

# Entanglement distribution in quantum complex networks

Martí Cuquet

Grup de Física Teòrica: Informació i fenòmens quàntics  
Departament de Física  
Facultat de Ciències



Universitat Autònoma de Barcelona

Tesi del programa de Doctorat en Física de la  
Universitat Autònoma de Barcelona  
escrita sota la direcció del  
Prof. John Calsamiglia Costa

Bellaterra, setembre de 2012

Copyright © 2012 Martí Cuquet Palau <[marti.cuquet@gmail.com](mailto:marti.cuquet@gmail.com)>

This work is licensed under the terms of the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported Licence. You are free to copy, communicate and adapt this work, as long as your use is not for commercial purposes, any derivative works are licensed under this license (or similar license to this one) and you attribute Martí Cuquet. The full license can be found at <http://creativecommons.org/licenses/by-nc-sa/3.0/>.

Source of epigraphs (p. iii): Joan Fuster, *Diari 1952-1960*, p. 429; Pere Calders, *Ronda naval sota la boira*.

Cover design by Judit Armengol Monpel <http://www.juditarmengol.com>

This thesis was written with vim and typeset by L<sup>A</sup>T<sub>E</sub>X, using typefaces TeX Gyre Pagella and Latin Modern Sans.

*La teoria dels homes se superposa a la realitat  
com una xarxa: no arriba a més. Per fortuna, la  
seva 'pesca' acostuma a ser fructuosa...*

Joan Fuster

*Sense cap particular al qual referir-se des d'aquí,  
l'autor saluda afectuosament el(s) seu(s)  
lector(s).*

Pere Calders



---

## Agraïments

---

Recollir en poques paraules els agraïments a una llista concreta de persones no és pas la part més senzilla d'escriure una tesi.

El primer lloc és per a en John Calsamiglia, que m'ha guiat en la meva entrada al món de la investigació i m'ha sabut contagiar la seva curiositat i entusiasme. Però més enllà de les seves explicacions, ajut i suport pràctic, sense els quals aquesta tesi no hauria estat possible, vull agrair també la seva proximitat i manera de ser que l'han convertit en un amic més.

Aquest agraïment s'estén també a tots els companys del Grup d'Informació Quàntica, Física Teòrica i l'IFAE. Als companys del primer despatx, en Toni Picón i en David Diego, per ajudar-me a entrar al dia a dia de la recerca i regalar-me moments genials i alguna bona caminada pels Pirineus, i als qui han estat els companys de despatx des d'aleshores, en Marc Montull, en Marc Ramon i la Julia Stasińska. També a la resta de companys del grup, Mariona, Gael, Bernat, Elio, Gabriele, Simone, Luca, i en especial en Julio per l'amistat, els bons consells i l'ajut en començar el que serà la meva nova etapa. Tampoc poden faltar l'Albert, l'Anna, l'Emili, en Marià i en Ramon, que fan que aquest grup gaudeixi d'un ambient tan agradable, ni la resta de gent que volta per Física Teòrica i l'IFAE, sobretot en Lluís, en Jordi, la Chiara, en Volker i tots els altres que han compartit dinar i conversa dia sí i dia també.

Un doctorat no s'acaba dins les quatre parets de la universitat. Durant tot aquest temps molta gent de fora m'ha fet costat. D'entre tots ells, vull mencionar aquells que han incidit d'alguna manera en aquesta tesi: l'Albert i el Cuervo, amics de sempre i companys de pis primer i de ciutat després, que a més em van fer un cop de mà amb els meus primers dubtes de programació —l'avantatge de viure amb dos informàtics—; la Judit, gràcies per la portada, i la Sophie, per resoldre els meus dubtes lingüístics; en Jordi i en Tadeo, que sempre s'han interessat pel meu

treball —molta sort amb la vostra recerca! També en Toni, en Milhouse, que ens ha deixat a tots amb un buit enorme.

Un doctorat és també un treball sovint aïllat i individualista. Per això he d'agrair no només la feina, l'esforç i la lluita de tota la gent de l'Assemblea de Tercer Cicle de la UAB, que representa una dedicació molt gran i potser poc reconeguda, sinó també la companyia, suport i aprenentatge constant que significa formar-ne part.

Finalment, és impossible agrair prou a tota la meva família, avis, pare, mare i germà, el suport i afecte que em donen cada dia, a més de l'interès sincer i entusiasme que mostren per la meva feina, encara que sovint sigui difícil d'explicar. Aquest agraïment és també per a la família de València.

I a la Diana, per fer que tot passés amb un somriure a la cara.

*Poble Sec, Barcelona  
Setembre de 2012*

---

# Contents

---

<b>Agraiments</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Complex networks</b>	<b>7</b>
2.1 Networks as graphs . . . . .	8
2.1.1 Vertices and edges . . . . .	9
2.1.2 Degree . . . . .	10
2.1.3 Clustering . . . . .	12
2.1.4 Paths and cycles . . . . .	13
2.1.5 Components . . . . .	14
2.1.6 Coloring . . . . .	15
2.2 Network models . . . . .	16
2.2.1 Random graphs . . . . .	17
2.2.2 The configuration model . . . . .	21
2.2.3 Small worlds . . . . .	23
2.2.4 A real-world example: the OpenPGP Web of Trust . . . . .	25
2.3 Percolation on complex networks . . . . .	27
2.3.1 Components in uncorrelated networks . . . . .	29
2.3.2 Generating functions . . . . .	33
2.3.3 Percolation and component sizes . . . . .	35
2.3.4 Mean component size, percolation threshold and the giant component . . . . .	36
2.4 Simulation of networks . . . . .	38
2.4.1 Graph data structures . . . . .	38
2.4.2 Random numbers . . . . .	40

2.4.3	Generation of graphs . . . . .	41
2.4.4	Algorithms on graphs . . . . .	43
2.4.5	Implementation of bipartite edges and the $q$ -swap . . . . .	45
<b>3</b>	<b>Distribution of bipartite entanglement</b>	<b>47</b>
3.1	Entanglement in pure states . . . . .	51
3.1.1	Entangled states . . . . .	51
3.1.2	Deterministic and probabilistic transformation of entanglement . . . . .	52
3.1.3	Entanglement swapping . . . . .	54
3.2	Entanglement percolation in pure-state networks . . . . .	55
3.2.1	Critical entanglement and long-distance entanglement probability . . . . .	55
3.2.2	Network model . . . . .	57
3.2.3	Network transformation with local knowledge . . . . .	58
3.2.4	Generating functions of the modified network . . . . .	60
3.2.5	Strategies to implement $q$ -swaps . . . . .	64
3.2.6	Network examples . . . . .	68
3.2.7	Explosive percolation: advance and delay of the transition . . . . .	74
3.3	Entanglement in mixed states . . . . .	76
3.3.1	States from amplitude-damping channels . . . . .	76
3.3.2	General states . . . . .	78
3.4	Limited-path-length percolation in mixed-state networks . . . . .	79
3.4.1	Network model, minimum fidelity and maximum path length . . . . .	81
3.4.2	Path length distribution and limited components . . . . .	82
3.4.3	Generating functions of the limited components . . . . .	83
<b>4</b>	<b>Distribution of multipartite entanglement</b>	<b>87</b>
4.1	Graph states . . . . .	90
4.1.1	Graph states in the interaction picture . . . . .	90
4.1.2	Graph states in the stabilizer formalism . . . . .	91
4.1.3	Mixed graph states . . . . .	92
4.1.4	Operations and measurements on graph states . . . . .	93
4.2	Network and noise model . . . . .	94
4.3	Multipartite purification protocol . . . . .	95
4.4	Protocols . . . . .	98
4.4.1	Bipartite A protocol . . . . .	99
4.4.2	Bipartite B protocol . . . . .	101
4.4.3	Purify subgraph and merge . . . . .	102
4.5	Fidelity . . . . .	104
4.6	Distribution of a closed linear cluster . . . . .	106

---

4.6.1	Exact solution via generating functions of the domains . . .	107
4.6.2	Comparison with the mean-field approximation . . . . .	114
4.7	Distribution of graph states in complex networks . . . . .	117
<b>5</b>	<b>Conclusions</b>	<b>121</b>
	<b>Bibliography</b>	<b>127</b>



---

## List of Figures

---

2.1	The seven bridges of Königsberg . . . . .	8
2.2	A graph example . . . . .	9
2.3	Examples of regular graphs . . . . .	10
2.4	Bethe lattices and Cayley tree . . . . .	11
2.5	Degree and clustering in a graph . . . . .	12
2.6	Shortest path and average path length . . . . .	13
2.7	Some graph examples . . . . .	14
2.8	Graph coloring . . . . .	15
2.9	The 8 graphs of the Gilbert model $N = 3$ ensemble . . . . .	17
2.10	Comparison between a Poisson and a scale-free degree distributions	22
2.11	Base lattice for the Watts-Strogatz model . . . . .	24
2.12	Watts-Strogatz model . . . . .	24
2.13	OpenPGP Web of Trust . . . . .	25
2.14	Degree sequence and clustering of the Web of Trust . . . . .	26
2.15	Examples of degree probabilities . . . . .	29
2.16	Graphical solution of Eq. 2.32 of the Erdős-Rényi network . . . . .	32
2.17	Adjacency matrix and adjacency list . . . . .	39
2.18	Implementation of the configuration model . . . . .	42
3.1	Entanglement swapping . . . . .	55
3.2	Model of a quantum network of bipartite pure states . . . . .	58
3.3	$q$ -swap transformation . . . . .	59
3.4	Degree sequence before and after 2, 3-swap in an Erdős-Rényi and a scale free network . . . . .	60
3.5	Example of the branching process before and after a 3-swap . . . . .	62
3.6	Schematic representation of $h_R(z)$ and $C_2(z)$ . . . . .	63
3.7	Implementation of 2-swap in a cluster of nodes with degree 2 . . . . .	64

3.8	Probability $\eta_2$ of performing a 2-swap . . . . .	65
3.9	Clusters of nodes with the same degree . . . . .	66
3.10	Critical entanglement of the Bethe lattice . . . . .	69
3.11	Entanglement percolation in the Erdős-Rényi model . . . . .	71
3.12	Entanglement percolation in a scale-free network . . . . .	72
3.13	Entanglement percolation in the Watts-Strogatz model . . . . .	73
3.14	Critical entanglement in the Watts-Strogatz model and entangle- ment percolation in the World Wide Web . . . . .	74
3.15	Explosive percolation . . . . .	75
3.16	Clusters of limited path length in the Web of Trust and the hon- eycomb lattice . . . . .	80
3.17	Scaling of $\langle s_l \rangle$ and $\hat{n}(l)$ in the Erdős-Rényi network . . . . .	82
3.18	Scaling of $\langle s_l \rangle$ and $\hat{n}(l)$ in the honeycomb lattice . . . . .	83
3.19	Normalized $l$ -limited mean component size in the Web of Trust and the honeycomb lattice . . . . .	84
3.20	Limited mean component size of a scale-free and the Watts-Strogatz network . . . . .	85
4.1	Graph state of 5 qubits . . . . .	90
4.2	Subprotocol P1 . . . . .	96
4.3	Bipartite A and B protocols . . . . .	100
4.4	Purify subgraphs and merge . . . . .	102
4.5	Subgraph protocol S1 and S2 in a linear cluster . . . . .	107
4.6	Schematic representation of the vector $\mathbf{x}$ and the noise effect . . .	108
4.7	Fidelity of a linear cluster . . . . .	113
4.8	Rescaled decay rate $f_N$ of a closed linear cluster with perfect pu- rification . . . . .	114
4.9	Fidelity decay rate $pf_N$ of a closed linear cluster . . . . .	116
4.10	Fidelity decay rate $pf_N$ of an Erdős-Rényi network . . . . .	120

# CHAPTER 1

---

## Introduction

---

The picture that we form from our world is based on our observations: the process through which we perceive things, and register them as relevant. We have always relied on this process to obtain information and reduce the uncertainty of our knowledge about the environment. In the Nineteenth century, Boltzmann quantified uncertainty as the number of possible microstates of a system, and provided a new interpretation of entropy as a measure of this uncertainty. This concept appears later in the work of [Shannon \(1948\)](#), in the middle of the last century, who quantifies the amount of information transmitted in a communication setting in terms of the probability of occurrence of an event from a set of alternative possibilities.

Last century saw also the origin of quantum mechanics, a theory that originated to describe many counterintuitive observations of microscopic systems that could not be explained by classical theories. Quantum mechanics has since then become a fundamental theory of physics, with a remarkable precision in the description of experiments. At the same time, it has provided a revolutionary interpretation of nature whose influence spreads to all disciplines of human knowledge. Quantum mechanics includes uncertainty as a fundamental property of nature, and not as a lack of information from our side. It also describes observation, or measurement, not as a mere obtention of information of a system but as a process that also disturbs and modifies it.

Quantum information theory emerged from this radical difference in the extraction of information from quantum systems. As much as classical and quantum systems have fundamental differences in their description and behavior, the newly born quantum information theory appeared to behave fundamentally differently to its classical counterpart, opening new possibilities in computation and communi-

cation, and allowing to perform certain tasks which would be otherwise unfeasible in a purely classical framework. The more prominent examples are quantum computers and quantum cryptography. The firsts, eg via Shor’s quantum factoring algorithm, provide an exponential speed up with respect to any classical algorithm; the latter guarantees fundamentally secure communication based on the laws of quantum mechanics. But, apart from practical implications, quantum information addresses a fundamental understanding of nature. The observation that our physical description of the world is represented by propositions, and that these propositions are associated to systems that can be measured to obtain information from them, puts (quantum) information right in the center of the interpretation of our world (Zeilinger, 1999). Entanglement is a striking feature of quantum mechanics, with a clear interpretation in terms of information, and with no classical equivalent. It reflects that if a quantum composite system is completely described by joint properties, one may be left without local knowledge about its subsystems. It can arise when these subsystems, initially separated and encoding local information, interact with each other. The final state may be represented by joint properties, and if there has not been information exchange with the environment this means that some, or all the individual subsystems do not carry information of their own. This gives rise to stronger correlations than those allowed classically. In the last decade, technology has reached a point in which quantum effects can not only be observed, but also controlled, bringing some applications of quantum information to the real world. Here, quantum communication is maybe the area which has more mid-term applications. Some examples are the already mentioned quantum cryptography (Gisin et al., 2002), teleportation (Bennett et al., 1993) and dense coding (Mattle et al., 1996).

In this thesis we deal with large systems, which are complex and difficult to describe in a comprehensive way. To study the properties of systems and predict their behavior, science develops models that try to capture the essence of such systems and make predictions of their behavior. Typically, we create simple models based on an ordered structure, so a few parameters suffice. Even if we only base the model on a collection of entities and the relations between them, without taking care of the exact nature of such relations, we can already explain some of the effects of different components. This is the idea behind network models, and graphs as their mathematical representation, which are very flexible because do not rely on the nature of interactions—just on their structure. However, these simple models present some limitations. “La teoria de l’home se superposa a la realitat com una xarxa: no arriba a més”<sup>1</sup>, said once Joan Fuster, complaining that our theories can only grasp of reality the same as a fishing net from the sea. This reasoning (which Fuster only used metaphorically) can be true in part due to our tendency to consider regular structures, while real-world systems are often driven by growing processes, or develop to optimize certain functionalities, that

---

<sup>1</sup>“The theory of man overlaps reality like a net: it reaches no more”, Joan Fuster, *Diari 1952–1960*, p. 429 (translation by M.C.).

---

create a complex structure that is neither regular nor completely random.

Complex networks that reproduce this intricate design turned out to be a great tool for many different disciplines in the study, under the same light, of a wide range of systems where different parties interact, be they of natural (Jeong et al., 2001; Scala et al., 2001), social (Watts and Strogatz, 1998; Amaral et al., 2000) or artificial (Pastor-Satorras et al., 2001; Yook et al., 2002) origin. Complex networks are defined by few statistical parameters, which however give rise to a non-trivial structure of complex networks—neither regular nor completely random—that is the source of many important phenomena not present in regular lattices. For example, complex networks have an inherent robustness to random errors, maintaining an important fraction of nodes highly connected (Newman et al., 2001). Another one, which is at the origin of the popular expression “This is such a small world”, is that nodes tend to cluster together while keeping short intervertex distances even for very large networks (Watts and Strogatz, 1998). All these effects are often observed in real graphs and have a deep impact on their performance. The bottom line is that the structure of these networks strongly affects their properties, and knowledge of the characteristics of such dependence can be exploited to control them and enhance their behavior.

In the last decade there has been an important increase of interest in complex networks (Albert and Barabási, 2002; Dorogovtsev et al., 2008; Newman, 2010), but they still remain quite unexplored in the quantum setting. Quantum networks (Cirac et al., 1997; Kimble, 2008), where nodes with (limited) quantum storage and processing power are coupled through quantum channels, are becoming a focus of interest in quantum information. The first motivation is to extend the paradigmatic bipartite quantum communication applications, eg quantum teleportation (Bennett et al., 1993) or quantum key distribution (Gisin et al., 2002), to a multipartite setting, where such bipartite protocols can be accomplished between arbitrary nodes of the network. The possibility of networks with non-trivial topologies can give rise to new phenomena and to applications that exploit the multipartite correlations. Of course, this puts forward a big variety of theoretical and technological challenges which can be addressed in short term. Indeed the first experimental steps have already been taken towards a quantum network in a first realization with two distant nodes that can store and interchange quantum information in an efficient and reversible way (Ritter et al., 2012). Lastly, such physical realizations are in principle scalable and hence open the door to perform highly controllable experiments on many-body phenomena, study multipartite entanglement (Jungnitsch et al., 2011; de Vicente et al., 2012), and could eventually perform more complex tasks like some implementation of the quantum Google page rank (Paparo and Martín-Delgado, 2012) or general distributed quantum computations.

The work contained in this thesis is primarily motivated by the idea that the interplay between these two fields—quantum information and complex networks—may give rise to a new understanding and characterization of natural systems.

Complex networks are of particular importance in communication infrastructures, as most present telecommunication networks have a complex structure. In the case of quantum networks, which are the necessary framework for distributed quantum processing and for quantum communication, it is very plausible that future quantum networks will acquire a complex topology resembling that of existing networks, or even that methods will be developed to use current infrastructures in the quantum regime. Not only are quantum complex networks interesting as the base of communication and distributed applications, but also as a new paradigm in network science. In the same spirit of quantum information, that considers that information is embodied in physical systems and therefore described by quantum mechanics, quantum networks arise from networks whose entities and their relations work in the quantum regime, and may give rise to radically different phenomena. Examples of these are a recent quantum random network model, showing different critical behavior from its classical equivalent (Perseguers et al., 2010b), transport modeled by quantum walks (Muelken and Blumen, 2011), search algorithms (Paparo and Martín-Delgado, 2012; Garnerone et al., 2012; Sánchez-Burillo et al., 2012) or even a model of a quantum social network (Cabello et al., 2011). One of the objectives of this thesis is thus to explore and draw the attention to the interaction between these two fields. Joan Fuster ends his at first pessimistic consideration saying “Per fortuna, la seva ‘pesca’ acostuma a ser fructuosa”<sup>2</sup>. We do believe that this interaction will be indeed very productive.

A central task in quantum networks is to devise strategies to distribute entanglement among its nodes. This thesis deals with the study of quantum networks with a complex structure, the implications this structure has in the distribution of entanglement and how their functioning can be enhanced by operating in the quantum regime. Linear networks have shown to be useful for long-distance bipartite entanglement distribution by means of quantum repeaters (Briegel et al., 1998). However, the study of entanglement distribution over higher dimensional networks is in its infancy. The first results show that in some scenarios the network topology can bring interesting effects like entanglement percolation (Acín et al., 2007; Perseguers et al., 2008, 2010a; Cuquet and Calsamiglia, 2009, 2011; Lapeyre Jr. et al., 2009; Broadfoot et al., 2009, 2010b; Wu and Zhu, 2011) that lead to new approaches to the problem. We will first consider a complex network of bipartite states, both pure and mixed, and study the distribution of long-distance entanglement. Then, we will move to a network with noisy channels and study the creation and distribution of large, multipartite states. We assume that the reader is familiar with quantum information theory, and provide only the essential definitions and concepts—we refer the reader to Nielsen and Chuang (2000). Instead, we have found more instructive to give an introduction to graph theory and complex networks.

