\rho + \frac{2(1-F)}{3}\,\Lambda_i(\hat{Y}_i\rho\hat{Y}_i) \tag{A.1}$$

where Λ_i stands to the partial transposition on qubit i.

Let us first check what is the action of the depolarising over each entry of the matrix $|l\rangle\langle m| \equiv |l_1...l_n\rangle\langle m_1...m_n|$:

$$\begin{aligned}
\mathcal{D}_i(|l\rangle\langle m|, F) &= \mathcal{D}_i(|l_1...l_i...l_n\rangle\langle m_1...m_i...m_n|, F) \\
&= \frac{1+2F}{3}|l\rangle\langle m| + \frac{2(1-F)}{3}\,\Lambda_i(\hat{Y}_i|l_1...l_i...l_n\rangle\langle m_1...m_i...m_n|\hat{Y}_i) \\
&= \frac{1+2F}{3}|l\rangle\langle m| + \frac{2(1-F)}{3}\,\Lambda_i\Big((-1)^{1+\overline{l_i}+m_i}|l_1...\overline{l_i}...l_n\rangle\langle m_1...\overline{m_i}...m_n|\Big) \\
&= \frac{1+2F}{3}|l\rangle\langle m| + \frac{2(1-F)}{3}\,(-1)^{1+\overline{l_i}+m_i}|l_1...\overline{m_i}...l_n\rangle\langle m_1...\overline{l_i}...m_n|
\end{aligned} \tag{A.2}$$

where $\hat{Y}|0\rangle = -i|1\rangle$, $\hat{Y}|1\rangle = i|0\rangle$. Now, considering any state $|\psi\rangle = \sum_{m=0}^{2^n-1}\alpha_m|m\rangle$ with density matrix given by $|\psi\rangle\langle\psi|$:

$$\begin{aligned}
\mathcal{D}_i(|\psi\rangle\langle\psi|, F) &= \sum_{l,m=0}^{2^n-1}\alpha_l^*\alpha_m\mathcal{D}_i(|l_1...l_i...l_n\rangle\langle m_1...m_i...m_n|, F) \\
&= \sum_{l,m=0}^{2^n-1}\alpha_l^*\alpha_m\Big[\frac{1+2F}{3}|l\rangle\langle m| + \frac{2(1-F)}{3}\,(-1)^{1+\overline{l_i}+m_i}|l_1...\overline{m_i}...l_n\rangle\langle m_1...\overline{l_i}...m_n|\Big] \\
&= \frac{1+2F}{3}|\psi\rangle\langle\psi| + \sum_{l,m=0}^{2^n-1}\alpha_l^*\alpha_m\frac{2(1-F)}{3}\,(-1)^{1+\overline{l_i}+m_i}|l_1...\overline{m_i}...l_n\rangle\langle m_1...\overline{l_i}...m_n|
\end{aligned} \tag{A.3}$$

Then, the fidelity can be calculated as follows:

$$\langle\psi|\,\mathcal{D}_i(|\psi\rangle\langle\psi|,F)\,|\psi\rangle = \frac{1+2F}{3} + \langle\psi|\Big[\sum_{l,m=0}^{2^n-1}\alpha_l^*\alpha_m\frac{2(1-F)}{3}\,(-1)^{1+\overline{l_i}+m_i}\,|l_1...\overline{m_i}...l_n\rangle\langle m_1...\overline{l_i}...m_n|\Big]|\psi\rangle$$

$$= \frac{1+2F}{3} + \frac{2(1-F)}{3}\sum_{l',m'=0}^{2^n-1}\sum_{l,m=0}^{2^n-1}\alpha_{l'}^*\alpha_l^*\alpha_m\alpha_{m'}\,(-1)^{1+\overline{l_i}+m_i}\cdot$$

$$\cdot\,\langle l_1'...l_i'...l_n'|l_1...\overline{m_i}...l_n\rangle\langle m_1...\overline{l_i}...m_n|m_1'...m_i'...m_n'\rangle$$

$$= \frac{1+2F}{3} + \frac{2(1-F)}{3}\sum_{l',m'=0}^{2^n-1}\sum_{l,m=0}^{2^n-1}\alpha_{l'}^*\alpha_l^*\alpha_m\alpha_{m'}\,(-1)^{1+\overline{l_i}+m_i}\cdot\delta_{l_1',l_1}...\delta_{l_i',\overline{m_i}}...\delta_{l_n',l_n}\cdot$$

$$\delta_{m_1',m_1}...\delta_{m_i',\overline{l_i}}...\delta_{m_n',m_n} \tag{A.4}$$

From this formula, we verify that the fidelity of the state will depend on the state in which the depolarising channel acts. Another important question would be to calculate the formula for when a subset of the qubits are affected by one depolarising channel. Denoting $\mathcal{A} = \{a_1, a_2, ..., a_m\}$ as the set of qubits with dimension $m = |\mathcal{A}| \leq n$, where the depolarising channels will act, we can calculate the final state from:

$$\mathcal{D}_{\mathcal{A}}(|\psi\rangle\langle\psi|,\{F_i\}_{i\in\mathcal{A}}) = \sum_{l,m=0}^{2^n-1}\alpha_l^*\alpha_m\mathcal{D}_{\mathcal{A}}(|l_1...l_i...l_n\rangle\langle m_1...m_i...m_n|,\{F_i\}_{i\in\mathcal{A}})$$

$$= \mathcal{D}_{\mathcal{A}\setminus a_1}\Big(\sum_{l,m=0}^{2^n-1}\alpha_l^*\alpha_m\Big[\frac{1+2F_{a_1}}{3}\,|l\rangle\langle m| + \frac{2(1-F_{a_1})}{3}\,(-1)^{1+\overline{l_{a_1}}+m_{a_1}}\,|l_1...\overline{m_{a_1}}...l_n\rangle\langle m_1...\overline{l_{a_1}}...m_n|\Big]\Big) \tag{A.5}$$

$$= ...$$

To simplify the calculations, let us separate the diagonal elements form the off-diagonal elements and derive general expressions only for diagonal, since the off-diagonal in the GHZ case will be simpler to calculate due to invariances under the depolarising channel. The diagonal terms of any state become the following:

$$\mathcal{D}_{\mathcal{A}}\,(|l_1...l_n\rangle\langle l_1...l_n|,\{F_i\}_{i\in\mathcal{A}}) = \mathcal{D}_{\mathcal{A}\setminus a_1}\Big(\Big[\frac{1+2F_{a_1}}{3}\,|l_1...l_n\rangle\langle l_1...l_n| + \frac{2(1-F_{a_1})}{3}\,|l_1...\bar{l}_{a_1}...l_n\rangle\langle l_1...\bar{l}_{a_1}...l_n|\Big]\Big)$$

$$= \mathcal{D}_{\mathcal{A}\setminus\{a_1,a_2\}}\Big(\Big[\frac{1+2F_{a_1}}{3}\cdot\frac{1+2F_{a_2}}{3}\,|l_1...l_n\rangle\langle l_1...l_n| + \frac{2(1-F_{a_1})}{3}\cdot\frac{1+2F_{a_2}}{3}\,|l_1...\bar{l}_{a_1}...l_n\rangle\langle l_1...\bar{l}_{a_1}...l_n|\Big] +$$

$$+\,\frac{1+2F_{a_1}}{3}\cdot\frac{2(1-F_{a_2})}{3}\,|l_1...\bar{l}_{a_2}...l_n\rangle\langle l_1...\bar{l}_{a_2}...l_n| + \frac{2(1-F_{a_1})}{3}\cdot\frac{2(1-F_{a_2})}{3}\,|l_1...\bar{l}_{a_1}...\bar{l}_{a_2}...l_n\rangle\langle l_1...\bar{l}_{a_1}...\bar{l}_{a_2}...l_n|\Big)$$

$$(...)$$

$$= \sum_{k=0}^{2^m-1}\prod_{i=1}^{m}\Big[\frac{1+2F_{a_i}}{3}\Big]^{\overline{k_i}}\cdot\Big[\frac{2(1-F_{a_i})}{3}\Big]^{k_i}\sigma_k^{\mathcal{A}}(|l_1...l_n\rangle\langle l_1...l_n|) \tag{A.6}$$

Considering the binary decomposition of $k = k_1k_2...k_m$, then $\sigma_k^{\mathcal{A}}(\rho) = \hat{X}_{a_1}^{k_1}\otimes...\otimes\hat{X}_{a_m}^{k_m}\,\rho\,\hat{X}_{a_1}^{k_1}\otimes...\otimes\hat{X}_{a_m}^{k_m}$.