The thesis is organized as follows. Chapter 2 is devoted to classical complex

---

<sup>2</sup>“Fortunately, its ‘catch’ is usually fruitful” (translation by M.C.).

---

networks. We introduce graphs as the mathematical representation of networks and define their basic properties. Then we present several models of networks with a complex structure. We begin with the models of [Gilbert \(1959\)](#) and [Erdős and Rényi \(1959\)](#), and use them to define random graphs. These simpler models cannot reproduce some of the important characteristics of complex networks, so we present two extensions—the configuration and the small-world model—that address the problem. As an example of a real-world network with these properties, we present the Web of Trust, a social network related to a cryptographic protocol. This network was presented in [Cuquet and Calsamiglia \(2011\)](#). The possibility to use a network for communication depends on its connectivity. We see how the size of connected components depends on the structural properties of the network, and then show how failures in the connections modify these components. This process, called percolation, is closely related to the distribution of entanglement, so we present the mathematical tools used to address it. At the end of the Chapter, we give an overview of the computer implementation of the network models, processes and transformations that we wrote to perform the numerical simulations contained in this thesis.

In Chapter 3, we study the distribution of bipartite entanglement in complex networks. The Chapter is divided in two parts. In the first one, we consider networks of bipartite pure states. We begin with a brief definition of bipartite entangled states and entanglement transformations. Then, we address the problem of entanglement distribution as an entanglement percolation process in a complex network. Within this approach, perfect entanglement is established probabilistically between two arbitrary nodes. We see that for large networks, the probability of doing so is a constant strictly greater than zero (and independent of the size of the network) if the initial amount of entanglement is above a certain critical value. Quantum mechanics offers here the possibility to change the structure of the network without need to establish new, “physical” channels. By a proper local transformation of the network, the critical entanglement can be decreased and the probability increased. We apply this transformation to the complex network models presented in the previous Chapter. Then we turn to the case of a noisy network of mixed states. We see that for some classes of states, the same approach of entanglement percolation can be used. For general mixed states, we consider a limited-path-length entanglement percolation constrained by the amount of noise in the connections. We see how complex networks still offer a great advantage in the probability of connecting two nodes. These results were reported in [Cuquet and Calsamiglia \(2009, 2011\)](#).

In Chapter 4, we move to the multipartite scenario. We study the creation and distribution of graph states with a structure that mimic the underlying communication network. Graph states are an important class of multipartite entangled states. In this case, we use a network of noisy channels, and consider that operations and measurements are also noisy. To correct this noise, entanglement purification can be used. Standard postselection purification protocols require a

number of resources that grows exponentially with the size of the system. In addition, the exact structure of a large complex network is often unknown: they are instead defined by local statistical properties and partial knowledge, which poses also a restriction in the use of these protocols. To overcome this, we propose a scheme to distribute and purify small subgraph, which are then merged to reproduce the desired state. We compare this approach with two bipartite protocols that rely on a central station and full knowledge of the network structure. We show that the fidelity of the generated graphs can be written as the partition function of a classical disordered spin system (a spin glass), and its decay rate is the analog of the free energy. Applying the three protocols to a one-dimensional network and to complex networks, we see that they are all comparable, and in some cases the proposed subgraph protocol, which needs only local information of the network, performs even better. The results presented in this Chapter were reported in [Cuquet and Calsamiglia \(2012\)](#).

Finally, we conclude in Chapter 5 with a summary of our results and an outlook.

## CHAPTER 2

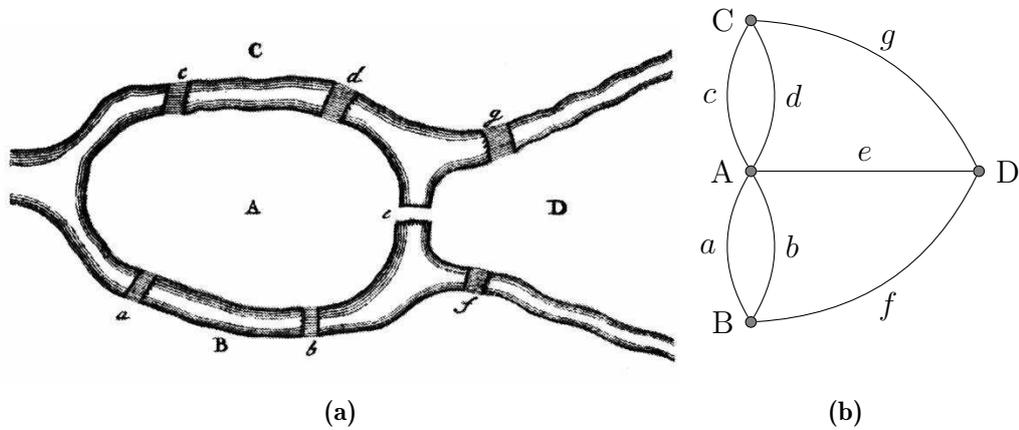
---

### Complex networks

---

Networks permeate all informational structures. They underlie natural, social and artificial systems where different parties interface, describing the flow of information between them. This ubiquity comes from the versatility of their definition: a network is a system defined by a set of entities and their binary relations. Differences in the characteristics of such relations, or interactions, and how they evolve give growth to different types of structures: regular lattices, completely random networks or, spanning the range between these two, complex networks, which do not have a regular structure but neither are completely random. This non-trivial topology is the source of features like the existence of an important fraction of highly connected nodes and the tendency of nodes to cluster together that are often observed in real graphs and have a deep impact in the performance of complex networks. Complex networks arise in many natural and socioeconomic phenomena: protein-protein interactions (Jeong et al., 2001) and protein folding (Scala et al., 2001), neural networks (White et al., 1986) and the human brain (Sporns et al., 2000), the power-grid (Watts and Strogatz, 1998), friendship networks (Amaral et al., 2000) and the Internet (Pastor-Satorras et al., 2001; Yook et al., 2002), to name just a few. Understanding their structural properties is very important as they crucially affect their functionality. For instance, the topology of a social network affects the spread of information (de Solla Price, 1965) or diseases (Klov Dahl et al., 1994), and the architecture of a computer network determines its robustness under router failures or intentional attacks (Cohen et al., 2000; Albert et al., 2000; Cohen et al., 2001).

Networks are naturally represented by graphs: mathematical objects which consist of two sets, elements of one being the relations between elements of the other. In this Chapter we introduce some concepts and techniques of graph theory

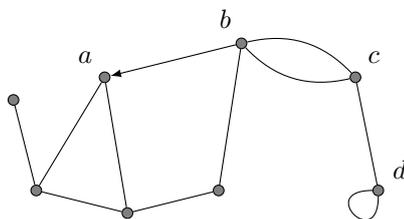


**Figure 2.1.** The seven bridges of Königsberg. (a) Figure in the original paper by Euler (1736), obtained from Mertens (2007). Land areas are labeled by upper-case letters A to D, and bridges by lower-case letters  $a$  to  $g$ . (b) Graph representation of the seven bridges. Vertices A to D are represented by dots, edges  $a$  to  $g$  by lines.

that will help us describe many properties of networks, and use them for networks in the quantum regime. We begin in Section 2.1 with the definition of a graph and its elements, vertices and edges, as well as its local and group properties. Then, in Section 2.2 we introduce the modelling of complex networks as ensembles of random graphs, which reproduce some of the most important properties found in real-world systems: a very short intervertex distance, a degree distribution that often is scale-free, and a high level of clustering. At the end of the Section we also present a real-world example of a social network, the Web of Trust of the OpenPGP classical cryptographic protocol, and some of its properties, like a scale-free degree distribution and the degree-dependent cluster coefficient. This real-world network will be later used in Chapter 3 as an example topology on which distribute bipartite entanglement. In the third Section (2.3) we describe in detail an important process that takes place in networks: percolation and the emergence of a large connected component. This process will be later related to the distribution of bipartite entanglement, and thus we present all the mathematical tools that have been recently developed in the complex networks field. Finally, in the last Section of the Chapter (2.4) we summarize the main data structures and algorithms that we have used in the simulation of classical and quantum complex networks.

## 2.1 Networks as graphs

The roots of graph theory can be traced back to the 18th century, when Leonhard Euler presented his work on the problem of the seven bridges of Königsberg (Euler,



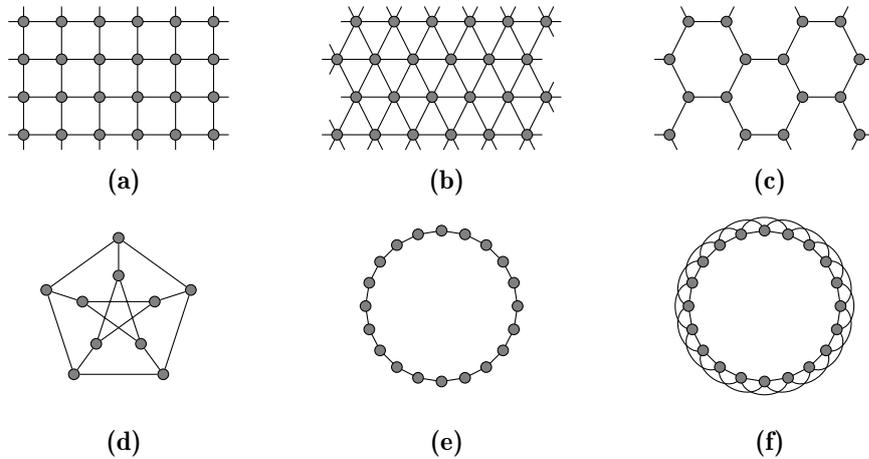
**Figure 2.2.** A graph with a directed edge  $(b, a)$ , a multiedge  $(b, c)$  and a self-edge  $(d, d)$ .

1736). This city was crossed by the river Pregolya, which had two islands connected between them and with the mainland city by seven bridges, as can be seen in 2.1a. The problem consisted in finding a walking route that crossed through every bridge exactly once—no more, no less. To solve it, Euler made an abstract reformulation: the only relevant information in the problem was the list of land areas and the list of bridges connecting them. This is nothing else than the mathematical object that is now called a graph, although the term “graph” did not appear until the second half of the 19th century.

### 2.1.1 Vertices and edges

In graph-theoretical language, each land area is called a **vertex** (or node, or point), and each bridge, an **edge** (or link, or line). The set of vertices  $V$  and the set of edges  $E$  define a **graph**  $G$ , which is the ordered pair of sets  $G = \{V, E\}$ . In this language, the number of vertices  $|V|$  is the order of the graph, and the number of edges  $|E|$ , its size. However, we will use the expression “size of the network” or “size of the graph” for the number of its nodes, or vertices,  $N = |V|$ .

Edges are pairs of elements of  $V$ , so if vertices  $u, v \in V$  are connected by an edge, then we write  $(u, v) \in E$ . Edges can be **directed**, if they can be followed only in one direction, or **undirected**, if both directions are valid. In a directed edge  $(u, v)$ , the order of its elements is important: the former,  $u$ , is the **source** of the edge, and the latter,  $v$ , is the **target**, so the edge can only be traversed from  $u$  to  $v$ . Undirected edges can be traversed in any direction and then  $(u, v) = (v, u)$ . We also speak of **directed graphs** and **undirected graphs** if all their edges are directed or undirected, respectively. There are two special types of edges: **self-edges** are edges whose two elements are the same vertex, and **multiedges** are edges attached to the same pairs of vertices. When a graph is allowed to have multiedges it is called a **multigraph**; the graph that represents the seven bridges problem (depicted in Figure 2.1b) is a multigraph, where  $a$  and  $b$ , and  $c$  and  $d$  are multiedges. However, they are usually forbidden in common graph models, or can be negligible.



**Figure 2.3.** Examples of regular graphs, where all vertices have the same degree. (a) The infinite square lattice, a 4-regular graph. (b) The infinite triangular lattice, a 6-regular graph. (c) The infinite honeycomb lattice, a 3-regular graph. (d) The Petersen graph, a 3-regular graph where all shortest paths are at most of length 2. (e) A ring of 20 vertices, each vertex connected to the 2 vertices nearest itself; it is a 2-regular graph. It is also called a cycle graph. (f) Same ring, but each vertex connected to the 4 vertices nearest itself, making it a 4-regular graph.

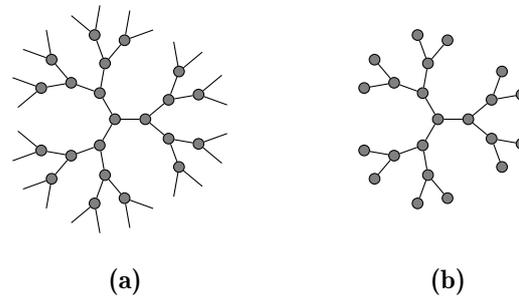
### 2.1.2 Degree

In the problem of the seven bridges, Euler rationalized as follows: except at the beginning and the end of the path, every time one arrives at a vertex through one of its edges, one has to leave through a different one. Therefore, the number of edges that are attached to a vertex must always be even, with the possible exception of two vertices (the start and the end of the path) which must have both either an odd or an even number of edges. This quantity, the number of edges attached to a vertex, is called the **degree** of a vertex. If edges are directed, then one can differentiate between the **incoming** and the **outgoing** degree of a vertex  $u$ , or in- and out-degree for short. The in-degree of  $u$  is the number of edges that have  $u$  as a target, and its out-degree is the number of edges that have it as a source. As it can be seen in Figure 2.1b, in the problem of the seven bridges no path exists that crosses through every bridge exactly once.

Edges have two ends, which contribute to the degree of the vertices they connect. Hence, the sum of the degrees of an undirected graph is related to the number of edges it has,  $M = |E|$ , by  $\sum_{u \in V} k_u = 2M$ . Using the **mean degree**,  $\langle k \rangle = \frac{1}{N} \sum_{u \in V} k_u$ , the number of vertices and edges are related by

$$\langle k \rangle = \frac{2M}{N}. \quad (2.1)$$

In a directed graph, the number of edges is equal to the sum of the in-degrees

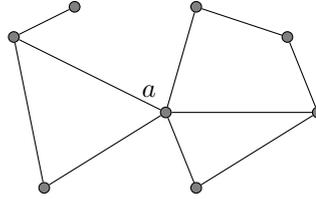


**Figure 2.4.** (a) Bethe lattice and (b) Cayley tree with coordination number (degree of the nodes)  $k = 3$ . Note that the Cayley tree has boundaries (nodes of degree 1, called leaves). There are  $k(k - 1)^{h-1}$  leaves in a Cayley tree of depth  $h$ . The Bethe lattice is infinite and does not contain boundaries: all nodes are of the same degree.

and the sum of the out-degrees. The mean in- and out-degrees are the same, and equal to half the mean of the total degree. When all the degrees of a graph are the same and equal to  $k$ , the graph is called a  **$k$ -regular graph**. Infinite lattices like the square, triangular and honeycomb are regular graphs. Figure 2.3 shows these lattices and examples of other regular graphs.

### The Bethe lattice

An interesting regular graph, which is extensively used as a simple toy model, is the **Bethe lattice**. The reason for that is that usually it is possible to solve analytically the statistical mechanics of models defined on it, and that, although being a regular graph, it shares some important properties with the complex networks that we will present in Section 2.2, such as a local tree-like structure and the small-world effect (as long as the degree of its vertices exceeds 2). A Bethe lattice with coordination number  $k$  is defined as an infinite regular graph where every vertex has the same degree  $k$  and is topologically equivalent to all the others, as shown in Figure 2.4a. Random regular graphs—graphs where all vertices have a fixed degree but edges are placed randomly, as the one in Figure 2.6—asymptotically approach Bethe lattices, making them a relevant model where analytical treatment is usually possible. It is important to note that, although a Bethe lattice has a local tree structure, it is not a tree. This allows to get rid of the border effects that the leaves (vertices of degree one) of a tree would produce, because these leaves appear in a number comparable to the total number of vertices. A graph as the Bethe lattice, but with border vertices of degree 1, is called a **Cayley tree** (Figure 2.4b). This tree has a total of  $N = 1 + k \left[ (k - 1)^h - 1 \right]$  vertices, where  $h = 0, 1, \dots$  is the depth of the tree (the distance from the central vertex to a leaf), and  $k(k - 1)^{h-1}$  leaves. Hence, in the thermodynamic limit  $h \rightarrow \infty$ , the fraction of leaves in the graph is  $\frac{1}{k-1}$ . The fact that it does not go to zero means that the boundary conditions do



**Figure 2.5.** Degree and clustering in a graph with  $N = 8$  and  $M = 10$ . The degree of vertex  $a$  is  $k_a = 5$ . The mean degree of the graph is  $\langle k \rangle = 2M/N = 2.5$ . The local clustering coefficient of vertex  $a$  is  $C_a = 1/5$ . The mean clustering coefficient of the graph is  $\langle C \rangle = 43/120 \simeq 0.36$ . The global clustering coefficient is  $C = 3/10 \simeq 0.33$ .

not become negligible. For a pedagogical review of the differences between Cayley trees and Bethe lattices, see for example that of [Ostilli \(2012\)](#).

### 2.1.3 Clustering

In a network, the degree of a vertex measures the relations between this vertex and the rest of the network. There is another local quantity, the **local clustering coefficient** ([Watts and Strogatz, 1998](#)), which takes a step further and measures the relations between neighbors of a vertex. In an undirected graph, the neighbors of a vertex of degree  $k_u$  can be paired in  $\binom{k_u}{2}$  different ways. The local clustering of  $u$  is defined as the fraction of edges that actually exist between its  $k_u$  neighbors:

$$C_u = \frac{\text{number of edges between neighbors of } u}{\text{number of possible edges between neighbors of } u}. \quad (2.2)$$

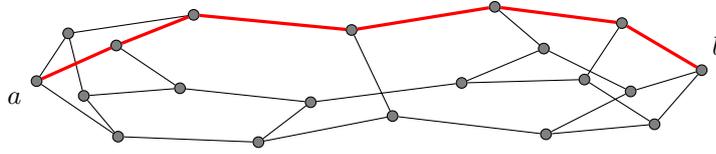
The local clustering coefficient of a vertex with degree zero or one is null by definition. The clustering can also be measured network-wide. The **global clustering coefficient**, or simply clustering coefficient, is the relation between the number of closed triples (groups of three vertices, all connected between them) and the number of connected triples (groups of three vertices, connected between them by either two or three edges) in the whole graph. It is usually written as

$$C = \frac{\text{number of closed triples}}{\text{number of connected triples}} = 3 \frac{\text{number of triangles}}{\text{number of connected triples}}, \quad (2.3)$$

and ranges between 0 and 1. Note that each triangle includes three closed triples, hence the 3 factor. The global clustering coefficient has not to be confused with the **mean clustering**,

$$\langle C \rangle = \frac{1}{N} \sum_{u \in V} C_u, \quad (2.4)$$

which is the average of the local coefficient; in general they take different values. It has to be kept in mind that if a graph has many vertices with degree zero or one, it can have a very low mean clustering, even if other vertices are highly clustered.



**Figure 2.6.** In red, one of the shortest paths (of length 6) between  $a$  and  $b$ . The average path length of the graph is  $l \simeq 2.8$ .

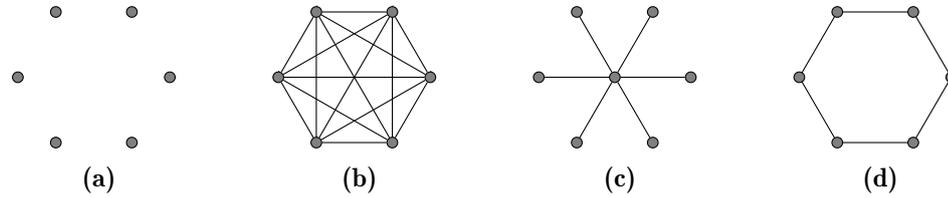
These clustering coefficients give insight into the transitivity of a network: the property that says that if  $a$  is related to  $b$ , and  $b$  is related to  $c$ , then  $a$  is related to  $c$  as well. This property is specially important in social networks, and is commonly stated by the phrase *a friend of my friend is also my friend*.

### 2.1.4 Paths and cycles

Until this point, we have presented graphs and its basic components: vertices and edges. But the rich variety of network phenomena appear when one considers also this basic elements as a group, and the patterns they give rise to. Hence, we now move to quantities that depend of groups of vertices.

In Euler's solution of the seven bridges problem, we already introduced the idea of a path. Formally put, a **path** from  $u_0$  to  $u_l$  is a sequence of vertices  $u_0, u_1, \dots, u_l$  such that every pair  $(u_i, u_{i+1})$  is an element of  $E$  for  $i = 0, 1, \dots, l - 1$ . If the starting and ending vertices are the same, then the path is said to be closed and it is called a **cycle**. The length of a path is the number of edges in it. A distance  $l_{u,v}$  between nodes  $u$  and  $v$  can be defined as the length of the smallest path that joins these two nodes (or  $\infty$  if the two nodes are not connected by any path). This distance is called **intervertex distance**, **minimum path length** or simply **path length**, if by context it does not lead to confusion with the length of an arbitrary (not minimal) path between  $u$  and  $v$ . The **neighborhood** of a vertex  $u$  is then the set of vertices at distance 1 from  $u$ , or equivalently  $\mathcal{N}_u = \{v : (u, v) \in E\}$ . We also presented the degree of a vertex,  $k_u$ . If a vertex has degree equal to zero it is called an isolated vertex; if it has degree equal to one, then the vertex is called a leaf. When neither self-edges nor multiedges are allowed, the degree of a vertex is equal to the number of its neighbors,  $k_u = |\mathcal{N}_u|$ . If the graph is **directed**, the incoming and outgoing neighborhoods of  $u$  are  $\mathcal{N}_u^{(\text{in})} = \{v : (v, u) \in E\}$  and  $\mathcal{N}_u^{(\text{out})} = \{v : (u, v) \in E\}$ , respectively.

Along with the degree and the clustering coefficient, the **average path length** is an important measure of a network. In short, a small average path length tells us if nodes in the network are not too far one from the other, so they can communicate fast and easily. It is formally defined as the average of the minimum path length



**Figure 2.7.** Some graph examples. (a) Empty graph. (b) Complete graph. (c) Star. (d) Cycle.

between all possible pairs of vertices,

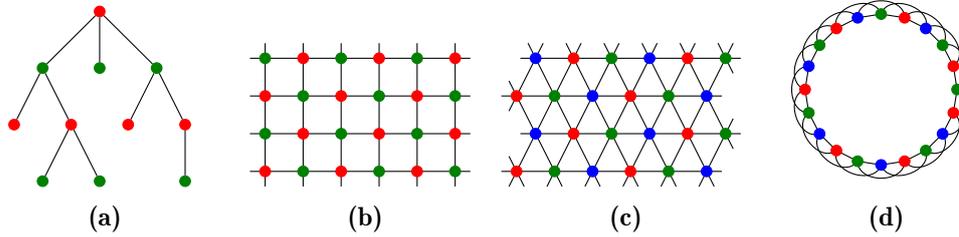
$$l_{\text{av}} = \frac{1}{N(N-1)} \sum_{u,v} l_{u,v}. \quad (2.5)$$

By means of the average path length, two classes of graphs can be differentiated: small-world and large-world networks. In a **small-world network**, growing the network while keeping the average degree fixed results in an average path length that increases at most *logarithmically* with the size of the network. If it increases faster than that, then the network is a **large-world**. This classification differentiates between regular lattices and networks with a complex topology. In a  $d$ -dimensional lattice, the average path length scales as  $l \sim N^{1/d}$ . On the contrary, complex networks are typically small-worlds. This property is known as the **small-world effect** (De Sola Pool and Kochen, 1978; Milgram, 1967) and appears in many real-world communication networks such as the Internet. We will come back to this effect in the next Section 2.2.

### 2.1.5 Components

A **connected component**, or cluster, is a subgraph where any two vertices are connected by at least one path of edges and to which no more vertices can be added without losing this property. If the graph is directed, a **strongly connected component** is the subgraph where pairs of vertices are mutually connected by a directed path. If the direction of the edges is ignored, then the component is **weakly connected**. Depending on whether a graph has only one or more than one connected component, it is called connected or disconnected, respectively. A graph can have many components of different sizes, and one may ask which is the biggest one and what size it has. When this size is of the order of the size of the whole graph, this component is called the **giant connected component**. This concept is closely related to that of a percolating cluster. We will come to that in Section 2.3.

An empty graph is a graph where all vertices are isolated (Figure 2.7a). On the other hand, a graph where any pair of vertices is connected by an edge is a complete graph (Figure 2.7b). Graphs with no cycles and a single connected



**Figure 2.8.** Graph coloring. (a) A tree is always 2-colorable. (b) The square lattice is also 2-colorable. (c) The triangular lattice is 3-colorable, but not 2-colorable. (d) A graph with clustering is not 2-colorable.

component are called **trees**, and are usually simpler to analyse than other graphs. Removing an edge from a tree creates a **forest**, which is a graph with no cycles but more than one connected component. A one-dimensional chain is a tree. Another type of tree is the star: a graph with a central node of degree  $k$ , connected to  $k$  leaves (Figure 2.7c).

### 2.1.6 Coloring

The problem of **coloring** a graph comes originally from the problem of coloring a map. It was conjectured in 1852 that four colors were sufficient to color a map in such a way that no adjacent regions had the same color. The author of the conjecture was Francis Guthrie, who observed that four colors were sufficient to color the map of counties of England. However, theorem was not proved until 1976 by [Appel and Haken \(1977\)](#), being the first theorem proved with the assistance of a computer.

Formally put, a vertex coloring of a graph is a special labelling of its vertices (in this case, the labels are called colors) such that no adjacent vertices share the same color. Of course, the coloring can also be applied to other elements, such as edges. Commonly, when no specification is made, a coloring of a graph is understood as its vertex coloring. A graph that can be colored with  $k$  colors is called  **$k$ -colorable**, and the smallest  $k$  for a graph is its **chromatic number**. Trees are always 2-colorable (Figure 2.8a), and also the square lattice (Figure 2.8b), but not the triangular (Figure 2.8c). More generally, graphs with clustering different from zero are not 2-colorable, as the two neighbors of a node that are also neighbors between them cannot share the same color (Figure 2.8d). In general, it is hard to decide if a graph is  $k$ -colorable—in fact, it is NP-complete if  $k \geq 3$ .

Among many applications of graph coloring, it is important in the purification of graph states, that will be used in Chapter 4. Initially, purification protocols for graph states were restricted to 2-colorable graphs<sup>1</sup> ([Dür et al., 2003](#); [Aschauer](#)

<sup>1</sup>Which anyway include important states such as the Greenberger-Horne-Zeilinger states ([Greenberger et al., 1989](#)) and cluster states ([Briegel and Raussendorf, 2001](#)).

et al., 2005), but were then extended to general colors (Kruszynska et al., 2006).

## 2.2 Network models

After the work of Euler (1736) on the seven bridges problem, where the term graph does not appear but that is considered the first work in this field, graph theory studied small and rather regular graphs. It was not until the mid 20th century that larger and more complex networks began to interest the scientific community. A broad range of disciplines studied the elements and relations of the systems in their field as networks. This research was pioneered by social sciences, where data acquisition allowed to reconstruct the first complex networks, such as that of relations between friends or coworkers<sup>2</sup>. A very important drawback with these experimental data was the difficulty to acquire large and reliable data. An early exception was the network formed by citations between scientific publications (de Solla Price, 1965), which provided larger and more accurate data. In any case, important properties of these networks were already revealed. Perhaps the best known example is Milgram's small world experiment (Milgram, 1967; Travers and Milgram, 1969), popularized by the phrase *six degrees of separation*, that suggested that society is a network with surprisingly short path lengths. In the experiment, participants were given a letter and a target recipient in a distant city, and asked to send the letter directly to the recipient, if they knew her. In case they did not, they had to write down their own name and send it with the letter to a personal acquaintance who they thought could know the recipient. Averaging over the letters that arrived at its destination, Milgram concluded that the average path length was around six. In the last decade, this experiment was repeated by Dodds et al. (2003) using email with very similar results.

To study the functionality and properties of complex networks, and how they are affected by the specific structure of the network, one can try to take the complete, exact description of the system and simply simulate its behavior. However, in many cases this is a formidable task, and then it is essential to use a model of the system. The model should include the main structural properties of the system—such as the mean degree, the degree distribution, or the clustering (Newman, 2002b)—, and the possibility to tune them, without giving explicitly its full topology. The reason for this simplification can be that maybe the structure is too complex to describe in its full generality, or perhaps even more commonly that there is no complete knowledge of the system, but instead only access to some statistical properties. In addition, this allows to control the structural parameters of the network and study their effect. Another important motivation is the usual necessity, or interest, to study the thermodynamical limit  $N \rightarrow \infty$  of these networks, which would otherwise be in many cases impossible or very challenging.

---

<sup>2</sup>See, for example, references in Catanzaro (2008).



**Figure 2.9.** The 8 graphs of the Gilbert model  $N = 3$  ensemble. Colons separate the Erdős-Rényi ensembles for  $N = 3$  and  $M = 0, 1, 2, 3$ , respectively.

Complex network models provide a consistent way to define networks in this limit, and tools can be developed to compute relevant asymptotic quantities.

Such models fix some specific quantities of the network, but are completely random in all other aspects. The first and simplest of these models, that are known as random graphs, are due to [Solomonoff and Rapoport \(1951\)](#), [Gilbert \(1959\)](#) and [Erdős and Rényi \(1959, 1960\)](#), who posed the foundations of modern complex network theory. However, results from both [Milgram](#) and [de Solla Price](#) showed discrepancies when confronted to the simple random models of [Gilbert](#) and [Erdős and Rényi](#) (eg in the expected average path length, or the level of clustering), but the difficulty to acquire large amounts of reliable data stopped from significant advance in the design of more accurate models. This fact changed with the emergence of Internet and computerization of data acquisition, that boosted the advance in the understanding of the structure and function of complex networks ([Albert and Barabási, 2002](#); [Bornholdt and Schuster, 2003](#); [Dorogovtsev et al., 2008](#); [Newman, 2010](#)). These new datasets allowed for a more systematic approach to networks, and revealed the importance of basic properties such as the degree distribution of the network and its clustering. It also served as a motivation to develop models that tried to reproduce the formation mechanisms that underly these networks, to see which properties emerge, rather than statistically describing the system.

In this Section we introduce the Gilbert and Erdős-Rényi random graph models, which already encode some of the qualitative behavior of real-world systems such as the small-world effect. Then we present the configuration model, that extends the previous models to reproduce graphs with a general degree distribution, and the Watts-Strogatz model, with a high level of clustering not present in the other models. Other important models that have been thoroughly studied in the literature include preferential attachment like the de Solla Price model ([de Solla Price, 1965](#)) and the Barabási-Albert model ([Barabási, 1999](#)), the hidden variables model ([Goh et al., 2001](#)) and exponential random graphs ([Holland and Leinhardt, 1981](#)).

### 2.2.1 Random graphs

A random graph is a statistical ensemble  $\mathcal{G}$  of graphs, with a probability  $P(G)$  assigned to every graph  $G$  in the ensemble. The two simplest and most commonly studied models of random graphs are those of [Gilbert \(1959\)](#) and [Erdős and Rényi](#)

(1959). They are commonly called classical random graphs or the Erdős-Rényi graph and treated as the same model, because they are asymptotically equivalent in the thermodynamical limit of infinite vertices,  $N \rightarrow \infty$ .

The **Erdős-Rényi model** (Erdős and Rényi, 1959) is the ensemble of graphs of  $N$  vertices and  $M$  edges, which are chosen uniformly at random among the  $\binom{N}{2}$  pairs of vertices. Hence, the ensemble contains  $\binom{\binom{N}{2}}{M}$  elements. It is maybe the paradigmatic random graph because it is maximally random under a single constraint—that the average degree of the graph is fixed. The **Gilbert model** (Gilbert, 1959), which was already studied by Solomonoff and Rapoport (1951), is the ensemble of graphs of  $N$  vertices, constructed by adding an edge between each of the  $\binom{N}{2}$  pairs of vertices with some probability  $p$ . The ensemble contains  $2^{\binom{N}{2}}$  graphs, each appearing with probability  $p^M(1-p)^{\binom{N}{2}-M}$ , where  $M$  is again the number of edges. The 8 graphs of the  $N = 3$  ensemble are depicted in Figure 2.9. The probability of obtaining a graph with  $M$  edges is

$$\binom{\binom{N}{2}}{M} p^M (1-p)^{\binom{N}{2}-M}, \quad (2.6)$$

which is the reason for this model being also called the Binomial model. The expected value of edges is  $\binom{N}{2}p$ . In statistical mechanics terminology, the Erdős-Rényi model is the analog of the canonical ensemble, and the number of edges  $M$  plays the role of the number of particles, while the Gilbert model is that of the grand canonical ensemble<sup>3</sup>. The analysis of the Gilbert model is technically easier than that of the Erdős-Rényi, but for simulations it is generally more convenient to create instances of the latter model. This is because if  $p$  is small enough (that is, if the graph is dense, as we will explain in a moment), one needs to generate only  $O(N)$  random edges, instead of checking the existence of all  $O(N^2)$  possible edges. Unless explicitly differentiated, from now on we will use the same term *Erdős-Rényi* for these two models.

Since a random graph is an ensemble of many single instances, it can happen that some of these instances have a given structural property while others do not. For this reason, one is usually interested in “typical” structures in the ensemble. Structural properties are “typical” if the probability of finding them in a given instance tends to 1 for  $N \rightarrow \infty$ , and one then says that **almost all** graphs in the ensemble have this property. Quite surprisingly, Erdős and Rényi (1959, 1960) observed that for a number of these properties, there exists a threshold parameter (or more exactly, a threshold function of  $N$ ) below which almost no graph has the property while above which almost all have it.

For some property  $O$  of a random graph, we can also calculate its average over the ensemble,

$$\langle O \rangle_G = \sum_{G \in \mathcal{G}} P(G) O(G), \quad (2.7)$$

<sup>3</sup>See, for example, Dorogovtsev (2010, pp. 9–10).

where  $O(G)$  is the value of the property in the graph realization  $G$ . In real scenarios, it is usually impractical or even impossible to reconstruct several graph realizations. Hence, in most real scenarios, but also in many theoretical models, only a single, large graph is studied. Some of the properties of the graph are **self-averaging**, meaning that for large systems (in the limit  $N \rightarrow \infty$ ) a property of a given graph realization  $G$  is the same as the average over different realizations of  $G \in \mathcal{G}$ . This happens when the property  $O$  is sharply peaked in its average value, and the graph is large enough to make fluctuations around the average vanish. Self-averaging is a very common feature (or working assumption, as we will later discuss) in disordered systems such as spin glasses (Mézard et al., 1987, pp. 7–8) and neural networks (Amit, 1989, pp. 187–189).

One of the basic properties of random graphs is their **degree distribution**,  $p_k$ . It is the probability that a randomly chosen vertex has degree  $k$ ,

$$p_k = \left\langle \frac{n_k(G)}{N(G)} \right\rangle_G, \quad (2.8)$$

averaged over the random graph. Here,  $n_k(G)$  is the number of vertices with degree  $k$  and  $N(G)$  the total number of vertices of the graph  $G$ . As we previously mentioned, sometimes a single, large network instance is studied and one then counts the frequency that a vertex has degree  $k$ . Although its proper name would be **degree sequence**, this is commonly referred also as its degree distribution, and the self-averaging assumption is made so it equals  $p_k$  in the limit of large networks.

However, self-averaging is not necessarily a given, and it should be checked for every particular model. In this regard, the simple example provided in the appendix of Bialas and Oles (2008) is very illustrative. Consider  $\mathcal{G}(N, k)$ , the ensemble of all  $k$ -regular graphs with  $N$  vertices (ie where all vertices have the same degree  $k$ , and their degree distribution is  $p_q = \delta_{q,k}$ ), and define a new ensemble  $\mathcal{G}(N)$  as the union of all  $\mathcal{G}(N, k)$ ,

$$\mathcal{G}(N) = \bigcup_k \mathcal{G}(N, k), \quad (2.9)$$

with probability

$$P(G) = \frac{w_k}{|\mathcal{G}(N, k)|} \quad (2.10)$$

assigned to every graph  $G \in \mathcal{G}(N)$ . Here  $|\mathcal{G}(N, k)|$  is the number of graphs in the ensemble  $\mathcal{G}(N, k)$  and  $w_k$  is an arbitrary probability distribution, with  $\sum_k w_k = 1$ . One can see that the degree distribution is equal to  $w_k$ :

$$p_q = \left\langle \frac{n_q(G)}{N(G)} \right\rangle_G = \sum_{G \in \mathcal{G}(N)} \frac{n_q}{N} P(G) = \sum_k \sum_{G \in \mathcal{G}(N, k)} \frac{w_k \delta_{k,q}}{|\mathcal{G}(N, k)|} = \sum_k w_k \delta_{k,q} = w_q. \quad (2.11)$$

This distribution is clearly not self-averaging, as single instances of a graph should be defined by a single degree  $k$ , not by an (arbitrary)  $w_q$ . That the self-averaging does not hold can be seen more rigorously by computing the variance of  $p_q$ :

$$\begin{aligned} \text{Var } p_q &= \sum_{G \in \mathcal{G}(N)} \left( \frac{n_q}{N} - w_q \right)^2 P(G) = \sum_k \sum_{G \in \mathcal{G}(N,k)} \frac{w_k (\delta_{k,q} - w_q)^2}{|\mathcal{G}(N, k)|} \\ &= \sum_k w_k (\delta_{k,q} - w_q)^2 = w_q - 2w_q^2 + w_q^2 \sum_k w_k = w_q - w_q^2. \end{aligned} \quad (2.12)$$

This quantity does not vanish on the limit  $N \rightarrow \infty$ , explaining that  $p_q$  (the degree distribution of the whole ensemble) is not a good description of a single graph. For a discussion of self-average in real-world networks, see for example the study of [Serrano et al. \(2007\)](#) of the World Wide Web. As is commonly done, in this thesis we will consider self-averaging as a working hypothesis and later confirm it with numerical results.

In the Gilbert model, a vertex has degree  $k$  according to the binomial distribution,

$$p_k = \binom{N-1}{k} p^k (1-p)^{N-1-k}, \quad (2.13)$$

and the mean degree is  $\langle k \rangle = p(N-1)$ . Usually we are interested in very large networks. In this case, a **dense** graph is that for which the number of edges is close to the maximum  $\binom{N}{2}$ , ie the mean is of the order of  $N$ . On the contrary, in a **sparse** graph,  $\langle k \rangle / N \rightarrow 0$  as  $N \rightarrow \infty$  and  $\langle k \rangle$  tends to a constant. Setting  $c \equiv \langle k \rangle$ , the degree distribution of a sparse graph becomes

$$p_k = \frac{c^k}{k!} e^{-c}. \quad (2.14)$$

Hence, in the limit of large  $N$ , where the Gilbert and Erdős-Rényi models coincide, they have a Poisson degree distribution. These random graph models are completely characterized by this distribution. In fact, the only constraint imposed is that their mean degree is equal to  $c$ , and this is the reason why they are also called maximally random graphs.