A.1 Distributing GHZ states using Star-Expansion Protocol

As seen in *Section 3.2*, the star-expansion protocol is identical to successive merges of GHZ states, keeping every Steiner node until the end when they are finally measured. Considering the merge of a m-GHZ

state distributed amongst the set of qubits $\mathcal{A} = \{a_1, a_2, ..., a_m\}$ with an n-GHZ state distributed amongst the set of qubits $\mathcal{B} = \{b_1, b_2, ..., b_n\}$ through the qubit $i = a_m$ of the first state with qubit $j = b_1$ from the second state, identically to *Figure 3.4*, the initial state is the following:

$$\mathcal{D}_{\mathcal{A}\setminus a_1}(|GHZ_m\rangle\langle GHZ_m|, \{F_i\}_{i\in\mathcal{A}\setminus a_1}) \otimes \mathcal{D}_{\mathcal{B}\setminus b_1}(|GHZ_n\rangle\langle GHZ_n|, \{F_i\}_{i\in\mathcal{B}\setminus b_1}) \tag{A.7}$$

After performing the necessary operations to merge the different GHZ states, the final state is also a GHZ state of $n + m - 1 = \tilde{n}$ qubits with the following form:

$$\begin{aligned}
&\sum_{k=0}^{2^{(\tilde{n}-1)}-1} \prod_{i=2}^{m} \left[\frac{1+2F_{a_i}}{3}\right]^{\bar{k}_{a_i}} \cdot \left[\frac{2(1-F_{a_i})}{3}\right]^{k_{a_i}} \prod_{i=2}^{n} \left[\frac{1+2F_{b_i}}{3}\right]^{\bar{k}_{b_i}\oplus k_{a_m}} \cdot \left[\frac{2(1-F_{b_i})}{3}\right]^{k_{b_i}\oplus k_{a_m}} \sigma_k^{\tilde{\mathcal{N}}}(|0\rangle^{\otimes\tilde{n}}\langle 0|^{\otimes\tilde{n}}) + \\
&+ \sum_{k=0}^{2^{(\tilde{n}-1)}-1} \prod_{i=2}^{m} \left[\frac{1+2F_{a_i}}{3}\right]^{\bar{k}_{a_i}} \cdot \left[\frac{2(1-F_{a_i})}{3}\right]^{k_{a_i}} \prod_{i=2}^{n} \left[\frac{1+2F_{b_i}}{3}\right]^{\bar{k}_{b_i}\oplus k_{a_m}} \cdot \left[\frac{2(1-F_{b_i})}{3}\right]^{k_{b_i}\oplus k_{a_m}} \sigma_k^{\tilde{\mathcal{N}}}(|1\rangle^{\otimes\tilde{n}}\langle 1|^{\otimes\tilde{n}}) + \\
&+ \prod_{i=2}^{m}\left[\frac{4F_{a_i}-1}{3}\right] \cdot \prod_{i=2}^{n}\left[\frac{4F_{b_i}-1}{3}\right] |0\rangle^{\otimes\tilde{n}}\langle 1|^{\otimes\tilde{n}} + \prod_{i=2}^{m}\left[\frac{4F_{a_i}-1}{3}\right] \cdot \prod_{i=2}^{n}\left[\frac{4F_{b_i}-1}{3}\right] |1\rangle^{\otimes\tilde{n}}\langle 0|^{\otimes\tilde{n}}
\end{aligned} \tag{A.8}$$

Where $\tilde{\mathcal{N}} = a_1, a_2, ..., a_{m-1}, b_1, b_2, ..., b_n$ are the $\tilde{n}$ qubits indices, $k = k_{a_2}k_{a_3}...k_{a_m}k_{b_2}k_{b_3}...k_{b_n}$ is the binary decomposition of k where the indexes depend on how the qubits are ordered, *i.e* the correspondence between a_i or b_i and the qubit number and $\oplus$ in here stands for the $mod(2)$ sum. To gain more intuition over what is happening in *Equation A.8*, some examples of merges are presented in *Table A.1*. The difference between this state and the regular $\tilde{n}$-GHZ state with $\tilde{n} - 1$ depolarising channels is, because the controlled-Z gate does not commute with the action of the depolarising channel, that the order in the middle terms of the depolarised state changes. The expected order should be the ordered binary decomposition of the numbers between 0 and $2^{\tilde{n}-1}$, but instead (which can be seen from the $\oplus k_{a_m}$ term in A.8), a controlled flip appears between the qubit involved in the controlled-Z operation and every qubit of the n-GHZ state. Moreover, when merging another GHZ state, the effect is cumulative.

Using this approach of merging successive GHZ states, one thing that cannot be forgotten is the Steiner nodes measurements, which we have been postponing for the end. A measurement in this case must always be accompanied by a signal of the outcome to perform a correction and retrieve the same GHZ form as before. In this case, the proper measurement is a X-measurement. If the measured qubit is on a node involved in the merge, *i.e* a Steiner node, then its measurement will result in moving the depolarising channel from that node to the center node plus some qubit SWAP operations which do not actually affect the fidelity of the final state since the GHZ state is invariant under SWAP operations. The other case in which the measured qubit is on a node not involved in the merge implies that such node is actually a leaf of the tree. This implies that, if the algebra for trees is monotone, that such leaf is necessarily a terminal and therefore should not be measured, leaving us with a final state in the following form (minus the SWAP operations to correct the non-GHZ components):

$$\mathcal{D}_{\mathcal{T}\setminus t_0}\left(\mathcal{D}_{t_0}^{\#(\mathcal{S})}\left(|GHZ\rangle\langle GHZ|, \{F_{s_i}\}_{s_i\in\mathcal{S}}\right), \{F_{t_i}\}_{t_i\in\mathcal{T}\setminus t_0}\right) \tag{A.9}$$

Since the analysis of the effect of depolarising channels on all qubits of a GHZ state is made in the next *Section A.2*, let's stick to the derivation of the final state after applying several depolarising channels

a_2	b_2	b_3
0	0	0
0	0	1
0	1	0
0	1	1
1	1	1
1	1	0
1	0	1
1	0	0

Merging a ϕ^+ pair with a 3-GHZ into a 4-GHZ

a_2	a_3	b_2
0	0	0
0	0	1
0	1	1
0	1	0
1	0	0
1	0	1
1	1	1
1	1	0

Merging a 3-GHZ with a ϕ^+ pair into a 4-GHZ

a_2	a_3	b_2	b_3
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0

Merging a 3-GHZ with a 3-GHZ into a 5-GHZ

Table A.1: Examples of possible merges considering that the merge always happens at the last qubit of the first state a_m. The correct way to read this table is the following: each line corresponds to the diagonal entry of the density matrix of the state after the merge and every time a 0 appears it corresponds to the term $\frac{1+2F_I}{3}$ and if a 1 appear, the corresponding term is $\frac{2(1-F_i)}{3}$. In the end, in each line the terms multiply and gives the correspondent value for each matrix entry.

to the same qubit of an GHZ state. Starting with the diagonal entries of the matrix $|0\rangle^{\otimes m}\langle 0|^{\otimes m}$ and $|1\rangle^{\otimes m}\langle 1|^{\otimes m}$ and using the second description of the depolarising channel (*Equation A.1*), from *Equation A.2* we get that:

$$\begin{aligned}\Lambda_i(\hat{Y}_i|l_1...l_i...l_n\rangle\langle l_1...l_i...l_n|\hat{Y}_i) &= \hat{Y}_i|l_1...l_i...l_n\rangle\langle l_1...l_i...l_n|\hat{Y}_i \\ &= (-1)^{1+\bar{l}_i+l_i}|l_1...\bar{l}_i...l_n\rangle\langle l_1...\bar{l}_i...l_n| \\ &= |l_1...\bar{l}_i...l_n\rangle\langle l_1...\bar{l}_i...l_n|\end{aligned} \tag{A.10}$$

And therefore, n applications of this operator over the same state results in:

$$\hat{Y}_i^{\,n}|l_1...l_i...l_n\rangle\langle l_1...l_i...l_n|\hat{Y}_i^{\,n} = \begin{cases}|l_1...l_i...l_n\rangle\langle l_1...l_i...l_n| & \text{if } n \text{ is even} \\ |l_1...\bar{l}_i...l_n\rangle\langle l_1...\bar{l}_i...l_n| & \text{if } n \text{ is odd}\end{cases} \tag{A.11}$$

From this we can verify that the final form will certainly depend on whether the number of depolarising channels is even or odd. To simplify notation consider the following:

$$\begin{aligned}\prod_{i=1}^{n}(a_i+b_i) &= a_1a_2...a_n + b_1a_2a_3...a_n + b_1b_2a_3...a_n + ... + b_1b_2...b_n \\ &= \sum_{\mathcal{Z}_k}\left(\prod_{i\in\mathcal{Z}_k}a_i\prod_{j\in\mathcal{N}\setminus\mathcal{Z}_k}b_j\right) + \sum_{\overline{\mathcal{Z}}_k}\left(\prod_{i\in\overline{\mathcal{Z}}_k}a_i\prod_{j\in\mathcal{N}\setminus\overline{\mathcal{Z}}_k}b_j\right) \\ &= E(\mathbf{a},\mathbf{b},n) + O(\mathbf{a},\mathbf{b},n)\end{aligned} \tag{A.12}$$

Where $\mathcal{Z}_k$ are the subsets of $\mathcal{N} = \{1,2,...,n\}$ with even cardinality and $\overline{\mathcal{Z}}_k$ are the ones with odd cardinality, *e.g* if $\mathcal{N} = \{1,2,3\}$ then $\mathcal{Z} = \{\{\},\{1,2\},\{1,3\},\{2,3\}\}$ and $\overline{\mathcal{Z}} = \{\{1\},\{2\},\{3\},\{1,2,3\}\}$. In the end of *Equation A.12* a new notation is introduced for the two different sums of even cardinality $E(\mathbf{a},\mathbf{b},n)$ and of odd cardinality $O(\mathbf{a},\mathbf{b},n)$ where $\mathbf{a}$ and $\mathbf{b}$ are the vectors comprised of the a_i and b_i terms, respectively.