The degree distribution is indeed one of the most fundamental properties of a random graph, and many of its features can be derived from it. As an example, from the Poisson degree distribution of the Erdős-Rényi model, one can easily obtain the logarithmic scaling of the average path length with respect to the size of the network—that is, the small-world effect introduced in [Section 2.1.4](#). This is done by the following reasoning. Starting from a randomly selected vertex, the number of first-neighbors (vertices at distance 1) is on average  $c$ . These first-neighbors have on average  $c$  new neighbors each, so the number of first- and second-neighbors of the initial vertex is  $c+c^2$ . To distance  $l$ , the total number of neighbors

is  $c + c^2 + \dots + c^l$ . The average path length  $l_{\text{av}}$  is approximately that at which the total number of neighbors up to that distance is equal to the size of the network,

$$1 + \sum_{l=1}^{l_{\text{av}}} c^l = N. \quad (2.15)$$

If we consider  $c \gg 1$ , this expression simplifies to  $c^{l_{\text{av}}} = N$ , ie

$$l_{\text{av}} = \frac{\log N}{\log c}. \quad (2.16)$$

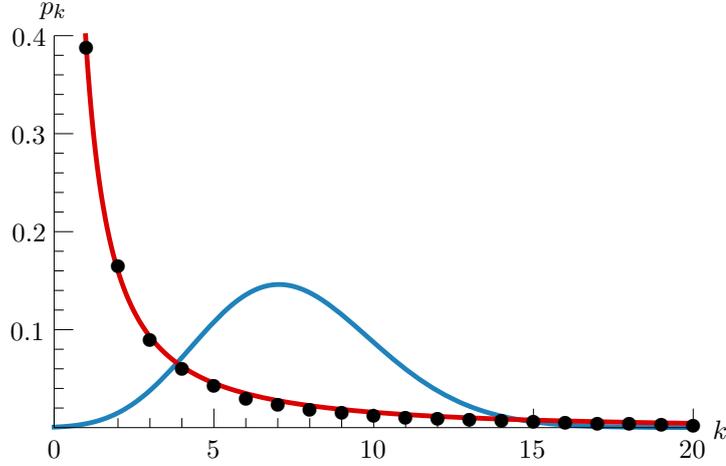
This is an approximation in two senses: we have neglected the possibility that there are loops that go back to vertices that are already counted, and we have assumed that the full network ( $N$  vertices) can be reached. Nevertheless, it gives a good estimate of the scaling of  $l_{\text{av}}$  and shows the small-world effect in the Erdős-Rényi model.

However, as we already pointed out at the beginning of this Section, when the first real-world complex networks were confronted to this simple model, significant differences arose even in their degree distributions. In the citation network of [de Solla Price \(1965\)](#), for example, it was already observed that the number of citations—the degree—followed a power-law distribution. We illustrate this behavior in [Figure 2.10](#), where we confront the Poisson degree distribution of [Eq. 2.14](#) and a power-law distribution with another real-world network that we will later introduce in [Section 2.2.4](#). Networks with such degree distributions are called scale-free networks and have many vertices of low degree but also a small number of very high-degree vertices, called **hubs**. This observation is, in fact, extremely common in many natural, social and artificial systems, and triggered the generalization of the Poisson random graph to more general degree distributions.

### 2.2.2 The configuration model

The configuration model ([Bender and Canfield, 1978](#); [Bollobás, 1980](#)) is a way to obtain random graphs with any degree distribution. More precisely, it is the ensemble of graphs with a particular degree *sequence*: the number of vertices with a specific degree is fixed, instead of having a degree according to a probability distribution. Since the degree sequence is fixed, so it is the number of edges,  $M = \frac{1}{2} \sum_u k_u$ . In a way, this makes the configuration model a generalization of the Erdős-Rényi model, with fixed  $M$ .

The algorithm to create a graph instance is the following. Start with  $N$  isolated vertices, and assign a random integer  $k_u$  to each of them, according to the desired degree distribution  $p_k$ . Each vertex  $u$  can be thought as having  $k_u$  “stubs”, or half-edges. Then, pairs of stubs are randomly selected and joined between them to create the  $M$  edges. Two considerations are in order. First, not all degree sequences are possible, and to successfully create the graph the sum of degrees must be even,



**Figure 2.10.** Comparison between a Poisson  $c = 7.54$  (blue) and a scale-free  $\tau \approx 1.23$ ,  $\kappa \approx 23.8$  (red) degree distributions  $p_k$ , given by Eqs. 2.14 and 2.19. The parameters of the degree distributions have been set to fit the real-world network described in Section 2.2.4, to which the black dots correspond. The power-law degree distribution has a long tail that extends far beyond the degree  $k = 20$  shown in the plot.

$\sum_u k_u = 2M$ . Second, the model allows the creation of self-edges and multiedges, so the ensemble consists of all the multigraphs with a specific degree sequence, each equiprobable. This is generally not a problem, as the number of self-edges or multiedges typically remains constant as  $N$  grows. When needed, one can also explicitly forbid the creation of self-edges or multiedges in the random edge creation process.

With this method, any degree distribution can be modeled. As we already mentioned, real-world networks do not have the Poisson degree distribution of the Erdős-Rényi model, but rather exhibit a power-law (scale-free) degree distribution,

$$p_k = Ck^{-\tau}, \quad (2.17)$$

with  $\tau$  a positive constant and  $C$  a normalization factor. Such degree distributions are characterized by a relatively important number of vertices with a degree much greater than the average. In heavy-tailed networks with these degree distributions, a cutoff in the maximum degree naturally appears in scenarios where very high degrees cannot exist due to, for example, targeted attacks, physical constraints, saturation effects, or simply the finiteness of the network size. To consider this, one can directly include a sharp cutoff,

$$p_k = \begin{cases} Ck^{-\tau} & \text{for } k \leq k_c \\ 0 & \text{for } k > k_c \end{cases}, \quad (2.18)$$

or instead consider scale-free degree distributions with an exponential cutoff,

$$p_k = Ck^{-\tau} e^{-k/\kappa}. \quad (2.19)$$

(Note that the normalization factor is different for every distribution.) The pure scale-free behavior can still be recovered by taking the  $\kappa \rightarrow \infty$  limit. We will later see that the cutoff  $\kappa$  strongly affects the network properties.

Unlike the Erdős-Rényi, the configuration model succeeds in reproducing the degree distribution of real-world networks. However, neither of them reproduce another very basic property: clustering. Indeed, in these models, the probability that any two vertices are neighbors is the same—independently of whether they share a common neighbor or not. Hence, for the Gilbert model the clustering coefficient is

$$\langle C \rangle = p = \frac{c}{N-1}, \quad (2.20)$$

which tends to zero for large networks. For a general uncorrelated degree distribution it is (Newman, 2010, p. 449)

$$\langle C \rangle = \frac{1}{N} \frac{(\langle k^2 \rangle - \langle k \rangle)^2}{\langle k \rangle^3}. \quad (2.21)$$

For networks of finite  $\langle k \rangle$  and  $\langle k^2 \rangle$ , this quantity tends to zero in the limit of large network size  $N \rightarrow \infty$ . On the contrary, typical values of  $\langle C \rangle$  in real-world networks are quite high: between 0.18 and 0.3 in the Internet at the domain level (Pastor-Satorras et al., 2001; Yook et al., 2002), and in the range 0.066–0.76 for different scientific collaboration networks (Newman, 2001a,b,c,d; Barabási et al., 2002). A large list of networks and their clustering coefficients, together with other properties, can be found in Newman (2010, p. 237).

Several algorithms and models have been proposed to produce random graphs with clustering; see, for example, that of Serrano and Boguñá (2005, 2006a,b). The configuration model itself was recently generalized to incorporate clustering (Newman, 2009) in an analytically solvable model. One of the first and more popular models, however, is the so-called “small-world model” (Watts and Strogatz, 1998), a model that spans between regular lattices and maximally random graphs.

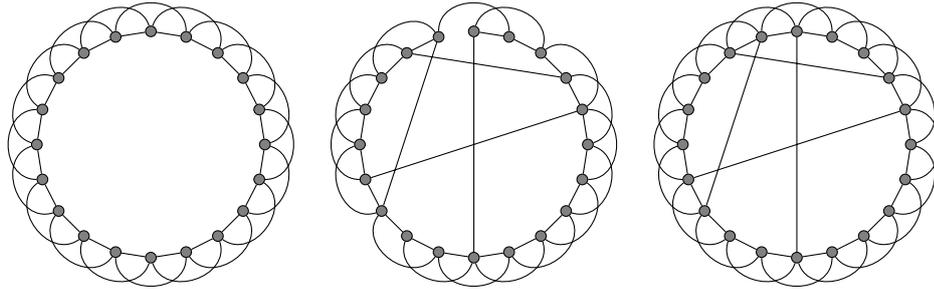
### 2.2.3 Small worlds

Aside from regular lattices, which have a high level of clustering, there are also many real-world networks with this property. This is especially true for social networks, but also for communication and biological networks. Watts and Strogatz (1998) introduced, in a paper that has become very influential, the **Small-world model**<sup>4</sup>, in an attempt to find a model that combined a significant level of clustering

<sup>4</sup>One should not confuse the *small-world effect* of very short average path lengths, which is common in complex networks, and the *small-world model*, that is the name of the model introduced by Watts and Strogatz (1998). To avoid confusion, we usually refer to the latter as the Watts-Strogatz model, but the original name of Small world is also very common in the literature and will sometimes appear here. For an (early) review of several small-world models, see for example Newman (2000).



**Figure 2.11.** Base lattice for the Watts-Strogatz model. All vertices are connected to their  $2k$  nearest vertices.



(a) Base lattice.

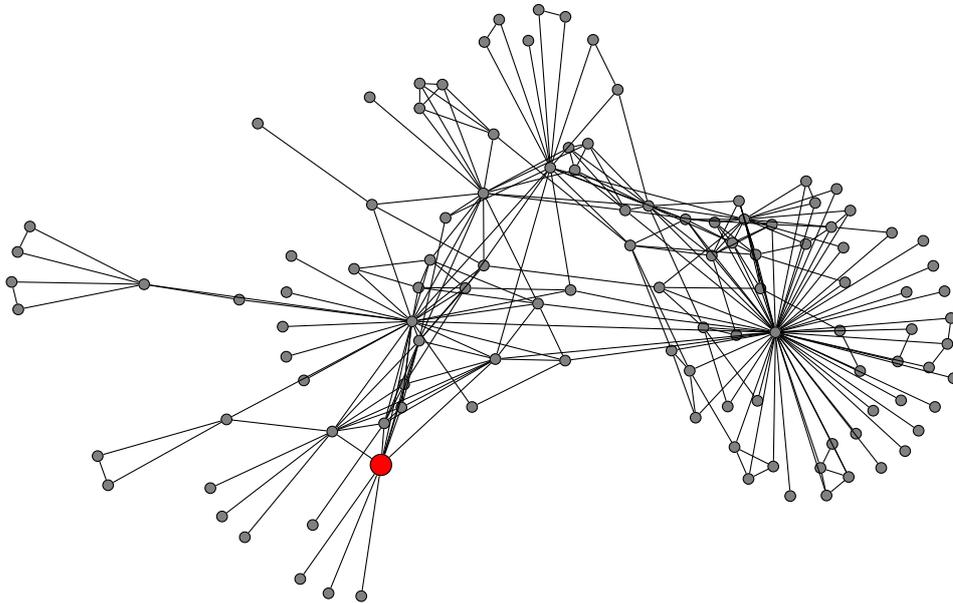
(b) Rewired WS.

(c) Added WS.

**Figure 2.12.** Watts-Strogatz (WS) model, with  $N = 20$  and  $k = 2$ . (a) The base lattice of Figure 2.11, shaped as a ring (with periodic boundary conditions). All vertices are connected to their  $2k$  nearest vertices, and there is a total of  $kN$  edges. (b) In the rewired WS, edges of the base lattices are rewired to a random vertex with probability  $\beta$ . The total number of edges remains constant, independently of  $\beta$ . (c) In the added WS, with probability  $\beta$  a new random edge is added for each of the  $kN$  edges of the base lattice. In average, the total number of added edges is  $\beta kN$ .

as well as a very short average path length, as most biological, technological and social networks do. This model is a random graph with an ordered local structure and a high level of clustering but still with a surprisingly low average path length. **Watts and Strogatz** observed that regular lattices usually have a high clustering but suffer the lack of very large average path length, while in the Erdős-Rényi and the configuration model, as we have already seen, the clustering coefficient tends to 0 for large networks, but the average path length grows only logarithmically. Their intermediate model, which have later been called **Watts-Strogatz model** so it is not confused with the small-world property common of most complex networks, starts from a regular network and rewires some of the edges to introduce randomness. These rewired edges act as shortcuts between highly connected parts of the network, so the average path length is strongly decreased while a high level of clustering is maintained.

The model is constructed as follows. It starts with a one-dimensional regular base lattice of  $N$  vertices. Each vertex has  $2k$  edges that connect it to its nearest vertices (see Figure 2.11), so there is a total of  $kN$  edges. It is common to consider periodic boundary conditions, so the base lattice takes the shape of a ring as in

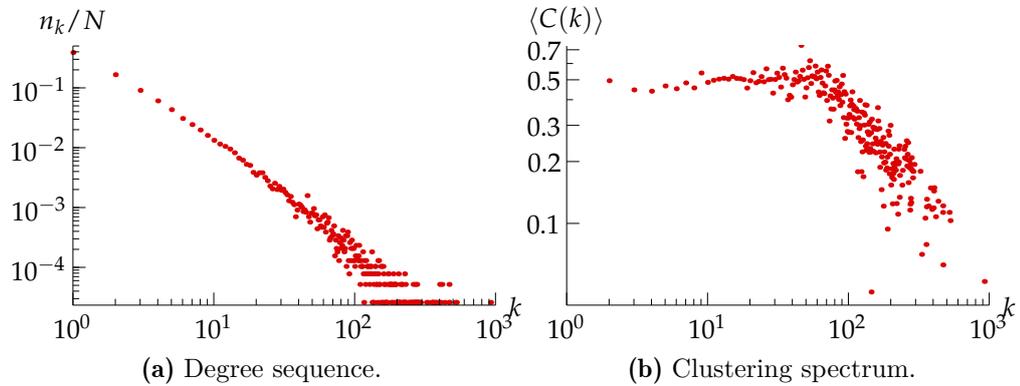


**Figure 2.13.** Network of all keys in the OpenPGP Web of Trust at a distance three or less from a key (bigger, in red) of the author of this thesis, and signatures between them.

Figure 2.12a. To decrease the average path length and introduce randomness, there are two slightly different versions of the model. In the original, depicted in Figure 2.12b, each edge in the base lattice is *rewired* to a random vertex with probability  $\beta$ . We call this the **rewired Watts-Strogatz model**, or rewired WS for short. In the second version (Figure 2.12c), for each edge in the base lattice we *add* a new random edge with probability  $\beta$ . We call these new edges **shortcuts**, and the version **added Watts-Strogatz model**, or added WS. Both versions have the same qualitative behavior, but for analytical treatment it may be convenient to use the latter (Moore and Newman, 2000).

#### 2.2.4 A real-world example: the OpenPGP Web of Trust

As an example of a real-world network, we present here the OpenPGP Web of Trust that we will use later in Chapter 3. The **Web of Trust** (see Figure 2.13 for a partial view) is a social network representing the trust between OpenPGP users, and is conceived to solve the authentication problem. Without going into much detail, OpenPGP is a standard encryption protocol for securing email communications using public key cryptography (Schneier, 1996). It is derived from PGP (Pretty Good Privacy), a computer program created by Phil Zimmerman in 1991. In public key protocols, users have a pair of keys: one which is public and accessible by everybody, and a private one which is used for decryption. If the sender user



**Figure 2.14.** Web of Trust of the classical cryptographic protocol OpenPGP. (a) Log-log plot of the degree sequence of the Web of Trust, as described in the text.  $n_k$  is the number of vertices of degree  $k$ , and follows a power law. (b) Log-log plot of the mean clustering coefficient for a specific degree,  $\langle C(k) \rangle$ . The size of the network is  $N = 38\,550$  and the number of edges  $M = 145\,388$ .

(let us call her Alice) wants to send a secure message to the receiver (Bob), she has to use Bob's public key to encrypt it. The authentication problem arises when Alice cannot verify if the key she is using is really owned by Bob. A solution to this problem is the Web of Trust, in which every user signs a public key if she trusts it. This process generates a directed network, where nodes correspond to public keys and a directed edge from key  $A$  to  $B$  means that the owner of key  $A$  has trusted and signed key  $B$ . To trust a key, usually a user has to meet with the key owner and check that he is really who he claims to be. This social model is relevant to quantum communication in the sense that at this point the two users could use the interaction to create some bipartite entanglement between them, and then separate, each keeping one of the parts. By repeatedly doing so between different pairs of users, as in the Web of Trust, a bipartite quantum network would be created.

Here we use the strongly connected component of the Web of Trust obtained from the Swiss keyserver<sup>5</sup> as of May 25, 2010, containing 41 459 keys and 424 577 signatures. In Chapter 3 we will study entanglement distribution in undirected networks. Hence, we consider only bidirectional edges, corresponding to users who mutually signed their keys. This leaves an undirected graph with 38 550 keys and 145 388 two-way signatures (Cuquet and Calsamiglia, 2011). The degree sequence (Figure 2.14a) is a power law, with mean degree  $\langle k \rangle = 7.54$  and  $\langle k^2 \rangle = 501$ .

A fit of the degree sequence to the power-law distribution with an exponential cutoff of Eq. 2.19 gives  $\tau \approx 1.23$  and  $\kappa \approx 23.8$ . As shown in Figure 2.10 on page 22, this distribution reproduces quite well the degree sequence of the Web of

<sup>5</sup>The Swiss Keyserver used to be accessible at <http://www.keys.ch.pgp.net:11371/pks/>, which now seems to be defunct. Public data from 2005 to 2012 is available at <http://www.lysator.liu.se/~jc/wotsap/index.html>.

Trust. However, the average clustering is  $\langle C \rangle = 0.290$ , high compared to that of a random graph of the same size and mean degree,  $1.95 \cdot 10^{-4}$  (see Eq. 2.20), and even to the one corresponding to the configuration model,  $1.65 \cdot 10^{-3}$  (see Eq. 2.21). Figure 2.14b shows the spectrum of the local clustering coefficient  $\langle C(k) \rangle$ , where the average is performed over nodes with the same degree:

$$\langle C(k) \rangle = \sum_{u \in V: k_u = k} C_u. \quad (2.22)$$

The scaling law of the clustering coefficient  $C(k)$  for high degrees suggests that this network has a hierarchical structure (Dorogovtsev et al., 2002; Ravasz and Barabási, 2003).

## 2.3 Percolation on complex networks

One of the primary features of networks is the existence, or not, of a giant connected component whose size is of the order of the size of the network,  $N$ . In general, a network can be composed of many disconnected components, each of a size that is small compared to  $N$ , or have an extensive component that in the asymptotic limit of very large networks has a size that is a finite fraction,  $S$ , of the total network size. That the network is in one of these two possible regimes depends on its structural parameters. For example, in the Erdős-Rényi model (Erdős and Rényi, 1959), for a mean degree smaller than unity,  $c < 1$ , there are only small components, the biggest of them of order  $\log N$ . Instead, for  $c > 1$ , the biggest component is of size  $SN$ ,  $S$  being a positive number independent of  $N$ . When the size of this biggest component is of order  $N$  it is called the giant connected component. The fraction  $S$  of nodes in the network that belong to the biggest component—from now on, we will call  $S$  simply the size of the giant component—is in fact the order parameter of a second order phase transition. In the thermodynamic limit, the critical point of this phase transition (in the case of the Erdős-Rényi model,  $c = 1$ , with a biggest component of order  $N^{2/3}$ ), marks the transition from a phase with  $S = 0$  to a phase with  $S > 0$ .

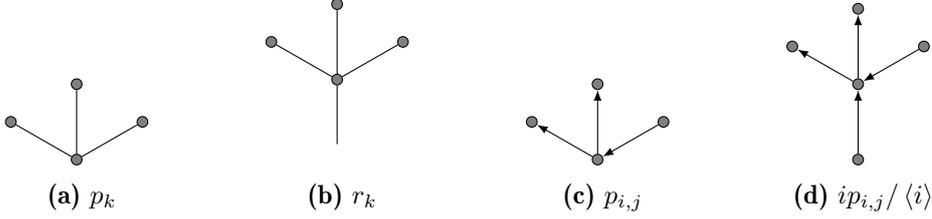
This concept is closely related to that of a percolating cluster. Percolation theory (Grimmett, 1989; Stauffer and Aharony, 1994) began as the study of how fluids filter through a porous medium, just like petroleum moves through rocks. In the last decades, it has become the subject of applied mathematics and graph theory, and found applications in a broad number of areas such as physics, material science, geography, communication technologies, as well as medical, biological and social sciences. In **bond percolation**, edges in a network can be occupied with a given probability, therefore connecting their end vertices, or empty, which means that the edge is not useful to connect directly the two nodes (which can, however, still be connected by a different, longest path). Similarly, in **site percolation**, vertices are the elements which can be occupied or empty with some probability.

Both processes give rise to similar phenomena. In particular, there exists a **percolation threshold** in the occupation probability: below a critical probability, all connected components are of finite size, while above it there exist only one giant connected component. In terms of percolation theory, which initiated as the study of the components—called clusters—in a regular lattice, the equivalent of the giant component is a **percolating cluster** that extends from one end to the other of the lattice. As before, this threshold is the critical point of a phase transition (with order parameter  $S$ ), and can be marked by the divergence of the average component size, which acts as a susceptibility in a magnetic material.

Percolation theory has direct applications in communicating over a network. A network may have a giant component by itself or even be completely connected, but sometimes these connections can fail, eg due to some noise, random failures or directed attacks. To be able to communicate in a network like this, two vertices need to establish a path of occupied (reliable) edges between them. In other words, they must belong to the same connected component. For very large networks, the relative size of the components tend to zero, except for the giant component, in the case it exists. Therefore, in the large network limit two vertices are connected if they both belong to the giant connected component—because other components have a negligible size. This happens with probability  $S^2$ . Of course, this probability is zero below the percolation threshold. Hence, solving this threshold and the size of the giant component as a function of the occupancy probability is crucial in determining the ability of a network to support communication.

Let us here remark the difference between the two ideas that we have introduced. The first one is related to structural properties of a network: the size and the number of its components. The second, on the other hand, is a process on such networks, which may have originally a giant connected component or be already disconnected. Of course, they are related, and the first can be viewed as a kind of percolation process in the complete graph.

The percolation threshold and the size of the giant connected component, as well as many other properties, strongly depend on the basic structure of the network (Moore and Newman, 2000; Callaway et al., 2000; Newman et al., 2001; Dorogovtsev et al., 2008) as well as on degree-degree correlations (Boguñá et al., 2003; Goltsev et al., 2008) and clustering (Newman, 2009). Therefore, a change in the structure of a network can affect its ability to communicate information. For example, the scale-free topology of Internet makes it resilient against the failure of random nodes (Cohen et al., 2000), but not against targeted attacks directed to its major hubs (Albert et al., 2000; Cohen et al., 2001). This relation between the structure of the network and the communication over it can be also exploited to benefit the earlier appearance of the giant cluster and to find architectures that allow communication even in the presence of noise. For regular lattices, only a few analytical results are known, mainly regarding the percolation threshold and critical exponents of the phase transition (Grimmett, 1989), but not for the size of the giant component. Hence, one usually turns to numerical simulations to explore



**Figure 2.15.** (a) Degree probability  $p_k$  (in this case, a node with degree 3). (b) Excess degree probability  $r_k$  (here, a node with excess degree 3: the neighbor used to arrive at the node is not counted). (c) Directed (in- and out-) degree probability  $p_{i,j}$ . (d) Directed (in- and out-) degree probability  $ip_{i,j}/\langle i \rangle$ , arriving from an edge (in this case, the in- and out-degrees are both 2).

these systems (Newman and Ziff, 2000). On the contrary, and despite the apparent intricate structure of complex networks, basic quantities like the threshold and the giant component size can be computed exactly for many different models, including the ones presented in Section 2.2.

In this Section we review the component structure and the main results of percolation in complex networks, and introduce the mathematical methods that are used to obtain them. This will come in handy in Chapter 3, where we study the distribution of bipartite entanglement in complex networks, a process that is closely related to percolation. The mathematical tools include primarily probability generating functions. Apart from its key role in the analytical treatment of percolation, they are a powerful and versatile tool that will also be used in Chapter 4, not only in the framework of complex networks but also to solve other combinatoric problems.

### 2.3.1 Components in uncorrelated networks

#### Networks with uncorrelated degree distributions

In Section 2.2.1, we introduced the degree distribution (Eq. 2.8) of a random graph,  $p_k$ . As shown in Figure 2.15a, it states what is the probability that, picking up a random vertex, it has degree  $k$ . There is another probability that, although related to  $p_k$ , does not coincide with it. Suppose that, instead of randomly selecting a vertex, we take an edge at random and follow it to one of its end vertices. The probability that this vertex has degree  $k$  depends on  $p_k$  and on its own degree, and can be easily calculated using the stubs picture of the configuration model. If we pick up one of the stubs at random, it will be attached to a specific vertex of degree  $k$  with probability  $k/2M$ . Since there are  $Np_k$  of such vertices, the probability that the stub connects to any vertex of degree  $k$  is

$$\frac{k}{2M}Np_k = \frac{kp_k}{\langle k \rangle}. \quad (2.23)$$

This probability reflects that it is more likely to arrive at a vertex with higher degree (and that the probability of arriving at an isolated vertex through an edge is, of course, zero). Using Eq. 2.23, we can easily calculate the average degree of a neighbor as the sum of  $k$  times the probability that the neighbor has degree  $k$ :

$$\sum_k k \frac{kp_k}{\langle k \rangle} = \frac{\langle k^2 \rangle}{\langle k \rangle}. \quad (2.24)$$

This average is only equal to  $\langle k \rangle$  if all nodes are of the same degree, and larger otherwise.

In general, the degrees  $k_u$  and  $k_v$  of the two ends of a random edge  $(u, v)$  can be correlated. **Uncorrelated networks** are networks where the probability  $p_{k_u, k_v}$  of this event can be factorized into the probabilities of finding vertices of degree  $k_u$  and  $k_v$  independently,

$$p_{k_u, k_v} = \frac{k_u p_{k_u}}{\langle k \rangle} \frac{k_v p_{k_v}}{\langle k \rangle}. \quad (2.25)$$

The Erdős-Rényi and the configuration models have uncorrelated degree distributions.

Many calculations are performed as if we were traversing the network, arriving at a vertex through an edge and leaving it through one, or all, the remaining edges of this vertex. The number of edges of a vertex different from the one used to arrive at it is the **excess degree** of the vertex (see Figure 2.15b). This quantity is just one less than the total degree. Hence, the probability  $r_k$  of having excess degree  $k$  comes from Eq. 2.23, as the probability of having total degree  $k+1$ . The probability distribution  $r_k$  is

$$r_k = \frac{(k+1)p_{k+1}}{\langle k \rangle}. \quad (2.26)$$

Note that it is properly normalized,  $\sum_{k \geq 0} r_k = 1$ . To keep notation simple, we will use  $k$  indistinctly for the degree and the excess degree. It will be clear by context, or by explicitly expressing it, to which of the two probabilities ( $p_k$  and  $r_k$ , respectively) the quantity  $k$  is associated. Note also that  $\langle k \rangle$  is always the mean degree,  $\langle k \rangle = \sum_k k p_k$ .

We can also consider directed complex networks, where edges can be traversed only in one specific direction. If edges are directed, then the probability degree distribution has to differentiate between the in- and out-degree. We denote them as  $i$  and  $j$ , respectively, and its corresponding degree distribution is  $p_{i,j}$  (Figure 2.15c). As for every edge there is a source and a target vertex, this distribution is constrained by

$$\sum_{i,j} (j-i)p_{i,j} = 0, \quad (2.27)$$

which means that the average in- and out-degree are the same, and half of the total degree  $k = i + j$ :

$$\langle i \rangle = \langle j \rangle = \frac{\langle k \rangle}{2}. \quad (2.28)$$

By a similar reasoning as before, the probability that following a random, *directed* edge to its target vertex, this vertex is of in- and out-degree  $i, j$  (Figure 2.15d), is

$$\frac{i}{M} N p_{i,j} = \frac{i p_{i,j}}{\langle i \rangle}. \quad (2.29)$$

### Tree-like approximation

We have already seen that the local clustering coefficient of an Erdős-Rényi graph is (Eq. 2.20):

$$\langle C \rangle = \frac{c}{N-1} \simeq \frac{c}{N}.$$

In this model,  $C = \langle C \rangle$ , and the number of connected triples is  $N \frac{\langle k^2 \rangle - \langle k \rangle}{2}$ . Using Eq. 2.3 and that in a Poisson distribution  $\langle k^2 \rangle - \langle k \rangle = \langle k \rangle^2$ , the number of triangles (cycles of length 3) is

$$\frac{\langle k \rangle^3}{6}. \quad (2.30)$$

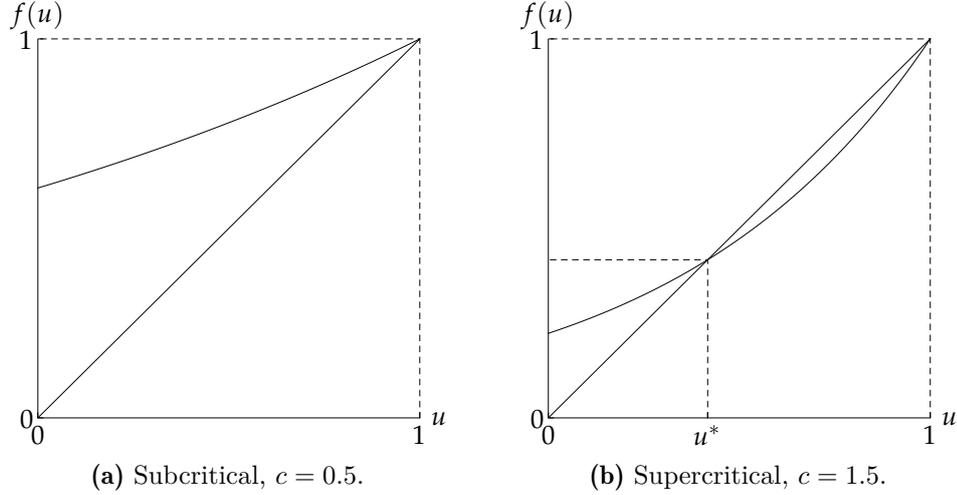
More generally, in an uncorrelated network, the number of cycles of length  $l$  is (Bianconi and Marsili, 2005)

$$\frac{1}{2l} \left( \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \right)^l, \quad (2.31)$$

valid for small cycles. This means that, in the limit of very large graphs, the number of finite cycles in the vicinity of a given vertex of an Erdős-Rényi graph tends to zero, and up to some distance from it the network is a tree. A network with this property is called **locally tree-like**, and it turns out to be a very useful one to compute various network quantities. General uncorrelated networks are locally tree-like as long as the second moment of the degree distribution does not diverge. Note however that there are still many long cycles. If not, the network would be disconnected by just the removal (or failure) of one edge.

### Small components and the giant component

In Section 2.1.5 we saw that a graph can be either connected or disconnected, and in particular that it can have many components. A general property of random graphs, that was already studied in the first papers by Solomonoff and Rapoport (1951), Gilbert (1959) and Erdős and Rényi (1959, 1960), is whether a graph has many small components or there exist the so-called giant component, which spans a significant fraction of the network. Taking the Gilbert model as an example, for  $p = 0$  (corresponding to  $c = 0$ ) the network consist only of isolated vertices, and hence there are as many components as vertices. On the other extreme, for  $p = 1$ , the network is fully connected so there is only a single component that includes all the vertices. What happens in between? In fact, for there to be a



**Figure 2.16.** Graphical solution of Eq. 2.32 of the Erdős-Rényi network. (a) In the subcritical phase,  $c < 1$ , there is only the trivial solution  $u^* = 1$ . (b) In the supercritical phase,  $c > 1$ , there is a second, smaller solution (in this case,  $u^* \approx 0.42$ ), which indicates that the probability of finding a finite component is smaller than one, and hence there exists a giant connected component.

single component, the network does not need to be fully connected. If we relax the condition even more, so we ask if there is a component of the order of the network, it turns out that even in the sparse regime (constant  $c$ ) the graph has such component. (For example, for  $c = 5$  the giant component contains around 99.3% of the vertices of the network.) It turns out that for  $c < 1$  all components are small (more exactly, they have size at most  $\log N$ ), while for  $c > 1$  there is a giant component of size  $SN$ , where  $S$  is a constant. The transition from the phase with  $S = 0$  and that with  $S > 0$  is continuous, and thus the fraction of nodes  $S$  in the giant component acts as an order parameter of a second-order phase transition, with critical point at  $c = 1$ .

In a general uncorrelated network, this critical point can be found by considering the probability  $u$  that, randomly selecting an edge and following to one of its ends, it does not connect to the giant component (or, equivalently, it connects to *any* finite component). There exists a giant component in the network if and only if this probability is strictly less than 1. The excess degree of the vertex reached through the randomly selected edge is  $k$  with probability  $r_k$ , so the probability  $u$  can be expressed in a self-consistent equation as

$$u = \sum_{k \geq 0} r_k u^k. \quad (2.32)$$

Note that the function  $f(u) = \sum_k r_k u^k$  always fulfills  $f(1) = 1$  (because the probability distribution  $r_k$  is correctly normalized), so  $u^* = 1$  is always a trivial solution. Note also that  $f(0) = r_0 > 0$  (as long as the probability  $p_1$  of a vertex to

have degree 1 is nonzero), and therefore Eq. 2.32 has a non-trivial solution (and the network, a giant component) when the slope of  $f(u)$  at  $u = 1$  is greater than the slope of  $u$ , ie when

$$f'(1) > 1. \quad (2.33)$$

Figure 2.16 shows that there exists only the trivial solution in the subcritical phase of the Erdős-Rényi model, while in the supercritical phase a second, non-trivial solution  $u^* < 1$  appears.

Using Eq. 2.26 together with the previous equation one arrives at the criterion of Molloy and Reed (1995) for the existence of a giant component:

$$\langle k^2 \rangle - 2 \langle k \rangle > 0. \quad (2.34)$$

Roughly, it says that there will be a giant component if the average number of second neighbors is greater than that of first neighbors, which is quite remarkable as it depends only on the local structure of the network.

### 2.3.2 Generating functions

The Molloy-Reed criterion for the critical point can also be found using the probability generating function formalism presented by Callaway et al. (2000) and extended in Newman et al. (2001). This formalism is very powerful and allows to compute a variety of network properties for uncorrelated degree distributions, such as the size of the giant component and the distribution of component sizes. It has also been applied to models with clustering (Newman, 2009) and the Watts-Strogatz model (Moore and Newman, 2000). In this Section, we introduce generating functions, and in the following Sections we present their specific application to percolation in random networks. A very nice review of generating functions and random graphs can be found in Newman (2003b) and Newman (2010).

Probability generating functions (Wilf, 2006) are another way to express probability distributions. They are defined as a power series whose coefficients are the probability mass function of a discrete random variable. As Wilf visually describes them in the introduction of his book on generating functions, “a generating function is a clothesline on which we hang up a sequence of numbers for display” (Wilf, 2006, p. 1). They are very helpful for “counting things”, and among other useful properties, they allow for a straightforward convolution of probabilities. Let us define them with an example. Consider the distribution of the probability  $p_k$  that a vertex has degree  $k$ , which is properly normalized so  $\sum_{k=0}^{\infty} p_k = 1$ . The **generating function** of the probability distribution  $p_k$  is

$$g_p(z) = \sum_{k=0}^{\infty} p_k z^k. \quad (2.35)$$

The function  $g_p(z)$  encodes all the information of the distribution, and in particular the probability  $p_k$  can be recovered by taking the  $k$ -th derivative of  $g_p(z)$  at  $z = 0$ ,

$$p_k = \frac{1}{k!} \left. \frac{d^k g_p(z)}{dz^k} \right|_{z=0}. \quad (2.36)$$

The generating function  $g_p(z)$  is correctly normalized,  $g_p(1) = 1$ . It is also convergent for  $|z| \leq 1$ .

The first moment of the distribution  $p_k$ , which corresponds to the average network degree  $\langle k \rangle$ , is just the first derivative evaluated at  $z = 1$ :

$$\langle k \rangle = g'_p(1) = \left. \sum_{k=1}^{\infty} k p_k z^{k-1} \right|_{z=1}. \quad (2.37)$$

This allows us to express the generating function for the excess degree,  $g_r(z)$ , as

$$g_r(z) = \sum_{k=0}^{\infty} r_k z^k = \sum_{k=1}^{\infty} \frac{k p_k}{\langle k \rangle} z^{k-1} = \frac{g'_p(z)}{g'_p(1)}. \quad (2.38)$$

Higher moments can be similarly found by taking more derivatives. In general, the  $n$ -th moment is

$$\langle k^n \rangle = \sum_k k^n p_k = \left( z \frac{d}{dz} \right)^n g_p(z) \Big|_{z=1}. \quad (2.39)$$

Convolution of independent distributions can be obtained simply by multiplication of their respective generating functions. Let us see that with the following simple example: the outcome of tossing a (possibly biased) coin. With probability  $p_0$  the outcome is 0, and with  $p_1 = 1 - p_0$  it is 1. The generating function for this probability is

$$p_0 + p_1 z, \quad (2.40)$$

and  $z$  can be seen as a variable that “counts” the number of “1s”. The generating function of the probability for the total outcome after two coin tosses, labeled (a) and (b), is

$$p_0^{(a)} p_0^{(b)} + p_0^{(a)} p_1^{(b)} z + p_1^{(a)} p_0^{(b)} z + p_1^{(a)} p_1^{(b)} z^2 = [p_0^{(a)} + p_1^{(a)} z] [p_0^{(b)} + p_1^{(b)} z]. \quad (2.41)$$

Similarly, the total number of edges emerging from  $n$  independent vertices (the sum of their degrees) is generated by  $[g_p(z)]^n$ .