Using this notation, taking into account the results obtained in *Equations* **??**, we can derive the final form of the n-GHZ state after applying m depolarising channels over one qubit (*w.l.o.g* consider this qubit to be the first one):

$$\begin{aligned}\tilde{\rho} \to\ & E(\mathbf{F},m)|0\rangle^{\otimes n}\langle 0|^{\otimes n} + O(\mathbf{F},m)|1\rangle\langle 1|\otimes|0\rangle^{\otimes n-1}\langle 0|^{\otimes n-1} + \prod_{i=1}^{m}\frac{4F_i-1}{3}\Big[|0\rangle^{\otimes n}\langle 1|^{\otimes n} + |1\rangle^{\otimes n}\langle 0|^{\otimes n}\Big] \\ &+ E(\mathbf{F},m)|1\rangle^{\otimes n}\langle 1|^{\otimes n} + O(\mathbf{F},m)|0\rangle\langle 0|\otimes|1\rangle^{\otimes n-1}\langle 1|^{\otimes n-1}\end{aligned} \tag{A.13}$$

Where $E(\mathbf{F},m) = E(\frac{2(1-\mathbf{F})}{3}, \frac{1+2\mathbf{F}}{3}, m)$ and $O(\mathbf{F},m) = O(\frac{2(1-\mathbf{F})}{3}, \frac{1+2\mathbf{F}}{3}, m)$, where the notation $f(\mathbf{F}) = \{f(F_1),...,f(F_m)\}$ was used. Using all of this, the final fidelity of the GHZ state can be derived after applying the depolarising channels described in *Equation A.9*:

$$f = \frac{E(\mathbf{F}_\mathcal{S},\#S)\cdot\prod_{F_{t_i}\in\mathbf{F}_\mathcal{T}}\frac{1+2F_{t_i}}{3} + O(\mathbf{F}_\mathcal{S},\#S)\cdot\prod_{F_{t_i}\in\mathbf{F}_\mathcal{T}}\frac{2(1-F_{t_i})}{3} + \prod_{F_i\in\mathbf{F}_\mathcal{T}\cup\mathbf{F}_\mathcal{S}}\frac{4F_i-1}{3}}{2} \tag{A.14}$$

Where $\mathbf{F}_\mathcal{S} = \{F_{s_i}\}_{s_i\in\mathcal{S}}$ and $\mathbf{F}_\mathcal{T} = \{F_{t_i}\}_{t_i\in\mathcal{T}\setminus t_0}$.

A.2 Distributing GHZ states using Scheme for Distributing Arbitrary States

As introduced in *Section 2.1.4, Equation 2.20*, the generalisation the GHZ state for an arbitrary m-qubit state is:

$$|GHZ_m\rangle = \frac{|0\rangle^{\otimes m} + |1\rangle^{\otimes m}}{\sqrt{2}} \tag{A.15}$$

The corresponding density matrix only has four entries different than zero, namely:

$$|GHZ_m\rangle\langle GHZ_m| = \frac{|0\rangle^{\otimes m}\langle 0|^{\otimes m} + |0\rangle^{\otimes m}\langle 1|^{\otimes m} + |1\rangle^{\otimes m}\langle 0|^{\otimes m} + |1\rangle^{\otimes m}\langle 1|^{\otimes m}}{2} \tag{A.16}$$

For applying a depolarising in each qubit of the m-qubit GHZ state, let us separate as before, the diagonal from the off-diagonal terms. The diagonal terms become:

$$\begin{aligned}
\mathcal{D}_{\mathcal{M}}(|0\rangle^{\otimes m}\langle 0|^{\otimes m}, \{F_i\}_{i\in\mathcal{M}}) &= \sum_{k=0}^{2^m-1}\prod_{i=1}^{m}\left[\frac{1+2F_i}{3}\right]^{\bar{k}_i}\cdot\left[\frac{2(1-F_i)}{3}\right]^{k_i}\sigma_k^{\mathcal{M}}(|0\rangle^{\otimes m}\langle 0|^{\otimes m}) \\
\mathcal{D}_{\mathcal{M}}(|1\rangle^{\otimes m}\langle 1|^{\otimes m}, \{F_i\}_{i\in\mathcal{M}}) &= \sum_{k=0}^{2^m-1}\prod_{i=1}^{m}\left[\frac{1+2F_i}{3}\right]^{\bar{k}_i}\cdot\left[\frac{2(1-F_i)}{3}\right]^{k_i}\sigma_k^{\mathcal{M}}(|1\rangle^{\otimes m}\langle 1|^{\otimes m})
\end{aligned} \tag{A.17}$$

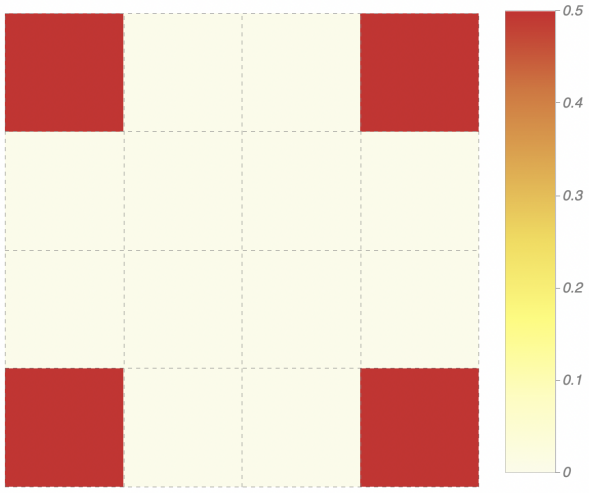
Where $\mathcal{M} = 1, 2, ..., m$ are the m qubits indices. The off-diagonal terms of this state are fairly simple to calculate since they are an eigenvector of $\Lambda_i(\hat{Y}_i\rho\hat{Y}_i^{\dagger})$, $\forall i \in \mathcal{M}$. The correspondent eigenvalue is -1, and therefore:

$$\begin{aligned}
\mathcal{D}_{\mathcal{M}}(|0\rangle^{\otimes m}\langle 1|^{\otimes m}, \{F_i\}_{i\in\mathcal{M}}) &= \prod_{i=1}^{m}\left[\frac{4F_i-1}{3}\right]|0\rangle^{\otimes m}\langle 1|^{\otimes m} \\
\mathcal{D}_{\mathcal{M}}(|1\rangle^{\otimes m}\langle 0|^{\otimes m}, \{F_i\}_{i\in\mathcal{M}}) &= \prod_{i=1}^{m}\left[\frac{4F_i-1}{3}\right]|1\rangle^{\otimes m}\langle 0|^{\otimes m}
\end{aligned} \tag{A.18}$$