Generating functions can also be used with the directed network to express the in- and out-degree distribution  $p_{i,j}$ . In this case, one needs two variables  $x$  and  $y$  to keep track of degrees  $i$  and  $j$ :

$$g_p(x, y) = \sum_{i,j=0}^{\infty} p_{i,j} x^i y^j. \quad (2.42)$$

The probability  $p_{i,j}$  is recovered by

$$p_{i,j} = \frac{1}{i!j!} \partial_x^i \partial_y^j g_p(x, y) \Big|_{\substack{x=0 \\ y=0}}, \quad (2.43)$$

where by  $\partial_x$  we denote the partial derivative with respect to  $x$ . The mean in-degree is

$$\langle i \rangle = \frac{\partial g_p(x, y)}{\partial x} \Big|_{\substack{x=1 \\ y=1}}, \quad (2.44)$$

and the probability in Eq. 2.29 is generated by

$$x \frac{\partial_x g_p(x, y)}{\partial_x g_p(1, 1)}. \quad (2.45)$$

### 2.3.3 Percolation and component sizes

So far we have seen the structural properties of networks with an uncorrelated degree distribution. Depending on the parameters of these distribution, the network can be made of many small connected components, or contain a giant one that spans a significant fraction of the network. But what happens when some of the edges, or vertices, are removed? In this case, an existing giant component might disappear, and new structures emerge. We already introduced that this failure of vertices or edges is a process called site or bond percolation, respectively. Percolation in lattices has been intensively studied by mathematicians and physicist since the late Fifties, but its extension to complex networks is subject of very recent studies. These have provided a deep understanding and applications in a wide range of fields, including, for example, epidemics (Lloyd, 2001; Newman, 2002a) and communication. In the study of the spread of a disease, people that get in contact with other people form a social network, and the probability of transmitting the disease between neighbors leads to a percolation process. In a communication network (say, for example, Internet), nodes or links may fail or be directly attacked with some probability, blocking the spread of information through them and possibly breaking the giant component into smaller, disconnected components. In these situations, there is a critical vertex or edge probability above which there still exists a giant (although possibly smaller) component, but below which this component breaks into pieces. This critical probability marks the difference between an epidemic outbreak or a confined disease, or between a functional network or a disconnected one.

All this crucial features—the existence of the giant connected component, its size and the critical probability at which it first appears, and also the distribution of component sizes—can be calculated by building upon generating functions (Newman et al., 2001). Let  $R_s$  be the probability that by following a random edge to one of its ends we arrive at a connected component of size  $s$ . Edges are occupied with probability  $\phi$  and empty with probability  $1 - \phi$ . Then with probability

$R_0 = 1 - \phi$  we reach a component of size zero (the edge cannot be traversed because it is empty). With probability  $\phi$  we arrive at a vertex, so the size of the component is at least one. This reached vertex has excess degree  $k$  with probability  $r_k$ , and thus in total the probability that we reach a component of size  $s$  (different from zero) depends on the probability that the size of the components that emerge of this vertex is in total  $s - 1$ :

$$R_s = \phi \sum_{k=0}^{\infty} r_k \sum_{s_1, s_2, \dots, s_k} R_{s_1} R_{s_2} \cdots R_{s_k} \delta_{1+s_1+s_2+\dots+s_k, s}. \quad (2.46)$$

We denote the generating function of this probability  $h_R(z)$ . By definition, it is a power series of  $z$  with coefficients  $R_s$ , so

$$h_R(z) = R_0 + R_1 z + R_2 z^2 + \cdots = 1 - \phi + \sum_{s=1}^{\infty} R_s z^s. \quad (2.47)$$

If we expand it using Eq. 2.46 and convolution of probabilities, we get an implicit equation:

$$h_R(z) = 1 - \phi + \phi \sum_{k=0}^{\infty} z r_k [h_R(z)]^k = 1 - \phi + \phi z g_r[h_R(z)]. \quad (2.48)$$

Here we used the generating function of the excess degree defined in Eq. 2.38.

Similarly, we can compute the probability  $P_s$  that a randomly chosen *vertex* belongs to a connected component of size  $s$ . Starting from such a vertex, with probability  $p_k$  there are  $k$  edges that emerge from it, leading to  $k$  new components of size  $R_{s_i}$  each. Thus

$$P_s = \sum_{k=0}^{\infty} p_k \sum_{s_1, s_2, \dots, s_k} R_{s_1} R_{s_2} \cdots R_{s_k} \delta_{1+s_1+s_2+\dots+s_k, s}. \quad (2.49)$$

Following the previous considerations we find that the function generating this probability is

$$h_P(z) = z g_p[h_R(z)]. \quad (2.50)$$

These two generating functions encode all the information of the distribution of component sizes.

### 2.3.4 Mean component size, percolation threshold and the giant component

Knowledge of the generating functions  $h_P(z)$  and  $h_R(z)$  allows for the derivation of important properties of the network. The main one of them is perhaps the percolation threshold, which is the critical point where a giant connected component emerges, expanding a finite fraction of the order of the network size. Such

critical point marks a phase transition in the network structure. Below it, only finite components exist. Another important property which also allows us to find the percolation threshold is the mean size of the finite components in the network—the *mean component size*. This average diverges when an infinite component emerges, acting as a susceptibility in a magnetic material and pointing out the critical transition.

Recall  $h_R(z)$  from Eq. 2.48, which generates the probability that following a random edge a component of a *finite* size is reached. Then, the probability, defined in Eq. 2.32, that a component of *any* finite size is reached, can be expressed as  $u \equiv h_R(1)$ . This gives an equation for  $\phi$  with two different regimes: below the critical probability,  $\phi < \phi^*$ , there is a unique real, positive solution  $u^* = 1$ , while above it,  $\phi > \phi^*$ , a new solution  $u^* < 1$  appears (as in the previously discussed Figure 2.16). This transition marks the critical percolation threshold  $\phi^*$ . Unfortunately,  $u = h_R(1)$  is usually a transcendental equation and has to be solved numerically.

Similarly, the probability that a randomly chosen vertex belongs to any finite component is  $h_P(1) = g_p(u)$ . This gives the general expression for the fraction of vertices belonging to the giant connected component:

$$S = 1 - h_P(1) = 1 - g_p(u). \quad (2.51)$$

Note that below the percolation threshold  $u^* = 1$  is the only solution and therefore there is no giant connected component,  $S = 1 - g_p(1) = 0$ . On the other hand, above  $\phi^*$  the solution  $u^* < 1$  brings a non zero size  $S > 0$ . Again, this equation is usually transcendental and has to be solved numerically.

Finally, the mean component size is the first moment of the probability distribution  $P_s$ , so it is simply given by

$$\langle s \rangle = \frac{h'_P(1)}{h_P(1)} = \frac{h'_P(1)}{1 - S}. \quad (2.52)$$

Note that we divided by  $h_P(1)$  because above the threshold this generating function is not normalized. As we said the mean component size diverges at the critical point, where an infinite component emerges. This is another fingerprint of the phase transition, and since

$$h'_P(1) = 1 + g'_p(1)h'_R(1) \quad \text{and} \quad h'_R(1) = \frac{\phi}{1 - \phi g'_r(1)} \quad (2.53)$$

the mean component size diverges when  $1 = \phi g'_r(1)$ . The percolation threshold is thus given by the well-known result

$$\phi^* = \frac{1}{g'_r(1)}. \quad (2.54)$$

Note that it depends only on  $g'_r(1)$  (which in turn depends on  $\langle k \rangle$  and  $\langle k^2 \rangle$ , see Eq. 2.34), so in general there is no need to find a closed expression for  $h_P(z)$  and  $h_R(z)$  to obtain the most important percolation properties of the network.

## 2.4 Simulation of networks

To test the analytical results, or to provide insight on a process when an exact solution is difficult to find, in this thesis we have used numerical simulations of complex networks. All the network numerical results of Chapter 3 have been simulated by an implementation of the network generation, transformation and analysis in C++. We have written a program that can generate a large set of graph models, including the Erdős-Rényi, the configuration and the Watts-Strogatz model, as well as use data from real-world networks. The program also performs simulations of entanglement percolation and can compute several network properties, such as the size of components and of the biggest component, of limited components, and distributions of path lengths. The program have been written using the flexible Boost Graph Library<sup>6</sup> (part of the C++ Boost Library<sup>7</sup>), that includes many graph data structures and algorithms that are easily extendible (Siek et al., 2002). We have also used the Mersenne Twister MT19937 algorithm, which generates high quality uniform pseudorandom numbers (Matsumoto and Nishimura, 1998), as implemented in the Boost Random Number Library<sup>8</sup>. We do not include the source code of the program, because it is about 9 000 lines of code long. Instead, in this Section we introduce some of the basic concepts of graph data structures and algorithms and explain how we have implemented them to obtain the simulation results. The full code is however temporarily accessible at <http://sindominio.net/~nilvar/code/qper/>, and available upon request.

### 2.4.1 Graph data structures

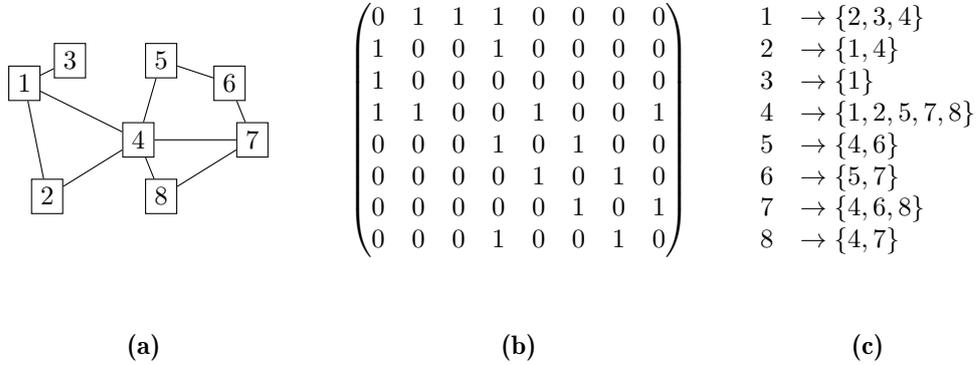
A graph can be implemented in many different ways, but any graph data structure contains the set of vertices and edges, and usually provides some basic graph operations such as adding and removing edges, testing whether two vertices are neighbors or not and listing the number of neighbors of a given vertex. Sometimes vertices and edges can have some attributes, such as the weight or the color. In this case, the data structure can also provide a way to read and change these properties, or store them externally. Here we introduce the two most common implementations, namely the adjacency matrix and the adjacency list.

---

<sup>6</sup>The latest version is accessible at <http://www.boost.org/doc/libs/release/libs/graph/>.

<sup>7</sup><http://www.boost.org/>.

<sup>8</sup><http://www.boost.org/doc/libs/release/libs/random/>.



**Figure 2.17.** Example of an undirected graph and its adjacency matrix (b) and adjacency list (c).

### Adjacency matrix

A graph  $G = \{V, E\}$ , with  $N = |V|$  the number of vertices, can be mapped to a  $N \times N$  square matrix  $A$ , defined as

$$A_{uv} \begin{cases} 1 & \text{if } (u, v) \in E, \\ 0 & \text{otherwise.} \end{cases} \quad (2.55a)$$

$$(2.55b)$$

$A$  is the **adjacency matrix** of the graph  $G$ , rows and columns correspond to vertices and each entry encodes the information whether the two corresponding vertices are connected by an edge or not (see Figure 2.17b). If the graph is undirected, the adjacency matrix is symmetric,  $A_{uv} = A_{vu}$ . If the graph has no self-loops, then the diagonal entries of  $A$  are zero.

The adjacency matrix can be extended to include multiple edges by setting  $A_{uv}$  as an integer that indicates the number of edges between  $u$  and  $v$ . Similarly, allowing  $A_{uv}$  to be a real number, it can encode the weights of edges.

### Adjacency list

The **adjacency list** (Figure 2.17c) represents the graph as a list of vertices, each of them storing a list of edges containing it (ie a list of its neighbors). If the graph is directed, then each vertex stores only its outgoing edges.

### Cost of operations

One usually decides among these two data structures depending on two factors. The first one is the required memory storage. The adjacency matrix uses an  $N \times N$  matrix, and thus requires  $O(N^2)$ . On the contrary, the adjacency list uses only  $O(N + M)$  memory, where  $M = |E|$  is the number of edges. If the graph is sparse,

**Table 2.1.** Required memory of storage and time complexity of some common operations in the adjacency matrix and adjacency list.  $N$  is the number of vertices,  $|V|$ ;  $M$  the number of edges,  $|E|$ ; and  $\alpha$  the maximum number of neighbors for any vertex in the graph.

	Adjacency matrix	Adjacency list
Storage	$O(N^2)$	$O(N + M)$
Add vertex	$O(N^2)$	$O(1)$
Add edge	$O(1)$	$O(1)$
Remove vertex	$O(N^2)$	$O(\alpha)$
Remove edge	$O(1)$	$O(\alpha)$
Access edge	$O(1)$	$O(\alpha)$

the adjacency list is much less demanding in terms of memory space. The second main factor is the time complexity of the operations that one needs to perform on the graph. Adding and removing vertices in the adjacency matrix is very slow, as one has to resize or copy the matrix, but at the same time adding, removing and checking the existence of an edge is done in constant time. The adjacency list, though, is more efficient if one needs to access the neighbors of a given vertex. Table 2.1 summarizes this cost. All the graphs that we have considered in this thesis are sparse. Hence, we have used the adjacency list in our simulations, which also provides the advantage to access very fast the neighbors of a vertex, and thus simplifies the implementation of the  $q$ -swap.

## 2.4.2 Random numbers

Most random number generators provide pseudo-random numbers that are uniformly distributed on a given integer range. These numbers are ready to be used to select, for example, a random vertex or edge of a network, or can be easily mapped to a uniform distribution on the range  $[0, 1)$ , so for example by drawing a random  $r \in [0, 1)$  one can decide that an edge is occupied in a percolation process if  $r < \phi$ . But sometimes, like in the configuration model, one needs random numbers that follow a specific degree distribution, which can be arbitrary. The Boost Random Number Generator Library already implements mappings from uniform generators to some distributions (eg the Poisson distribution), but here we need a method to obtain general distributions (and in particular, we are interested in reproducing the power-law with exponential cutoff of Eq. 2.19). The two main ways to achieve that are the transformation method and the rejection method, which can also be combined in an hybrid method with features of both approaches (Newman and Barkema, 1999, p. 396).

Suppose we want to obtain a random number  $x$  distributed according to some normalized function  $f(x)$  which is zero outside an interval  $[x_{\min}, x_{\max})$ , and we have a generator that provides uniform random numbers  $r \in [0, 1)$ . The probability that

a random  $X$  falls between  $x_{\min}$  and  $x$  is the cumulative distribution function

$$F(x) = \int_{x_{\min}}^x f(x') dx'. \quad (2.56)$$

This probability is equal to that of choosing  $r$  uniformly at random, so we can identify

$$r = F(x). \quad (2.57)$$

Hence, to obtain  $x$  the function  $F$  must be inverted. This is the so-called **transformation method**, which is very convenient because it produces a random  $x$  from every  $r$ .

Sometimes, however, it is not possible to obtain  $F^{-1}$ . Then, one can use the **rejection method** to generate a number with probability proportional to  $f(x)$ : one generates two uniform random numbers,  $x \in [x_{\min}, x_{\max})$  and  $r \in [0, 1)$ , and keep the former if

$$r < \frac{f(x)}{f_{\max}}. \quad (2.58)$$

Here  $f_{\max}$  is the maximum value of  $f(x)$  in the interval  $[x_{\min}, x_{\max})$ . If this condition is not fulfilled, then a new pair has to be generated, until an acceptable  $x$  occurs. This method is much more direct to implement, and allows to obtain random numbers according to any distribution, but has to main drawbacks:  $x_{\min}$  and  $x_{\max}$  must be finite, and it is much less efficient than the transformation method, as commonly one has to produce many random numbers to obtain a useful one.

These two methods can be combined in the **hybrid method**. As before, one has to generate two random numbers, but in this case  $x$  is generated by the transformation method according to a distribution  $g(x)$  that satisfies  $g(x) \geq f(x)$  for all  $x \in [x_{\min}, x_{\max})$  and for which  $G^{-1}$  can be obtained. Then,  $x$  is accepted as in the rejection method if

$$r < \frac{f(x)}{g(x)}. \quad (2.59)$$

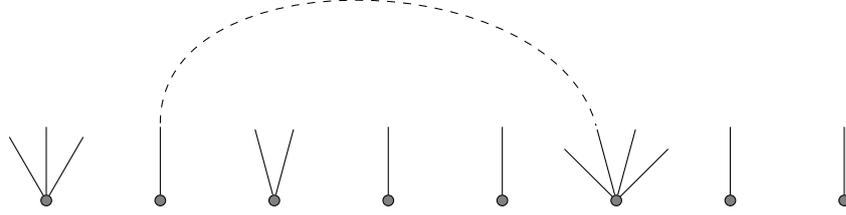
This is more efficient than the rejection method, as  $g(x)$  adapts better to  $f(x)$  than  $f_{\max}$ , and does not require  $x_{\min}$  and  $x_{\max}$  to be finite. Note that in the hybrid method the function  $g(x)$  has to be normalized (and hence the function  $f(x)$  is not).

### 2.4.3 Generation of graphs

#### Erdős-Rényi model

The Boost Graph Library (BGL) comes with a generator of the Erdős-Rényi model, the `sorted_erdos_renyi_iterator`<sup>9</sup>, so we directly used it to simulate Erdős-

<sup>9</sup>Note that in old versions of the BGL, there is a bug in this generator (see Changeset 56651, <https://svn.boost.org/trac/boost/changeset/56651>). This has been fixed since version 1.41.0.



**Figure 2.18.** Implementation of the configuration model. Random  $k_u$  stubs are assigned to node  $u$ , for every node. Then, a pair of stubs is selected uniformly at random among all possible stubs and joined to form an edge.

Rényi networks. (To be more accurate, it generates an instance of the Gilbert model.) The BGL also has a generator for a graph with a power-law degree distribution, based on [Palmer and Steffan \(2000\)](#). However, it only generates graphs with a specific degree distribution  $\beta k^\alpha$ .

### Configuration model

General graphs within the configuration model, ie with uncorrelated degree distribution, are relatively easy to generate ([Newman et al., 2001](#)). First, a set of  $N$  numbers  $\{k_u\}$  randomly chosen to follow the desired degree distribution is generated, so each vertex  $u$  has  $k_u$  stubs or “half edges” associated with it. In [Chapter 3](#) we will simulate networks of a degree distribution that follows a power law with an exponential cutoff, ie those of [Eq. 2.19](#). Let us rewrite the distribution here for convenience:

$$p_k = Ck^{-\tau}e^{-k/\kappa}. \quad (2.60)$$

To generate a random number  $k$  that follows this distribution, we use the hybrid method explained in [Section 2.4.2](#). First, a random  $k$  proportional to the exponential distribution  $e^{-k/\kappa}$  is produced by the transformation method using a uniform random number  $r_1 \in [0, 1)$ :

$$k = \lceil -\kappa \log(1 - r_1) \rceil. \quad (2.61)$$

Note that we compute  $\log(1 - r_1)$  instead of  $\log r_1$ , to make sure that the logarithm is always finite and does not give an error. Then,  $k$  is accepted with probability  $k^\tau$  by producing a new  $r_2 \in [0, 1)$  and checking if

$$r_2 < k^\tau. \quad (2.62)$$

Otherwise, a new pair  $r_1, r_2$  is produced and the process repeated as many times as necessary.

If the sum of the generated random numbers  $\sum_u k_u$  is odd, a new set is generated until an even sum is obtained, so all stubs can be joined. Then, pairs of

stubs are selected randomly and joined to form edges until there are no stubs left, as depicted in Figure 2.18. Note that these pairs are selected uniformly at random among all the stubs, not among all the vertices. In our simulations we do not allow self-loops or multiple edges, but the code has the option to allow them.

### Watts-Strogatz model

The BGL comes with a generator for the Watts-Strogatz model with rewired shortcuts, called `small_world_iterator`. However, we were interested in the model with added shortcuts, for which the analytical treatment is simpler (Moore and Newman, 2000). Hence, we implemented our own generator, which is very simple: it first creates the base lattice (see Figure 2.11) and then for every edge in the original lattice adds an edge among two nodes chosen uniformly at random if  $r < \beta$ , for a uniform random number  $r \in [0, 1)$  and a shortcut probability  $\beta$ .

### Regular graphs

We also checked previously known entanglement percolation results in regular lattices (Acín et al., 2007), so we could (numerically) compare the full value of  $S$  for any  $\phi$ . To do so, we implemented a generator for the honeycomb and triangular lattices, and also for the Cayley tree—which, however, has very strong border effects.

### Real-world datasets

As a check of the models with real-world network data, we considered two network datasets. The first one, of the World Wide Web, was obtained by Albert et al. (1999) and consists of websites in the domain `nd.edu`. It is accessible at <http://www.nd.edu/~networks/resources.htm>, together with many other interesting datasets. The second one, of the Web of Trust in the OpenPGP cryptographic protocol, was obtained directly by us from the Swiss keyserver, as explained in Section 2.2.4. It is accessible at <http://sindominio.net/~nilvar/netdata/>, with links to other network data repositories.

## 2.4.4 Algorithms on graphs

### Breadth First Search

The Breadth First Search (BFS) is a standard search algorithm on a graph, upon which many other algorithms are based. It is already implemented in the BGL in a flexible form that allows easy customization. Here we briefly review it, as it is the main algorithm that we later use to implement  $q$ -swaps operations on a network. The BFS begins at a given node  $u$  (sometimes called root) and explores all its neighbors  $v \in \mathcal{N}_u$ . It continues by exploring all the unexplored neighbors

of each first-neighbor of  $u$ , ie all  $w \in \mathcal{N}_v$  for  $v \in \mathcal{N}_u$  and so on, until it has explored the whole component and there are no more unvisited nodes. If the graph is connected, the BFS visits all its vertices; if not, it explores only a component, and then one has to start a new search from an unvisited vertex if the full graph has to be explored. In fact, one of the uses of the BFS is to check if a given graph is connected. The main alternative to BFS is Depth First Search (DFS), which instead of examining a component by “layers” explores first all the vertices that can be reached from a first neighbor before moving to the next one.

The actual implementation of the Breadth First Search is done by adding all newly found vertices to a “First-In, First-Out” (FIFO) queue, so the first vertex added will be the first to be examined for new neighbors.

### Incremental connected components algorithm

In simulations of entanglement percolation, we need to compute the size of the components in the network for some edge probability  $\phi$  (and, in particular, the size of the *biggest* component). The BFS algorithm can be used for that, exploring all components for every  $\phi$ . The running time of the BFS is  $O(N + M)$ , because all  $N$  are explored and every edge of each node is checked for unvisited vertices. Since we are interested in the percolation properties of the network for many different  $\phi \in [0, 1]$ , use of the BFS would mean to explore the full graph every time. Instead, one can use a different approach called **incremental connected components**, which makes use of disjoint-sets data structure that maintains and updates efficiently a collection of disjoint sets that corresponds to connected components in the graph (Siek et al., 2002). To study the connected components of a graph  $G$  with edge probability  $\phi$ , for many different values of  $\phi \in [0, 1]$ , we start with an empty graph  $G'$ , with  $V(G') = V(G)$ , and then for every *incremental*  $\phi$  from 0 to 1, each edge  $e \in E(G)$  is added to  $G'$  if a uniform random number  $r \in [0, 1)$  fulfills  $r < \phi$ . Every time an edge is added to  $G'$ , the disjoint-sets is updated, and after all edges corresponding to a given  $\phi$  are checked, the size of the disjoint-sets is calculated. In all, the full process takes  $O[N + M\alpha(M, N)]$  and  $\alpha$  is a function that grows very slow, being smaller than 5 for any practical purpose (Siek et al., 2002).

### Limited-path-length components

In the case of limited-path-length components, that we will introduce in Chapter 3, the previous incremental connected components algorithm cannot be use. Here we want to compute which is the mean size of a component whose vertices are separated by at most a path of length  $l$  from a central vertex, and hence use a modified version of the BFS algorithm that stops when it arrives at vertices that are at a distance  $l$  from the starting one, and returns the number of explored vertices.

### Path length distribution

Finally, to find the shortest path length between all pairs of vertices in a graph, so we can obtain the distribution of path length and the average path length  $l_{av}$ , we use Johnson's algorithm, already in the Boost Graph Library (Johnson, 1977; Siek et al., 2002). This algorithm takes  $O(N^2 \log V + VE)$  and is especially useful in the case of sparse graphs.

### 2.4.5 Implementation of bipartite edges and the $q$ -swap

In the quantum network model that we introduce in Chapter 3, edges can hold several copies of a quantum bipartite state that acts as a quantum channel between the nodes it connects. The probability of using each of these edges for quantum teleportation, that we relate to the percolation probability in a graph, depends on the number of copies and the type of state. This information is implemented in the program using a *property map* of the graph, that maintains a list of the edges together with two variables that encode the number and type of states in the edge. When an entanglement swapping is performed (see Section 3.1.3), the type and number of states in the involved edges change. The full operation that transforms local parts of the network is the  $q$ -swap (see Section 3.2.3). This transformation is performed on a target vertex and its neighbors, and makes use of entanglement swapping. As we will see, the  $q$ -swap is done on vertices with a specific degree  $q$  and has the property that it cannot be performed on a given vertex if some of its neighbors have undergone a different  $q$ -swap. Therefore, the order in which vertices are selected as target of  $q$ -swaps is important.

In our program, vertices are selected as possible targets using a search algorithm. First, a random vertex is selected, a  $q$ -swap is performed on it (if it has the appropriate degree  $q$ ), and then new vertices are explored, as in the BFS algorithm. For each new vertex explored, a  $q$ -swap is performed again if none of its neighbors were previously used as targets, otherwise the following vertex is explored. Once the component is fully explored, the program jumps randomly to an unexplored vertex of a different component and starts again. Optionally, one can select a different search algorithm, as the DFS, or explore vertices in a random order. At the end, both the structure and the types and number of states in some edges are changed, leading to an heterogeneous percolation where edges can have different occupancy probability.



## CHAPTER 3

---

### Distribution of bipartite entanglement

---

In the previous Chapter, we presented several models of networks, which were developed to reproduce the behavior of complex systems and of the processes that take place in them. These complex systems can be of a highly diverse origin, but share the common essential quality by which the local, non-trivial relations between its elements define the functioning of the whole. As it was also pointed out, in the last one or two decades they have been subject of intense interdisciplinary research, spanning from social sciences to biology and technology. However, they still remain quite unexplored in the quantum regime. This strikes with the fact that quantum networks (Kimble, 2008) where nodes communicate between them through quantum channels are essential to quantum information processing and distributed applications, and large quantum communication networks have a good chance to develop in a structure similar to their classical counterpart.

One of the key tasks in these networks is the transmission of quantum information between two distant nodes of the network. As the previous Chapter suggests, this task depends not only on the quality of the connections between nodes and on the amount of resources, but also on the underlying structure of the network. Therefore, understanding how structural properties affect the functionality of the network will allow both the design of better network architectures and the modification of existing ones that make communication feasible at farther distances, among a greater number of nodes or in the presence of higher levels of noise.

In this Chapter, we study communication strategies in a quantum network, and in particular we address the basic question—*how can we connect two distant nodes in a network?* More precisely, we consider a network where neighbors share some bipartite entanglement, and our goal is to establish a bipartite entangled state between two nodes that may be separated by a long distance. Although the

direct transmission of entanglement has already been experimentally achieved over long distances of about a hundred kilometers over free space (Ursin et al., 2007; Fedrizzi et al., 2009; Ma et al., 2012), entanglement distribution at a longer range or over noisier channels, such as optical fibers (Cirac et al., 1997; Kimble, 2008; Ritter et al., 2012), will most probably need the use of intermediate repeaters.

The basic operation that propagates entanglement over several intermediate stations is entanglement swapping (Żukowski et al., 1993), an implementation of quantum teleportation (Bennett et al., 1993) that teleports an entangled state. However, if the entanglement between neighbors is not maximal, or more realistically if the channels or the operations involved in the propagation introduce some noise, the resulting distributed entanglement after the swapping decays exponentially with the number of swaps—that is, with the length of the path that joins the two nodes. A promising proposal to overcome this difficulty is the use of quantum repeaters in a one-dimensional chain of quantum channels (Briegel et al., 1998; Dür et al., 1999) that alternate the swapping of many entangled states generated between neighbors with their purification (Bennett et al., 1996b; Deutsch et al., 1996; Dür and Briegel, 2007). While swaps introduce noise and reduce the entanglement, the purification steps recover it so it can be propagated to farther distances. This approach has already several experimental proposals, and experimental implementations of some of its building blocks<sup>1</sup>. This technique is promising because it requires a number of resources that scales polynomially with the distance (Dür et al., 1999), but it exploits only the entanglement present in a single path connecting the two nodes: the “network” nodes are part of is just a one-dimensional line.

The two nodes that one needs to entangle, however, may be embedded in a more realistic, higher dimensional network. In this case, the presence of many different paths can be used to deal with the exponential decay of entanglement with distance. A probabilistic approach to the distribution of entanglement that makes use of this high connectivity was first proposed by Acín et al. (2007) and then extended for different networks and sources of noise. Opposite to the scheme of quantum repeaters, which uses many states in a single path, in this case one considers a full network with just one or several states shared between neighbors, and tries to *probabilistically* distill perfect singlets from imperfect sources. In this way, the presence of many different paths can compensate for the low probability of entangling two distant nodes, much as in a percolation effect. In a different approach, more related to that of quantum repeaters, Lapeyre Jr. et al. (2012) used the entanglement in secondary paths to purify that present in the shortest path connecting the two nodes.

Here we will focus on the probabilistic generation of long-distance entanglement between two arbitrary nodes of a quantum network, and use methods and techniques from percolation theory. In this case, the structure of the network will be of utter importance in determining its functionality (Newman, 2002b). We will

<sup>1</sup>See for example Sangouard et al. (2011) and references therein.

consider that the network is of a complex topology, and that nodes can perform only local operations on the qubits they possess. Moreover, we restrict the information that every node has of the full network: the decision to perform these operations depends only on a restricted knowledge of the local network structure (the distribution of its neighbors) and on general statistical properties of the network, such as the mean degree.

There are several reasons to study networks with this topology and information restrictions. First of all, present telecommunication networks, such as the Internet, have a complex structure defined by its scale-free degree distribution and clustering, and it is very plausible that future quantum communication networks will develop a similar structure, or even that present networks will be able to operate in the quantum regime. Knowing how this structure affects the distribution of quantum information, and in particular of entanglement, is then a key problem for future distributed quantum tasks that rely on this type of networks.

Second, complex networks provide many interesting effects that are not present in lattices with a regular structure. These effects include, for example, a high resilience to node and edge failures, the small-world effect and compact path-length distribution, a richer structure and the presence of communities. Understanding their implications could be useful in the design of new quantum networks.

Regarding the limited knowledge about the network structure, this restriction motivates the design of protocols that are efficient in the use of classical communication and that can operate in parallel, without need to coordinate with other regions of the network. This is of a special importance when the network that we consider is very large, which makes it impractical to explore the network in its whole before implementing the quantum protocol, or to keep a “central” record of what is going on every part of the network. It is also important when it comes to considering that the network, in fact, is not a static system but can evolve in time, while keeping its statistical description more or less stable.

Finally, and from a more theoretical perspective, although few qubit states have been successfully characterized, an understanding of multipartite entanglement of many-particle systems is still lacking. As we will show here, some quantities which are exceedingly hard to calculate for states defined over regular lattices can be carried with ease in complex networks, which are defined through statistical properties. We will come back to this again on Chapter 4.

The relation between percolation and entanglement in a large system is already present in the work by [Calsamiglia et al. \(2005\)](#), followed by [Hartmann et al. \(2007a\)](#); [Calsamiglia et al. \(2007\)](#). There, the entanglement between particles of a spin gas—a system of a large number of spins, initially in a product state, that move classically and interact upon collisions, creating entanglement—is described by the underlying weighted graph that arises from the interactions, and two spins can localize entanglement ([Verstraete et al., 2004](#); [Popp et al., 2005](#)) if and only if they are in the same component of the graph. After some time, a significant

fraction of the spins in the gas has interacted with itself, and a “giant entangled component” emerges.

Entanglement percolation, which makes use of such higher dimensional networks, was first proposed by [Acín et al. \(2007\)](#). Its goal is to establish a maximally entangled state (which via quantum teleportation amounts to a perfect single-use quantum channel) between two arbitrary nodes of a network, where nodes are connected by partially entangled states. This can be easily achieved between two neighboring nodes, since a partially entangled state can be converted into a maximally entangled state with a probability that depends on the initial amount of entanglement. One can do the same with all states shared between neighbors, creating components of nodes that are connected by a maximally entangled state. Using then entanglement swapping ([Żukowski et al., 1993](#)), a maximally entangled state can be propagated from two neighboring nodes to any two nodes that belong to the same component, in analogy to the spin gas example introduced before ([Calsamiglia et al., 2005](#)). If the initial entanglement between neighbors is small, so is the probability of obtaining maximally entangled states from it, and the quantum network consists of many small connected components. Two nodes can establish a maximally entangled state if they belong to the same component, which in this case happens with a probability that tends to zero for large networks. If, instead, this entanglement is above a critical value, corresponding to the percolation threshold of the network, a giant component exists of size  $S > 0$ , and the probability that two nodes can establish a maximally entangled state tends to that of belonging to the giant component,  $S^2$ .

[Acín et al. \(2007\)](#) showed that by a quantum preprocessing, the quantum honeycomb lattice—a regular, two-dimensional lattice—could be transformed into the triangular lattice, which has a lower percolation threshold, and hence the probability of establishing long-distance entanglement was non-zero for a lower amount of initial entanglement. This result was latter extended to other types of regular lattices ([Perseguers et al., 2008](#); [Lapeyre Jr. et al., 2009](#)), complex networks ([Cuquet and Calsamiglia, 2009, 2011](#); [Wu and Zhu, 2011](#)), the use of multipartite entanglement ([Perseguers et al., 2010a](#)) and the presence of noise ([Broadfoot et al., 2009, 2010b,a](#); [Cuquet and Calsamiglia, 2011](#)).

Percolation theory has also found interesting applications in other areas of quantum information. For example, in the creation of a two-dimensional cluster state—which is a universal resource for measurement-based quantum computation (MQC)—there can be qubit losses, leading to a site percolation. If the occupancy of the cluster is above the percolation threshold of the square lattice, the state is still useful for MQC, otherwise it is not ([Browne et al., 2008](#)). Also in the domain of MQC, [Kieling et al. \(2007b\)](#) used a method based on percolation to efficiently construct cluster states where the percolation probability comes from the use of non-deterministic entangling gates. In quantum walks, the missing edges of a dynamical percolation process have been used as a model of decoherence ([Kollár et al., 2012](#)).

This Chapter is divided in two main parts, each consisting of two Sections. The first two Sections are devoted to networks of bipartite pure states. In Section 3.1 we present the basic notation and concepts that will be used in the rest of the Chapter, specifically for pure states: entanglement transformations and entanglement swapping. However, we assume that the reader is familiar with quantum information theory and do not give an introduction to it; instead, we refer the reader to Nielsen and Chuang (2000) for an introduction to the subject and the common notation used. Then, in Section 3.2, we study probabilistic distribution of bipartite entanglement in networks where edges consist of pure, non-maximally entangled states. We see how the structure of the network affects the percolation of entanglement, and propose a family of local transformations that can decrease the critical entanglement and increase the probability of establishing a long-distance entangled pair. We use the methods presented in Chapter 2 to solve analytically the case of networks with uncorrelated degree distributions, and confront this solution with extensive numerical simulations. For the case of correlated networks, such as the Watts-Strogatz model, and a real-world network, we see that simulations give similar enhancement results. Finally, we relate the network transformation with recent research in the field of complex networks regarding what has been termed explosive percolation.

In the following two Sections, we turn to noisy networks of bipartite mixed states. In Section 3.3 we present entanglement transformations and swapping in the case of mixed states, and discuss how the previous percolation results can be adapted. In the case of rank-two mixed states arising from an amplitude damping channel, for example, the strategies developed in the previous Section can be directly applied. Then, in Section 3.4, we study how noise limits the maximum number of stations through which information can be repeated and solve the corresponding limited-path-length percolation problem.

## 3.1 Entanglement in pure states

### 3.1.1 Entangled states

In the first part of the Chapter we will restrict to the entanglement of bipartite pure states  $|\psi\rangle$  of two qubits. In their most general form, these states can be written as

$$|\psi\rangle = \sum_{i,j} c_{ij} |i\rangle_A \otimes |j\rangle_B, \quad (3.1)$$

where  $A$  and  $B$  are two subsystems, and  $\{|i\rangle_A\}$  and  $\{|j\rangle_B\}$  form an orthonormal basis of the corresponding Hilbert spaces,  $\mathcal{H}_A$  and  $\mathcal{H}_B$ . The state is normalized,  $\sum_{i,j} |c_{ij}|^2 = 1$ , and it is entangled if it cannot be written as  $|\psi\rangle = |\psi_1\rangle_A \otimes |\psi_2\rangle_B$ , where  $|\psi_{1,2}\rangle_{A,B}$  is a pure state in system  $A, B$ .

States like this will be distributed among two different nodes  $A$  and  $B$  of the

network, each holding one of the two qubits, and thus making them neighbors, in the sense that they share some entanglement that can be used as a single-use quantum channel. Each node will hold as many qubits as entangled states share with its neighbors. We will allow local operations and classical communication (LOCC), meaning that every node will be able to perform any quantum operation on its qubits and to communicate classical information to its neighbors. Hence, up to a local change of basis—which is allowed by our assumptions—, the state  $|\psi\rangle$  can be rewritten as

$$|\psi\rangle = \sqrt{\lambda_0} |00\rangle + \sqrt{\lambda_1} |11\rangle, \quad (3.2)$$

where  $|00\rangle$  is the compact form of  $|0\rangle_A \otimes |0\rangle_B$ . This expression is called the Schmidt decomposition, and the Schmidt coefficients  $\sqrt{\lambda_0}$  and  $\sqrt{\lambda_1}$  are non-negative real numbers satisfying  $\lambda_0 + \lambda_1 = 1$ . Without loss of generality, the basis can be chosen so that  $\lambda_0 \geq \lambda_1$ , and  $|\psi\rangle$  is entangled if  $\lambda_1 \neq 0$ . From Eq. 3.2 one can easily see that the square of the Schmidt coefficients,  $\lambda_0$  and  $\lambda_1$ , coincide with the eigenvalues of the reduced state of the system  $\rho_A = \text{Tr}_B |\psi\rangle \langle \psi|$  or  $\rho_B = \text{Tr}_A |\psi\rangle \langle \psi|$  (for pure states, both reductions have the same spectra).

### 3.1.2 Deterministic and probabilistic transformation of entanglement

Since any two states with the same Schmidt coefficients are locally equivalent, all the information about the entanglement of the two-qubit state  $|\psi\rangle$  is encoded in the single parameter  $\lambda_0$ : the degree of mixedness of the local reductions fixes the amount of entanglement. In fact, entanglement cannot increase under the action of LOCC (Bennett et al., 1996a; Popescu and Rohrlich, 1997; Vedral et al., 1997; Horodecki et al., 2000), and it is only possible to deterministically transform the state  $|\psi\rangle$ , with largest Schmidt coefficient  $\lambda_0$ , into a different state  $|\varphi\rangle$ , with largest Schmidt coefficient  $\mu_0$ , via LOCC if and only if  $\lambda_0 \leq \mu_0$ .