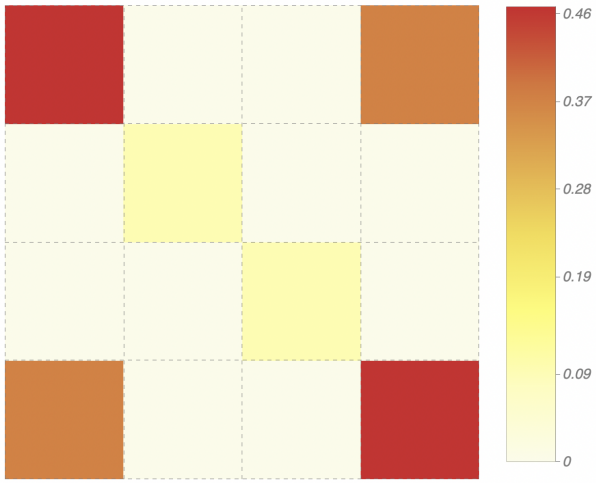
Calculating the fidelity is then simple. From the definitions of measure of fidelity introduced in *Section 2.1.3*, we use $f = \langle\psi|\rho|\psi\rangle$. The final expression then becomes:

$$\begin{aligned}
\langle GHZ_m|\mathcal{D}_{\mathcal{M}}(|GHZ_m\rangle\langle GHZ_m|, \{F_i\}_{i\in\mathcal{M}})|GHZ_m\rangle &= \frac{\prod_{i=1}^{m}\frac{1+2F_i}{3} + \prod_{i=1}^{m}\frac{2(1-F_i)}{3} + \prod_{i=1}^{m}\frac{4F_i-1}{3}}{2} \\
&= \frac{\prod_{i=1}^{m}\frac{1+\gamma_i}{2} + \prod_{i=1}^{m}\frac{1-\gamma_i}{2} + \prod_{i=1}^{m}\gamma_i}{2}
\end{aligned} \tag{A.19}$$

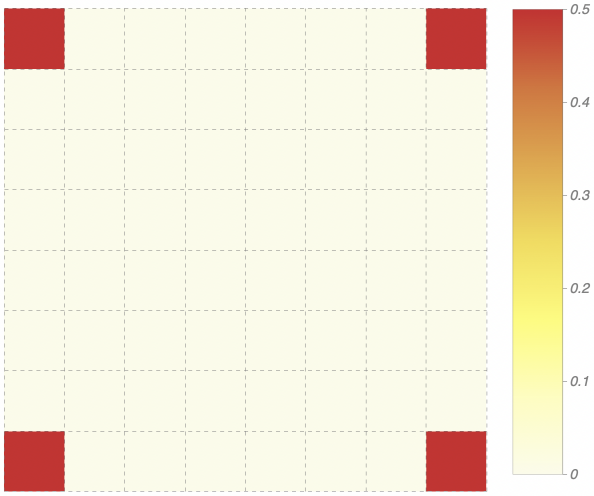
In *Figure A.1* the form of these completely depolarised states is presented in a manner that is visually explicit, ranging from only two qubits to four qubits.



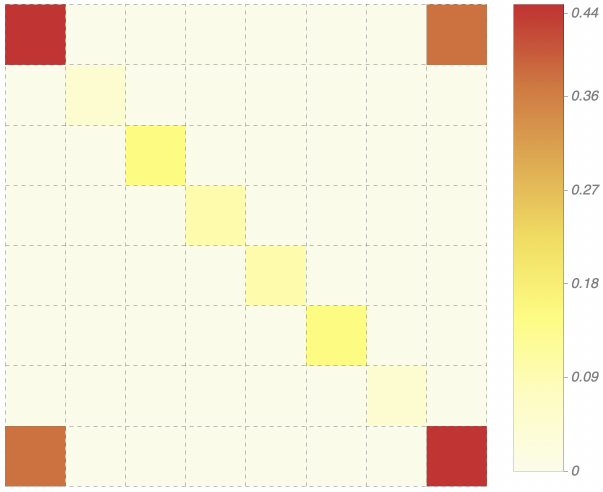
(a) Usual entangled pair $|\phi^+\rangle$ with no depolarising channel.



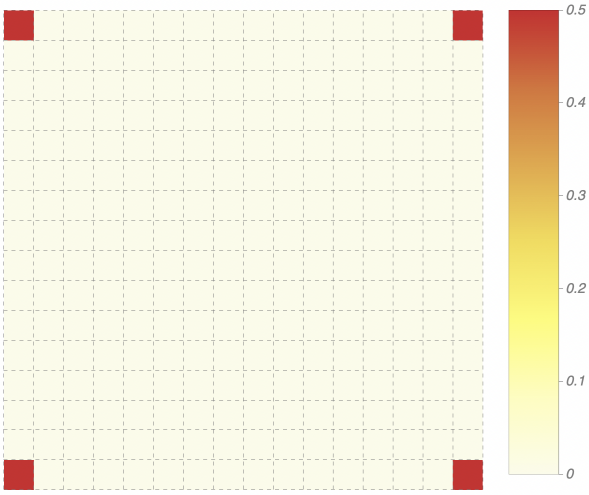
(b) Usual entangled pair $|\phi^+\rangle$ with depolarising channel - $F = 0.9$.



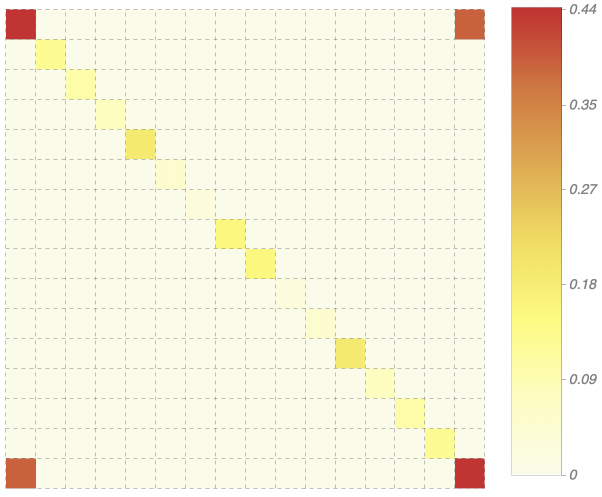
(c) Usual GHZ state with 3 qubits $|GHZ_3\rangle$ with no depolarising channels.



(d) Usual GHZ state with 3 qubits $|GHZ_3\rangle$ with a depolarising channel in each qubit - $F_1 = 0.95; F_2 = 0.9; F_3 = 0.99$



(e) Usual GHZ state with 4 qubits $|GHZ_4\rangle$ with no depolarising channels.



(f) Usual GHZ state with 4 qubits $|GHZ_4\rangle$ with a depolarising channel in each qubit - $F_1 = 0.95; F_2 = 0.9; F_3 = 0.99; F_4 = 0.98$

Figure A.1: Matrix entries plot for the the GHZ states with two (equivalent to a $|\phi^+\rangle$ pair) and three qubits with and without depolarising channels to illustrate the form of a completely depolarised GHZ state.

Appendix B

Monotonicity and Isotonicity Proofs

B.1 Fidelity from Entanglement Swapping Metric

The metric for the fidelity in the γs change of variables is given by the following algebra: $\gamma : \Big([1/3; 1) \cup \{0\}, \geq, (1/3; 1), (0; 1), 0, \oplus_\gamma, g\Big)$ where $\oplus_\gamma$ is the following binary operation:

$$\begin{aligned} \oplus_\gamma : (0;1) \times [1/3;1) &\longrightarrow (0;1) \\ (\gamma_{i:j}, \gamma_{j:k}) &\longmapsto \gamma_{i:j} \cdot \gamma_{j:k} \end{aligned} \tag{B.1}$$

And with $g(\cdot)$ being the following function:

$$g(\gamma) = \begin{cases} \gamma & \gamma \geq 1/3 \\ 0 & \gamma < 1/3 \end{cases} \tag{B.2}$$

Monotonicity

This metric is essentially a multiplication plus a constraint if the value of γ falls below a certain threshold. Using multiplication properties:

$$\gamma_{i:j} \cdot \gamma_{j:k} \leq \gamma_{i:j} \qquad \forall \gamma_{i:j} \in [1/3; 1) \cup \{0\}, \forall \gamma_{j:k} \in [1/3; 1)$$

If after multiplying $\gamma_{j:k}$ the γ value drops below $1/3$, then the ordering still maintains. If the value of $\gamma_{i:j}$ is already below $1/3$, then both paths $i : j$ and $i : k = i : j \oplus j : k$ are equally impossible, both have the special signature weight, which is 0 in this case, and the ordering still maintains since $0 \leq 0$.

Isotonicity

The usual multiplication is trivially an isotonic metric, which can be seen given two different γ values $\gamma^{(1)}_{i:j}, \gamma^{(2)}_{i:j} \in [1/3; 1[\cup \{0\}$ such that $\gamma^{(1)}_{i:j} \geq \gamma^{(2)}_{i:j}$:

$$\begin{gathered} \gamma^{(1)}_{i:j} \geq \gamma^{(2)}_{i:j} \Rightarrow \\ \Rightarrow \gamma^{(1)}_{i:j} \cdot \gamma_{j:k} \geq \gamma^{(2)}_{i:j} \cdot \gamma_{j:k} \qquad \forall \gamma_{j:k} \in [1/3; 1) \end{gathered}$$

The case where one or both paths fall below the threshold, it is easy to check that the order maintains, again because $a \geq 0$, $\forall a \in [1/3, 1)$ and $0 \geq 0$ respectively.

B.2 Waiting time Metric

The metric for the waiting time is given by the following algebra: $t_{wait} : \left(\mathbb{R}^+ \cup \infty, \leq, \mathbb{R}^+, \mathbb{R}^+, \infty, \oplus_t, \mathrm{id}_{\mathbb{R}^+}\right)$ where $\oplus_t$ is the following binary operation:

$$\begin{aligned} \oplus_t : \mathbb{R}^+ \times \mathbb{R}^+ &\longrightarrow \mathbb{R}^+ \\ (t_{i:j}, L_{j:k}/c) &\longmapsto t_{i:j} + L_{j:k}/c \end{aligned} \tag{B.3}$$

Monotonicity

Proving the monotonicity is fairly straightforward since the metric is essentially an addition:

$$t_{i:j} + L_{j:k}/c \geq t_{i:j} \qquad \forall t_{i:j} \in \mathbb{R}^+, \forall L_{j:k}/c \in \mathbb{R}^+$$

Isotonicity

Again, since the metric is essentially an addition, $\forall t^{(1)}_{i:j}, t^{(2)}_{i:j} \in \mathbb{R}^+$ such that $t^{(1)}_{i:j} \leq t^{(2)}_{i:j}$:

$$\begin{aligned} t^{(1)}_{i:j} \leq t^{(2)}_{i:j} \Rightarrow \\ \Rightarrow t^{(1)}_{i:j} + L_{j:k}/c \leq t^{(2)}_{i:j} + L_{j:k}/c \qquad \forall L_{j:k}/c \in \mathbb{R}^+ \end{aligned}$$

B.3 Memory Decoherence Time Metric

The metric for the memory decoherence time is given by the following algebra: $\sigma : \left(\mathbb{R}^+ \cup \infty, \geq, \mathbb{R}^+, \mathbb{R}^+, \infty, \oplus_\sigma, \mathrm{id}_{\mathbb{R}^+}\right)$ where $\oplus_\sigma$ is the following binary operation:

$$\begin{aligned} \oplus_\sigma : \mathbb{R}^+ \times \mathbb{R}^+ &\longrightarrow \mathbb{R}^+ \\ (\tau_{i:j}, \tau_{j:k}) &\longmapsto \Big[\frac{1}{\tau_{i:j}} + \frac{1}{\tau_{j:k}}\Big]^{-1} \end{aligned} \tag{B.4}$$

Monotonicity

Proving the monotonicity is more complicated than a simple addition, but easily verifiable:

$$\begin{aligned} \Big[\frac{1}{\tau_{i:j}} + \frac{1}{\tau_{j:k}}\Big]^{-1} &\leq \Big[\frac{1}{\tau_{i:j}}\Big]^{-1} \\ &= \tau_{i:j} \qquad \forall \tau_{j:k} \in \mathbb{R}^+ \end{aligned}$$

Isotonicity

Proving the isotonicity $\forall \tau^{(1)}_{i:j}, \tau^{(2)}_{i:j} \in \mathbb{R}^+$ such that $\tau^{(1)}_{i:j} \geq \tau^{(2)}_{i:j}$:

$$\begin{aligned}
&\tau_{i:j}^{(1)} \geq \tau_{i:j}^{(2)} \Rightarrow \\
&\Rightarrow \frac{1}{\tau_{i:j}^{(1)}} \leq \frac{1}{\tau_{i:j}^{(2)}} \Rightarrow \\
&\Rightarrow \frac{1}{\tau_{i:j}^{(1)}} + \frac{1}{\tau_{j:k}} \leq \frac{1}{\tau_{i:j}^{(2)}} + \frac{1}{\tau_{j:k}} \Rightarrow \\
&\Rightarrow \Big[\frac{1}{\tau_{i:j}^{(1)}} + \frac{1}{\tau_{j:k}}\Big]^{-1} \geq \Big[\frac{1}{\tau_{i:j}^{(2)}} + \frac{1}{\tau_{j:k}}\Big]^{-1} \qquad \forall \tau_{j:k} \in \mathbb{R}^+
\end{aligned}$$

B.4 Probability of Success Metric

Let us first prove the results for the metric considering that $t = 1$. This associated algebra is $p_{suc} : \Big([0;1), \geq, (0;1) \times (0;1), [0;1), 0, \oplus_p, \mathrm{id}_{[0;1]}\Big)$ where $\oplus_p$ is the following binary operation:

$$\begin{aligned}
\oplus_p : [0;1) \times (0;1)^2 &\longrightarrow [0;1) \\
(p_{i:j}, (p_{j:k}, k_j)) &\longmapsto p_{i:j} \cdot p_{j:k} \cdot k_j
\end{aligned} \tag{B.5}$$

Monotonicity

Similarly to the fidelity metric, the operation performed is the multiplication which is trivially monotonic, even though that this time, two different factors describe each edge.

$$p_{i:j} \cdot p_{j:k} \cdot k_j \leq p_{i:j} \qquad \forall p_{j:k}, k_j \in (0,1)$$

Isotonicity

Proving the isotonicity only requires multiplying both sides of the inequality by the value of the new edge. Given two different $p_{i:j}^{(1)}, p_{i:j}^{(2)} \in [0;1[$ such that $p_{i:j}^{(1)} \geq p_{i:j}^{(2)}$:

$$\begin{aligned}
&p_{i:j}^{(1)} \geq p_{i:j}^{(2)} \Rightarrow \\
&\Rightarrow p_{i:j}^{(1)} \cdot p_{j:k} \cdot k_j \geq p_{i:j}^{(2)} \cdot p_{j:k} \cdot k_j \qquad \forall p_{j:k}, k_j \in (0;1)
\end{aligned}$$

B.5 Fidelity Metric for Distributing Arbitrary States

Considering the algebra for trees $f_{GHZ} : \Big([1/2;1) \cup \{0\}, \geq, (1/2;1), (0;1)^3, 0, \oplus_{GHZ}, h\Big)$ where $\oplus_{GHZ}$ is the following binary operation:

$$\begin{aligned}
\oplus_{GHZ} : (0;1)^3 \times (1/2;1) &\longrightarrow (0;1)^3 \\
(\{a,b,c\}, f_{i:j}) &\longmapsto \{a \cdot \frac{1+2f_{i:j}}{3}, b \cdot \frac{2(1-f_{i:j})}{3}, c \cdot \frac{4f_{i:j}-1}{3}\}
\end{aligned} \tag{B.6}$$

And $h(\cdot)$ is the following function:

$$h(\{a,b,c\}) = \begin{cases} \frac{a+b+c}{2} & , \text{ if } \frac{a+b+c}{2} \geq 1/2 \\ 0 & , \text{ if } \frac{a+b+c}{2} < 1/2 \end{cases} \tag{B.7}$$

Alternatively, to simplify the transition from creating paths and creating starts, the algebra can be transformed to take in γ values and output values for the fidelity of the state. This is done from the following equivalence:

$$\begin{aligned}\frac{1+2f_{i:j}}{3} &= \frac{1+\gamma_{i:j}}{2}\\ \frac{2(1-f_{i:j})}{3} &= \frac{1-\gamma_{i:j}}{2}\\ \frac{4f_{i:j}-1}{3} &= \gamma_{i:j}\end{aligned}$$

Which results in the following algebra $f_{\gamma GHZ} : \Big([1/2;1)\cup\{0\}, \geq, (1/3;1), (0;1)^3, 0, \oplus_{\gamma GHZ}, h\Big)$ where $\oplus_{\gamma GHZ}$ is given by:

$$\begin{aligned}\oplus_{GHZ} : (0;1)^3 \times (1/3;1) &\longrightarrow (0;1)^3\\ (\{a,b,c\},\tilde{\gamma}) &\longmapsto \{a\cdot\frac{1+\tilde{\gamma}}{2}, b\cdot\frac{1-\tilde{\gamma}}{2}, c\cdot\tilde{\gamma}\}\end{aligned} \tag{B.8}$$

Monotonicity

Proving the monotonicity of the fidelity metric is simple. For simplifications of the proofs consider the 3-tupple $\{\prod_i \frac{1+\gamma_i}{2}, \prod_i \frac{1-\gamma_i}{2}, \prod_i \gamma_i\}$ correspondent to the values of $\{a,b,c\}$. Every term of the 3-tupple is itself monotonic, since for every star of p paths:

$$\begin{aligned}a = \prod_{i\in P}\frac{1+\gamma_i}{2} &\geq \prod_{i\in P}\frac{1+\gamma_i}{2}\cdot\frac{1+\tilde{\gamma}}{2}, \forall\tilde{\gamma}\in[1/3,1]\\ b = \prod_{i\in P}\frac{1-\gamma_i}{2} &\geq \prod_{i\in P}\frac{1+\gamma_i}{2}\cdot\frac{1-\tilde{\gamma}}{2}, \forall\tilde{\gamma}\in[1/3,1]\\ c = \prod_{i\in P}\gamma_i &\geq \prod_{i\in P}\gamma_i\cdot\tilde{\gamma}, \forall\tilde{\gamma}\in[1/3,1]\end{aligned} \tag{B.9}$$

This implies that:

$$a+b+c \geq a\cdot\frac{1+\tilde{\gamma}}{2} + b\cdot\frac{1-\tilde{\gamma}}{2} + c\cdot\tilde{\gamma} \qquad \forall\tilde{\gamma}\in(1/3;1) \tag{B.10}$$

Isotonicity

Every term of the 3-tupple is trivially isotonic, however, the sum is not isotonic. This is something common to happen, when we have metrics that are sums or multiplications of two or more individually isotonic metrics, especially when the addition of one path is not identical in form for each of the individually isotonic metrics, *i.e* is not separable for the final sum or multiplication.

Considering the following inequality (in which we removed the 1/2 factor, even though the results are the same):

$$a+b+c = \prod_{i\in P_1}\frac{1+\gamma_i}{2} + \prod_{i\in P_1}\frac{1-\gamma_i}{2} + \prod_{i\in P_1}\gamma_i \geq \prod_{i\in P_2}\frac{1+\gamma_i}{2} + \prod_{i\in P_2}\frac{1-\gamma_i}{2} + \prod_{i\in P_2}\gamma_i = a'+b'+c' \tag{B.11}$$

After extending the star with another path, the inequality to verify becomes:

$$a\cdot\frac{1+\tilde{\gamma}}{2} + b\cdot\frac{1-\tilde{\gamma}}{2} + c\cdot\tilde{\gamma} \geq a'\cdot\frac{1+\tilde{\gamma}}{2} + b'\cdot\frac{1-\tilde{\gamma}}{2} + c'\cdot\tilde{\gamma}$$

The lack of isotonicity comes from the fact that even though that the inequality in *Equation B.11* is observed, because every element of the 3-tupple does not evolve in the same way, when adding some paths, the ordering might be inverted, resulting in non-isotonicity. To prove the non-isotonicity we present one case in which it is not verified:

$$P_1 = \{\gamma_1 = 0.90; \gamma_2 = 0.7408\} \qquad P_2 = \{\gamma_3 = 0.7; \gamma_4 = 0.95\}$$

$$\prod_{i \in P_1} \frac{1+\gamma_i}{2} + \prod_{i \in P_1} \frac{1-\gamma_i}{2} + \prod_{i \in P_1} \gamma_i = 1.50008$$
$$\prod_{i \in P_2} \frac{1+\gamma_i}{2} + \prod_{i \in P_2} \frac{1-\gamma_i}{2} + \prod_{i \in P_2} \gamma_i = 1.4975$$

This implies that P_1 is a better star than P_2. However, when adding another path, the ordering might switch, depending on the γ value of the path, *e.g*:

$$\begin{aligned} \tilde{\gamma} = 0.6 \Rightarrow & \begin{cases} a \cdot \frac{1+\tilde{\gamma}}{2} + b \cdot \frac{1-\tilde{\gamma}}{2} + c \cdot \tilde{\gamma} = 1.06283 \\ a' \cdot \frac{1+\tilde{\gamma}}{2} + b' \cdot \frac{1-\tilde{\gamma}}{2} + c' \cdot \tilde{\gamma} = 1.06275 \end{cases} \\ \tilde{\gamma} = 0.8 \Rightarrow & \begin{cases} a \cdot \frac{1+\tilde{\gamma}}{2} + b \cdot \frac{1-\tilde{\gamma}}{2} + c \cdot \tilde{\gamma} = 1.27822 \\ a' \cdot \frac{1+\tilde{\gamma}}{2} + b' \cdot \frac{1-\tilde{\gamma}}{2} + c' \cdot \tilde{\gamma} = 1.27825 \end{cases} \end{aligned} \tag{B.12}$$

Path-Isotonicity

To prove the path-isotonicity consider two paths with correspondent γ values of $\gamma_1, \gamma_2 \in [1/3, 1)$ such that:

$$\gamma_1 \geq \gamma_2 \tag{B.13}$$

The inequality to prove (see *Definition 4.1.3*) is therefore, $\forall \{a, b, c\} \in (0, 1)$:

$$\begin{aligned} a \cdot \frac{1+\gamma_1}{2} + b \cdot \frac{1-\gamma_1}{2} + c \cdot \gamma_1 &\geq a \cdot \frac{1+\gamma_2}{2} + b \cdot \frac{1-\gamma_2}{2} + c \cdot \gamma_2 \\ \frac{a+b}{2} + \left[\frac{a-b}{2} + c\right]\gamma_1 &\geq \frac{a+b}{2} + \left[\frac{a-b}{2} + c\right]\gamma_2 \\ \left[\frac{a-b}{2} + c\right]\gamma_1 &\geq \left[\frac{a-b}{2} + c\right]\gamma_2 \\ \gamma_1 &\geq \gamma_2 \end{aligned} \tag{B.14}$$

This algebra for trees is path-isotone.

Appendix C

Complexity Calculations

The main quantity we need to calculate to derive the expression for the fidelity, given the algorithms structures, is h_{paths}, which describes the average number of optimal paths across a network. To calculate this quantity, we will first divide our network in neighbouring regions, *i.e* number of paths that have structurally the same distance from the starting node. For each of these paths we analyse the possibility of adding optimal paths with distance no shorter than the minimum distance between such node and the center node and adding paths with bigger distances, but that maintain the optimality, *i.e* they are functionally better (better fidelity and probability of success).

Denote by z_k the neighbours at distance k, Γ^k being k independent realisations of the γ value, *i.e* the distribution of the γ value for a path with k edges and finally, P^k being k independent realisations of the p value. $\mathbb{P}\Big(\cdot\Big)$ stands for the probability of something happening.

Let us begin with the paths at distance $d = 1$:

$$z_1 \cdot \mathbb{P}\Big(\Gamma > \gamma_{trunc}\Big) + z_2 \cdot \Big[\mathbb{P}\Big(\Gamma^2 > \gamma_{trunc} \ \wedge \Gamma^2 > \Gamma \wedge P^2 < P\Big) + \mathbb{P}\Big(\Gamma^2 > \gamma_{trunc} \ \wedge \Gamma^2 < \Gamma \wedge P^2 > P\Big)\Big] + \\ + z_3 \cdot \Big[\mathbb{P}\Big((\Gamma^3 > \gamma_{trunc} \ \wedge \Gamma^3 > \Gamma \wedge P^3 < P\Big) + \mathbb{P}\Big(\Gamma^3 > \gamma_{trunc} \ \wedge \Gamma^3 < \Gamma \wedge P^3 > P\Big)\Big] + ... \tag{C.1}$$

At distance $d = 2$:

$$z_2 \cdot \mathbb{P}\Big(\Gamma^2 > \gamma_{trunc}\Big) + z_3 \cdot \Big[\mathbb{P}\Big(\Gamma^3 > \gamma_{trunc} \ \wedge \Gamma^3 > \Gamma^2 \wedge P^3 < P^2\Big) + \mathbb{P}\Big(\Gamma^3 > \gamma_{trunc} \ \wedge \Gamma^3 < \Gamma^2 \wedge P^3 > P^2\Big)\Big] + \\ + z_4 \cdot \Big[\mathbb{P}\Big((\Gamma^4 > \gamma_{trunc} \ \wedge \Gamma^4 > \Gamma^2 \wedge P^4 < P^2\Big) + \mathbb{P}\Big(\Gamma^4 > \gamma_{trunc} \ \wedge \Gamma^4 < \Gamma^2 \wedge P^4 > P^2\Big)\Big] + ... \tag{C.2}$$

And so on. At distance $d = n$:

$$z_n \cdot \mathbb{P}\Big(\Gamma^n > f_{trunc}\Big) + \sum_{k=n+1}^{+\infty} z_k \cdot \Big[\mathbb{P}\Big(\Gamma^k > \Gamma^n > \gamma_{trunc} \ \wedge P^k < P^n\Big) + \mathbb{P}\Big(\gamma_{trunc} < \Gamma^k < \Gamma^n \ \wedge P^k > P^n\Big)\Big] \tag{C.3}$$

Performing the sum of all distances up to a structural maximum distance (*e.g* the diameter), we can

therefore derive an expression for the total number of paths in the network from the starting node:

$$N \cdot h_{paths} = \sum_{n=1}^{d_{max}} z_n \cdot \mathbb{P}\Big(\Gamma^n > \gamma_{trunc}\Big) + \\ + \sum_{n=1}^{d_{max}} \sum_{k=n+1}^{\infty} z_k \cdot \Big[\mathbb{P}\Big(\Gamma^k > \Gamma^n > \gamma_{trunc} \ \wedge P^k < P^n\Big) + \mathbb{P}\Big(\gamma_{trunc} < \Gamma^k < \Gamma^n \ \wedge P^k > P^n\Big)\Big] \tag{C.4}$$

Since we are interested only in the behaviour of this function, up to a constant, some simplifications can be made to calculated the average number of paths behaviour. The simplifications are:

1. $$\mathbb{P}\Big(\Gamma^k > \gamma_{trunc}\Big) \simeq 1 \quad \text{for } k \leq d_{max}$$

   Given that from the scaling presented in *Section 4.5.1*, by guaranteeing functional connectivity up to the diameter of the network, then the value of $\mathbb{P}\Big(\Gamma^k > \gamma_{trunc}\Big)$ is close to 1 for any $k \leq d_{max}$. It is easy to understand that if the network parameters were not properly scaled, then this approximation would become invalid since the values for $\mathbb{P}\Big(\Gamma^k > \gamma_{trunc}\Big)$ would quickly decrease before $k = d_{max}$.

2. Let X be a random variable distributed uniformly in $[a, 1]$. The probability $\mathbb{P}\Big(X^k > X^n\Big)$ is given, in the limit where $a \to 1$:

$$\lim_{f_{min} \to 1} \mathbb{P}\Big(X^k > X^n\Big) = \frac{T(k+n, n)}{(k+n+1)!} \quad , k > n \tag{C.5}$$

$$\text{where,} \quad T(n,k) = \sum_{j=0}^{k} ((-1)^j \cdot (k-j)^{n+1}) \cdot \binom{n+1}{j} \tag{C.6}$$

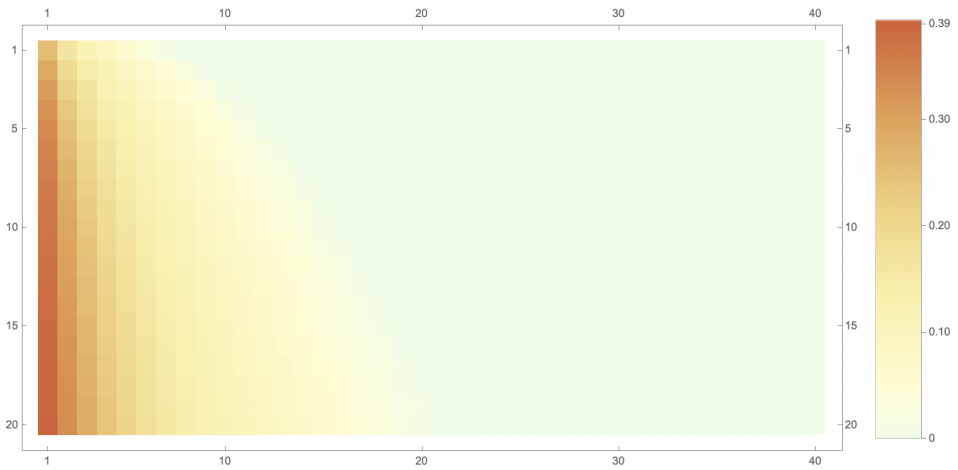


Figure C.1: Probability k independent realisations of Γ has a γ value bigger than n independent realisations of Γ in the limit where $\gamma_{min} \to 1$

Moreover, if we want to create an upper-bound, it is easy to verify that this probability never exceeds $1/2$ and is only equal if $k = n$, this is, both distributions are identical.

3.

$$\mathbb{P}\Big(\Gamma^k > \Gamma^n > \gamma_{trunc} \ \wedge P^k < P^n\Big) = \mathbb{P}\Big(\Gamma^k > \gamma_{trunc}\Big) \cdot \mathbb{P}\Big(\Gamma^k > \Gamma^n \ \wedge P^k < P^n\Big)$$

While $n \leq d_{max}$, the same thing does not happen with k which is summed up to infinity. Because of this, when calculating this terms, two important factors come to play, from the items above, resulting in a contribution that is proportional to d_{max}, *i.e* instead of summing up to infinity, given that the probabilities will tend quickly to small values, we only need to sum up to $\beta \cdot d_{max}$

4.

$$\mathbb{P}\Big(\Gamma^k > \Gamma^n \ \wedge P^k < P^n\Big) = \mathbb{P}\Big(\Gamma^k > \Gamma^n\Big) \cdot \mathbb{P}\Big(P^k < P^n\Big)$$

Since they are unrelated characteristics of the network. Moreover, if $\mathbb{P}\Big(\Gamma^k > \Gamma^n\Big) = p_1$ and $\mathbb{P}\Big(P^k < P^n\Big) = (1 - p_2)$, then:

$$\begin{aligned}\mathbb{P}\Big(\Gamma^k > \Gamma^n\Big) \cdot \mathbb{P}\Big(P^k < P^n\Big) + \mathbb{P}\Big(\Gamma^k < \Gamma^n\Big) \cdot \mathbb{P}\Big(P^k > P^n\Big) &= p_1(1 - p_2) + (1 - p_1)p_2 \\ &= p_1 + p_2 - 2p_1p_2 \\ &\simeq p_1 + p_2 = 2p_1 = 2p_2\end{aligned}$$

This approximation is by excess and gives an interesting insight if we wanted to generalise for an arbitrary number of objectives, namely, that it would grow at most linearly with the number of objectives.

Using the before mentioned approximations, we arrive at a final expression for the total number of optimal paths of:

$$N \cdot f_{paths} = \sum_{n=1}^{d_{max}} z_n \cdot 1 + \sum_{n=1}^{d_{max}} \sum_{k=n+1}^{\beta \cdot d_{max}} 2z_k \cdot \frac{T(k+n, n)}{(k+n+1)!} \tag{C.7}$$

$$\tag{C.8}$$

where z_n is the number of paths beginning in the start node with lenght n. For an Erdós-Renyi, under the tree approximation, this number is equivalent to the number of nth nearest nieghbours $z_n = \langle\lambda\rangle^n$. For the square lattice, this number is more complex to calculate, but is related with the Pascal's Triangle.

C.1 Erdös-Rényi

Using this consideration, we can now calculate the average number of paths for a properly scaled Erdós-Renyi network:

$$d_{max} = \frac{\log N}{\log \langle\lambda\rangle} \tag{C.9}$$

$$\gamma_{min} = (\gamma_{trunc})^{\frac{\alpha}{d_{max}}} \tag{C.10}$$

Using this, we get that:

$$N \cdot h_{paths} = N + \sum_{n=1}^{d_{max}} \sum_{k=n+1}^{\beta \cdot d_{max}} 2\langle\lambda\rangle^k \cdot \frac{T(k+n,n)}{(k+n+1)!} \tag{C.11}$$

$$\leq N + \sum_{n=1}^{d_{max}} \eta N \tag{C.12}$$

$$= N + d_{max} \cdot \eta N \tag{C.13}$$

$$= N(1 + d_{max}) \tag{C.14}$$

$$= N(1 + \eta \frac{\log N}{\log \langle\lambda\rangle}) \tag{C.15}$$

Moreover, this value for η may depend on the number of objectives and the average degree.

C.2 Square Cyclical Lattice

A SCL network structural shortest-paths are easily described by the Pascal's triangle for each quadrant (a SCL network has four distinct quadrants as seen in *Figure C.2*). From this, we get that for each structural distance from the center node, the number of paths will grow exponentially (observe the pink traced line in *Figure C.2*):

$$z_k = 4 \cdot 2^k - 4 \tag{C.16}$$

If the average number of paths for each node grew exponentially, then the algorithm complexity would grow exponentially as well, which is not the case. Imposing that at each distance d, the number of optimal paths $h_{path}^{(d)}$ grows linearly with the distance (we will try to justify this in *Section C.2.1*), then the results are the following:

$$f_{paths}^{(d)}(node) \simeq \beta \cdot d \tag{C.17}$$

$$z_d = \sum_{nodes} \beta \cdot d = 4d \cdot \beta \cdot d \sim d^2 \tag{C.18}$$

$$N \cdot h_{paths} = \sum_{d=1}^{d_{max}} z_d + \sum_{d=1}^{d_{max}} \sum_{k=d+1}^{\beta \cdot d_{max}} 2z_k \cdot \frac{T(d,k+d+1)}{(k+d+1)!} \tag{C.19}$$

$$\sim \sum_{d=1}^{d_{max}} d^2 + \sum_{d=1}^{d_{max}} \sum_{k=d+1}^{\beta \cdot d_{max}} 2z_k \cdot \frac{T(d,k+d+1)}{(k+d+1)!} \tag{C.20}$$

$$\sim d_{max}^3 + \sum_{d=1}^{d_{max}} \eta d_{max}^3 \tag{C.21}$$

$$= d_{max}^3 + \eta d_{max}^4 \tag{C.22}$$

$$= \sqrt{N}^3 + \eta \sqrt{N}^4 \tag{C.23}$$

$$= N(\sqrt{N} + \eta N) \tag{C.24}$$

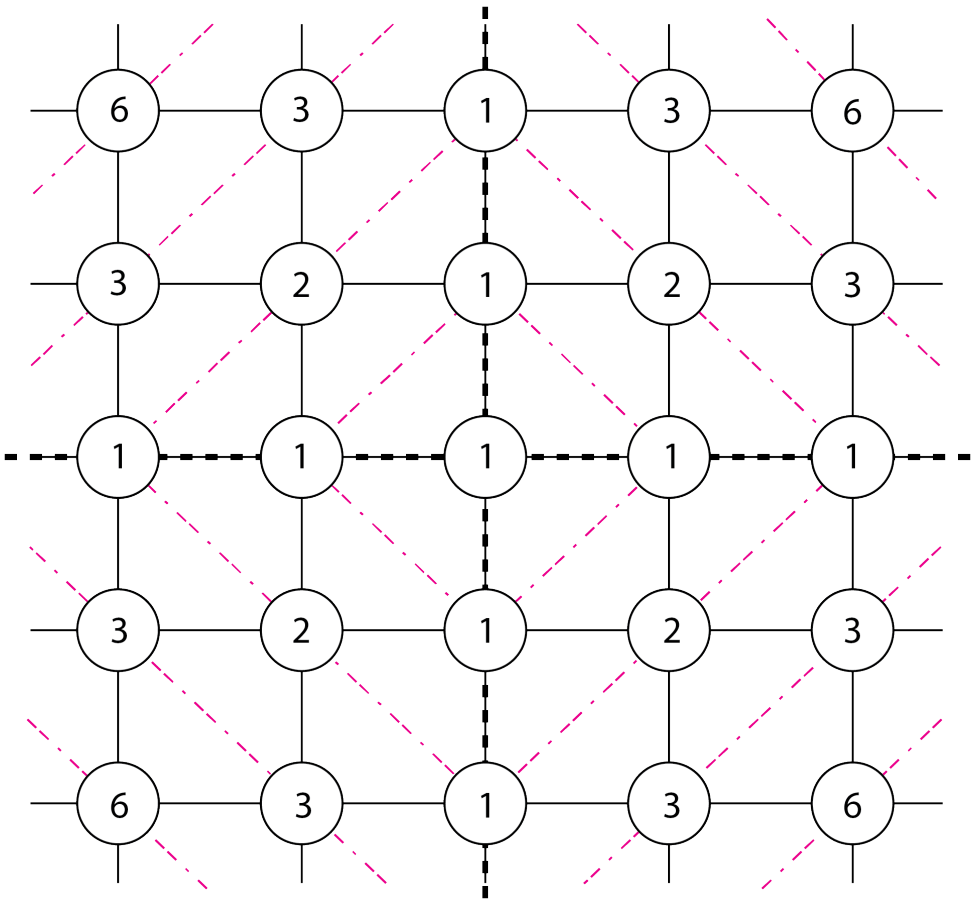


Figure C.2: SCL network representation with quadrants and number of structural shortest-paths for each node.

C.2.1 Number of Optimal Paths for SCL Network

In this section we present an argument for the linearity of the number of optimal paths growing linearly with the distance to the centre node, even though the number of structural shortest-paths grows exponentially.

For each node at a distance n, the number of paths accepted will depend on the number of possible shortest-paths there and the probability of them being eligible:

$$\mathbb{P}\left(\text{Path}_n^i \not\partial \text{Path}_n^j\right) = \mathbb{P}\left(\Gamma^n > \Gamma^n\right) \cdot \mathbb{P}\left(P^n < P^n\right) + \mathbb{P}\left(\Gamma^n < \Gamma^n\right) \cdot \mathbb{P}\left(P^n > P^n\right) \tag{C.25}$$

$$= p_1(1-p_2) + (1-p_1)p_2 \tag{C.26}$$

$$= \frac{1}{2} \tag{C.27}$$

using that $\mathbb{P}\left(\Gamma^n > \Gamma^n\right)$ where both Γ^n are independent realisations of paths with dimension n. If the paths have some intersection, *i.e* links of the network in common, the probability above calculated will be the same since $\mathbb{P}\left(\Gamma^n > \Gamma^n\right) = 1/2, \forall n$. Moreover:

$$\mathbb{P}\left(\text{Path}_n^i \not\partial \{\text{Path}_n^j, \text{Path}_n^k\}\right) = \mathbb{P}\left(\text{Path}_n^i \not\partial \text{Path}_n^j \wedge \text{Path}_n^i \not\partial \text{Path}_n^k\right) \tag{C.28}$$

$$= \mathbb{P}\left(\text{Path}_n^i \not\partial \text{Path}_n^j\right) \cdot \mathbb{P}\left(\text{Path}_n^i \not\partial \text{Path}_n^k\right) \tag{C.29}$$

$$= \frac{1}{2} \cdot \frac{1}{2} \tag{C.30}$$

Using this, we verify that the probability of adding n paths to a set of m paths (see *Figure C.3*), all with the same distance (equivalently parameters distributions) decreases exponentially with the number of paths present and number of paths to add. Given the structure of the network (the Pascal's triangle like structure), the shortest-path length paths can always come from, at best, two previous nodes. Performing the algorithm, when the nodes are visited the first time all the paths written and are non-dominated. Then, when revisited by the other neighbour, it only adds the non-dominated ones and removes the dominated, so throughout the algorithm, in this network, we are always dealing with merging two different sets of paths. Let us first consider only the shortest ones.

Finding the set of non-dominated paths, all with the same distance, from two lists has some interesting properties:

1. Every option has a probability of happening

2. The minimum number of paths is:

$$\min \#X_p(Paths_n^{(i)} \cup Paths_n^{(j)}) = \min\{\#X_p(Paths_n^{(i)}) \ , \ \#X_p(Paths_n^{(j)})\}$$

   where X_p stands for the set of non-dominated paths and $Paths_n^{(i)}$ stands for the list of paths indexed by i at distance n. This minimum number comes from the fact that each of the sets is composed by non-dominated paths.

3. The maximum number of paths is:

$$\max \#X_p(Paths_n^{(i)} \cup Paths_n^{(j)}) = \#X_p(Paths_n^{(i)}) \ + \ \#X_p(Paths_n^{(j)})$$

where no paths dominate each. This as seen before is very unlikely, with the probability of happening decreasing exponentially.

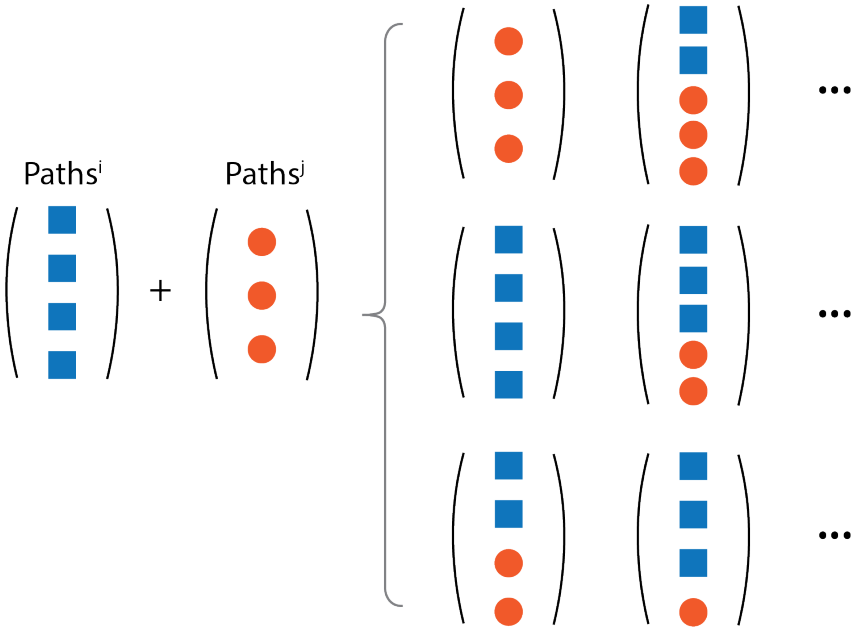


Figure C.3: Some of the possible choices of finding a the set of non-dominated paths from two different lists of paths with the same distributions.

Because of this, the exponential growth of the number of structural shortest-paths is counterbalanced by the exponential decrease in the probabilities, resulting in a linear growth in the number of optimal paths with the distance from the center node.