This statement is a particular result of the more general necessary and sufficient condition due to Nielsen (1999) for pure states  $|\alpha\rangle$  and  $|\beta\rangle$  shared between two  $d$ -dimensional subsystems  $A$  and  $B$ , which makes use of majorization theory. For a real vector  $\mathbf{a} = (a_0, \dots, a_{d-1})$  in  $d$  dimensions, we use  $\mathbf{a}^\downarrow$  to denote the reordering of  $\mathbf{a}$  such that  $a_0^\downarrow \geq a_1^\downarrow \geq \dots \geq a_{d-1}^\downarrow$ . Then, given two  $d$ -dimensional real vectors  $\mathbf{a} = (a_0, \dots, a_{d-1})$  and  $\mathbf{b} = (b_0, \dots, b_{d-1})$ ,  $\mathbf{a}$  is **majorized** by  $\mathbf{b}$  (or  $\mathbf{b}$  majorizes  $\mathbf{a}$ ), written as  $\mathbf{a} \prec \mathbf{b}$ , if

$$\sum_{j=0}^k a_j^\downarrow \leq \sum_{j=0}^k b_j^\downarrow \quad (3.3)$$

for all  $k = 0, \dots, d-1$  (and equality instead of inequality at  $k = d-1$ ). Equivalently, if  $\mathbf{a} \prec \mathbf{b}$  then the two vectors are related by a doubly stochastic matrix,  $\mathbf{a} = D\mathbf{b}$ . If the vector  $\mathbf{a}$  and  $\mathbf{b}$  are probability vectors, one can generate  $\mathbf{a}$  from  $\mathbf{b}$

by a mixing stochastic process. That is,  $\mathbf{a}$  is more mixed than  $\mathbf{b}$ . For example, for  $d = 2$ ,  $D$  takes the form

$$D = \begin{pmatrix} t & 1-t \\ 1-t & t \end{pmatrix}, \quad (3.4)$$

with  $0 \leq t \leq 1$ . Then one can obtain, say,  $(0.6, 0.4)$  from  $(0.7, 0.3)$  by a mixing process with  $t = 0.75$ . Hence,  $(0.6, 0.4) \prec (0.7, 0.3)$ , ie the probability distribution  $(0.6, 0.4)$  is more disordered than  $(0.7, 0.3)$ .

[Nielsen \(1999\)](#) took this result and premised that entanglement cannot increase under LOCC. Considering the reduced states of one of the subsystems,  $\rho_\alpha = \text{Tr}_B |\alpha\rangle\langle\alpha|$  and  $\rho_\beta = \text{Tr}_B |\beta\rangle\langle\beta|$ , and the vectors  $\boldsymbol{\alpha}$  and  $\boldsymbol{\beta}$  of its eigenvalues (which are the squares of the Schmidt coefficients), then the result of [Nielsen](#) is that  $|\alpha\rangle$  can be transformed *deterministically* to  $|\beta\rangle$  by local operations on  $A$  and  $B$  and classical communication between them if and only if  $\boldsymbol{\alpha} \prec \boldsymbol{\beta}$ .

The majorization relation introduces a partial order on the set of bipartite pure states, so for two generic bipartite states  $|\alpha\rangle$  and  $|\beta\rangle$  it might not be always possible to transform one into the other. This defines different classes of entanglement. In the case where  $|\alpha\rangle$  can be transformed into  $|\beta\rangle$ , it was shown by [Lo and Popescu \(2001\)](#) that the transformation can be done by a single measurement on the system  $A$  and a unitary operation on  $B$  depending on the outcome, so (only) one-way classical communication is needed.

However, for our case of two qubits (and  $d = 2$ ), the order induced by the majorization relation is total, and the state which can be deterministically transformed to any other is the maximally entangled

$$|\Phi_{00}\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle). \quad (3.5)$$

This state is one of the four elements of the Bell basis,

$$|\Phi_{ab}\rangle = \frac{1}{\sqrt{2}} (|0a\rangle + (-1)^b |1\bar{a}\rangle), \quad (3.6)$$

with  $a$  and  $b$  either 0 or 1, and  $\bar{a}$  denoting the bit-complement of  $a$ , ie  $\bar{a} = a \oplus 1$ .

The situation changes substantially if one allows for probabilistic transformations, ie stochastic local operations and classical communication (SLOCC). The results of [Nielsen](#) were extended to this case by [Vidal \(1999\)](#): given two pure states  $|\alpha\rangle$  and  $|\beta\rangle$ , defined as in the previous paragraphs, the maximal probability of obtaining  $|\beta\rangle$  from  $|\alpha\rangle$  is

$$\phi(|\alpha\rangle \rightarrow |\beta\rangle) = \min_{k \in [0, d-1]} \frac{\sum_{j=k}^{d-1} \alpha_j^\downarrow}{\sum_{j=k}^{d-1} \beta_j^\downarrow}. \quad (3.7)$$

The criterion by [Nielsen \(1999\)](#) of the necessary and sufficient conditions for the deterministic transformation is the  $\phi(|\alpha\rangle \rightarrow |\beta\rangle) = 1$  case. From this general result, it follows the case of interest here ([Lo and Popescu, 2001](#); [Bose et al.](#),

1999): a generic two qubit state can be transformed to a maximally entangled state (Eq. 3.5) by LOCC with optimal probability

$$\phi_1 = \min \{1, 2(1 - \lambda_0)\}. \quad (3.8)$$

This probability was first termed as **entanglement of single pair purification** (Bose et al., 1999). However, in the following we call  $\phi_1$  the **singlet conversion probability** of  $|\psi\rangle$ , or SCP for short (Acín et al., 2007), to denote its relation to the bond percolation probability. Apart from the already cited references, a nice review of majorization and its relation to bipartite states can be found in Nielsen and Chuang (2000, pp. 573–580) and Nielsen and Vidal (2001).

If instead of a single copy of  $|\psi\rangle$ , nodes  $A$  and  $B$  share  $n$  copies of this state,  $|\psi\rangle^{\otimes n}$ , the probability that they obtain *one* maximally entangled state  $|\Phi_{00}\rangle$  from them is

$$\phi_n \equiv \phi(|\psi\rangle^{\otimes n} \rightarrow |\Phi_{00}\rangle) = \min \{1, 2(1 - \lambda_0^n)\}. \quad (3.9)$$

### 3.1.3 Entanglement swapping

Two separated parties,  $A$  and  $B$ , that share a maximally entangled state, like the one in Eq. 3.5, can teleport the state of a qubit from  $A$  to  $B$  by performing some joint operations on the qubit and the local part of the entangled state of  $A$ , and sending two bits of classical information to  $B$  (Bennett et al., 1993). This **quantum teleportation** scheme can also be used to transmit the state of a subsystem of a larger system, that may be in an entangled states. This process is commonly known as **entanglement swapping** (Żukowski et al., 1993), and allows to entangle two systems that have not previously interacted.

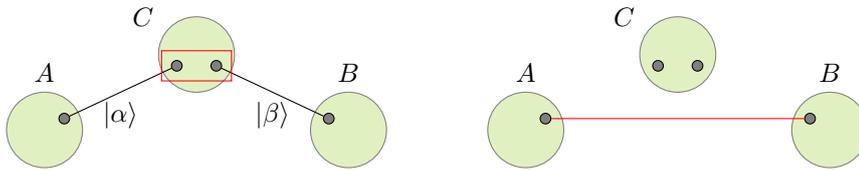
Here we present a variation of such scheme that involves non-maximally entangled state. The scheme is depicted in Figure 3.1. A central node  $C$  shares two entangled states  $|\alpha\rangle = \sqrt{\lambda_0}|00\rangle + \sqrt{\lambda_1}|11\rangle$  and  $|\beta\rangle = \sqrt{\mu_0}|00\rangle + \sqrt{\mu_1}|11\rangle$  with  $A$  and  $B$ , respectively. To swap the entanglement, qubits in  $C$  are projected onto the Bell state basis. The resulting state between the two qubits in  $A$  and  $B$  is

$$\frac{\sqrt{\lambda_0\mu_0}|00\rangle \pm \sqrt{\lambda_1\mu_1}|11\rangle}{\sqrt{\lambda_0\mu_0 + \lambda_1\mu_1}}, \text{ each with probability } \frac{\lambda_0\mu_0 + \lambda_1\mu_1}{2}, \quad (3.10a)$$

or

$$\frac{\sqrt{\lambda_0\mu_1}|01\rangle \pm \sqrt{\lambda_1\mu_0}|10\rangle}{\sqrt{\lambda_0\mu_1 + \lambda_1\mu_0}}, \text{ each with probability } \frac{\lambda_0\mu_1 + \lambda_1\mu_0}{2}. \quad (3.10b)$$

This results in an average SCP equal to  $2 \min\{\lambda_1, \mu_1\}$ . Hence, if the two initial states are the same,  $|\alpha\rangle = |\beta\rangle = \sqrt{\lambda_0}|00\rangle + \sqrt{\lambda_1}|11\rangle$ , the average SCP between  $A$  and  $B$  after one entanglement swapping is the same as the one before the swap between  $A$  and  $C$ , or  $C$  and  $B$ , which is a rather surprising result (Bose et al., 1999). This is however not true for further entanglement swappings, and in particular the



**Figure 3.1.** Entanglement swapping. Small dots represent qubits; large, green circles are nodes of a network. Initially, a central node  $C$  shares two entangled states  $|\alpha\rangle$  and  $|\beta\rangle$  with  $A$  and  $B$ , respectively (represented by black lines). Entanglement swapping consists in performing a Bell measurement on the two qubits of  $C$  (red rectangle). If the entanglement that is actually teleported is, for example, that between  $A$  and  $C$ , then  $C$  has to transmit the two classical bits of the measurement outcome to  $B$ , who will apply a unitary transformation on its qubits based on this information, and obtain one of the four states in Eq. 3.10 (red line).

use of a measurement basis of non-maximally entangled pairs of qubits (Perseguers et al., 2008) or the teleportation via non-maximally entangled states (Modlowska and Grudka, 2008) can perform better.

## 3.2 Entanglement percolation in pure-state networks

We first focus on networks of pure, non-maximally entangled states  $|\psi\rangle$  (see Eq. 3.2). Every edge connecting two nodes in the quantum network holds  $n$  of these states, and thus neighboring nodes have some probability  $\phi_n$  of getting a maximally entangled state from its non-maximal resources. This probability depends on the amount of entanglement per state and the number  $n$  of them, as shown in Eq. 3.9. If they succeed, these maximally entangled states can then be used for perfect teleportation, ie they are equivalent to a single-use ideal quantum channel.

### 3.2.1 Critical entanglement and long-distance entanglement probability

This strategy can be readily mapped into a bond percolation problem, which we call **direct entanglement percolation**. In previous works, it has also been called classical entanglement percolation (Acín et al., 2007), but since it uses quantum operations we will use the *direct* term to avoid confusions. The goal of entanglement percolation is to *probabilistically* establish a long-distance, maximally entangled state between two arbitrary nodes  $A$  and  $B$  of a large network. Using entanglement swapping, any two nodes are able to share maximal entanglement if both of them belong to the same component of maximally entangled nodes. Selecting a node  $A$  of the quantum network at random, it belongs to a component  $\mathcal{C}$  of size

$s$  with probability  $P_s$ . The probability that a second node  $B$  belongs to the same component is related to the relative size of the component,

$$P(B \in \mathcal{C}) = \frac{s-1}{N-1}. \quad (3.11)$$

Hence, the probability  $P_{AB}$  that two random nodes  $A$  and  $B$  can establish a maximally entangled state is

$$P_{AB} = \frac{\langle s \rangle - 1}{N - 1}, \quad (3.12)$$

where  $\langle s \rangle$  is the mean component size.

We are interested in a regime—if it exists—where  $P_{AB}$  is independent of  $N$ . As we introduced in Chapter 2, there exists a critical probability  $\phi^*$ , called the percolation threshold, that in this case of a quantum network can be related to a **critical entanglement**. Below the threshold, components are of size at most  $O(\log N)$ , and therefore  $P_{AB}$  tends to zero in the limit of large networks. Above the threshold, where  $\langle s \rangle$  diverges (the giant connected component appears),

$$P_\infty \equiv \lim_{N \rightarrow \infty} P_{AB} = \lim_{N \rightarrow \infty} \frac{\langle s \rangle - 1}{N - 1} = S^2 > 0, \quad (3.13)$$

ie two nodes can establish a maximally entangled state if they both belong to the giant component, which happens with probability strictly greater than 0. Hence, in this case the percolation threshold marks the critical amount of entanglement that is needed so that the probability of entangling two random nodes is nonzero. This threshold, or equivalently this critical entanglement, is independent of the distance that separates the two nodes, but depends strongly on the network topology. Hence, for a given type of edges  $\phi_n$ , long-distance entanglement will only be possible for networks fulfilling  $\phi_n > \phi^*$ .

Remarkably, due to the quantum nature of the connections, it is possible to drastically change the network topology by local actions: a particular measurement is done on qubits within the same node, destroying some of the connections but establishing new ones between second neighbors. Thus, a quantum preprocessing of the network can be carried before edges are converted into singlets, so the new structure provides a lower percolation threshold or a bigger giant component. This preprocessing can be done using entanglement swapping, and has been termed quantum entanglement percolation (Acín et al., 2007). Following the previous argument, we will call this strategy **swapped entanglement percolation**.

The quantum preprocessing allows to change, for example, a honeycomb lattice with two states  $|\psi\rangle$  per edge into a triangular one, where new edges have the same singlet conversion probability of a single state  $|\psi\rangle$  (Acín et al., 2007). Edges before and after the preprocessing have different singlet conversion probabilities, namely  $\phi_2$  and  $\phi_1 \leq \phi_2$ , respectively, so one has to be careful that, even if the new threshold

is lower, the decrease in the SCP can lead to a worse functioning of the quantum network. To compare the performance of these two percolation strategies (direct and swapped) in a specific network, we use the SCP of the state  $|\psi\rangle$ ,  $\phi_1$ . The percolation threshold of the honeycomb lattice and of the triangular lattice are

$$\phi_{\square}^* = 1 - 2 \sin(\pi/18) \quad (3.14a)$$

and

$$\phi_{\triangle}^* = 2 \sin(\pi/18), \quad (3.14b)$$

respectively (Sykes and Essam, 1964). In the original honeycomb lattice with  $|\psi\rangle^{\otimes 2}$  at every edge, the critical entanglement per edge is given by  $\phi_2 = \phi_{\square}^*$ . Equivalently, the critical entanglement per state is

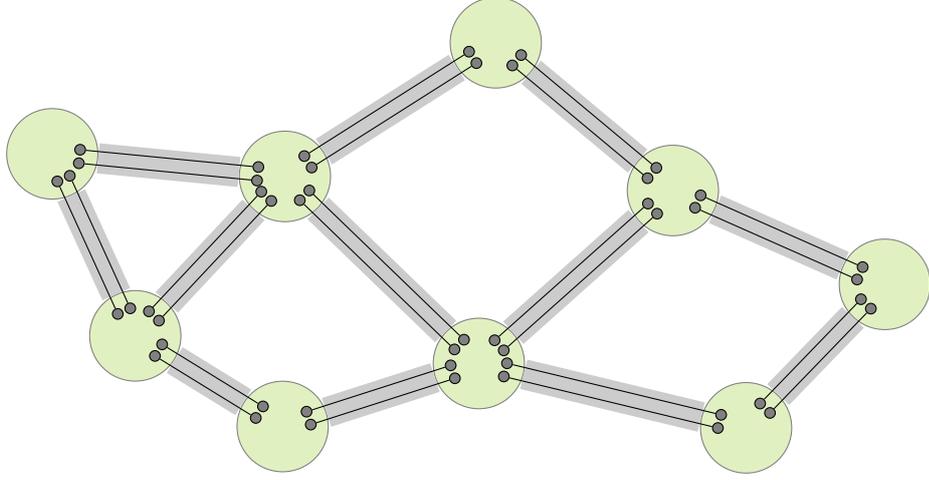
$$\phi_1 = 2 - \sqrt{2 \left(1 + 2 \sin \frac{\pi}{18}\right)} \approx 0.3585. \quad (3.15)$$

Above this value, a giant component of maximally entangled states exists, and  $P_{\infty} > 0$ . In the triangular lattice resulting from the quantum processing, the critical entanglement that had to be present in the original states  $|\psi\rangle$  is  $\phi_1^* = \phi_{\triangle}^* \approx 0.3472$ . Hence, the gain in the minimum amount of initial entanglement is of about 3% (Acín et al., 2007; Perseguers et al., 2008). A similar enhancement in the percolation threshold has also been observed in other regular structures (Perseguers et al., 2008; Lapeyre Jr. et al., 2009). As mentioned above, analytical results for  $P_{\infty}$  (or equivalently for the size of the giant component) are exceedingly hard to obtain and one has to rely on numerical results—see for instance Lapeyre Jr. et al. (2009), where in addition to numerical simulations some approximate results for  $\phi \rightarrow 1$  are given.

Here, we want to show that swapped entanglement percolation is not a peculiar property of some particular lattices, but it is a rather universal feature. For this purpose, we will broaden the study to a whole new class of networks: complex networks. We consider a complex network, of which we do not know the exact structure but only some general statistical properties, and decide whether to transform or not the local neighborhood of a given node based only on locally accessible information, such as the degree of this node. We will use generating function techniques, like the ones described in Section 2.3.2 of the previous Chapter, to calculate the percolation threshold and the size of the giant component for the modified network, and to obtain which is the best transformation of a given family.

### 3.2.2 Network model

We consider a quantum network in which neighboring nodes share  $n = 2$  copies of a bipartite pure entangled state of two qubits (see Eq. 3.2),  $|\psi\rangle^{\otimes 2}$ . A node of



**Figure 3.2.** Model of a quantum network of bipartite pure states. Large, green circles are nodes in the network. Each node of degree  $k$  contains  $2k$  qubits, represented as small dots. These qubits are entangled with qubits from neighboring nodes: each edge in the network (wide, gray lines) holds two entangled states  $|\psi\rangle$  (black lines).

degree  $k$  holds then  $2k$  qubits, as shown in Figure 3.2. Recall Eq. 3.8, which gives the singlet conversion probability for a single copy of  $|\psi\rangle$ :

$$\phi_1 = \min \{1, 2(1 - \lambda_0)\}.$$

Using Eq. 3.9, the singlet conversion probability of an edge  $|\psi\rangle^{\otimes 2}$  is

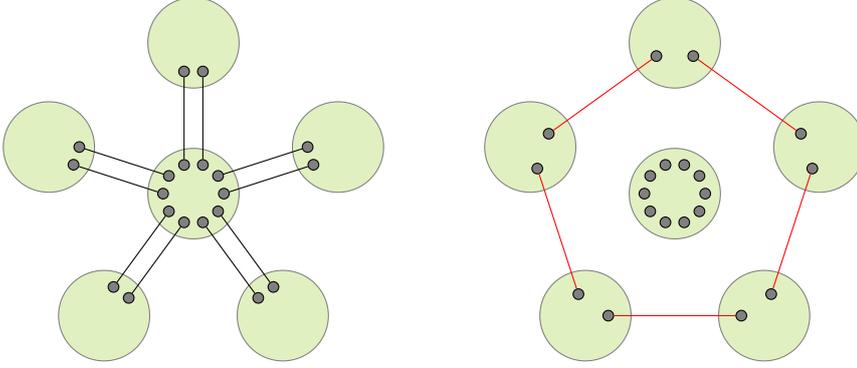
$$\phi_2 = \min \{1, 2(1 - \lambda_0^2)\} = \min \{1, 2\phi_1 - \phi_1^2/2\}. \quad (3.16)$$

With this probability two neighbors can establish a perfect channel between them. As we discussed above, for two distant nodes this probability depends on the structure of the network. In direct entanglement percolation, it will be strictly greater than zero if  $\phi_2 > \phi^*$ . The critical entanglement per state is thus

$$\phi_1^* = 2 - \sqrt{4 - 2\phi^*}. \quad (3.17)$$

### 3.2.3 Network transformation with local knowledge

We study the quantum preprocessing of the network—the swapped entanglement percolation—using a family of transformations that are applied on specific nodes. These transformations, that we call  $q$ -swap, require only local information of the network: the degree of a target node and the status of its neighbors. The  $q$ -swap (see Figure 3.3) is built upon basic transformations, namely entanglement swapping, that we introduced previously in Section 3.1.3. It performs entanglement

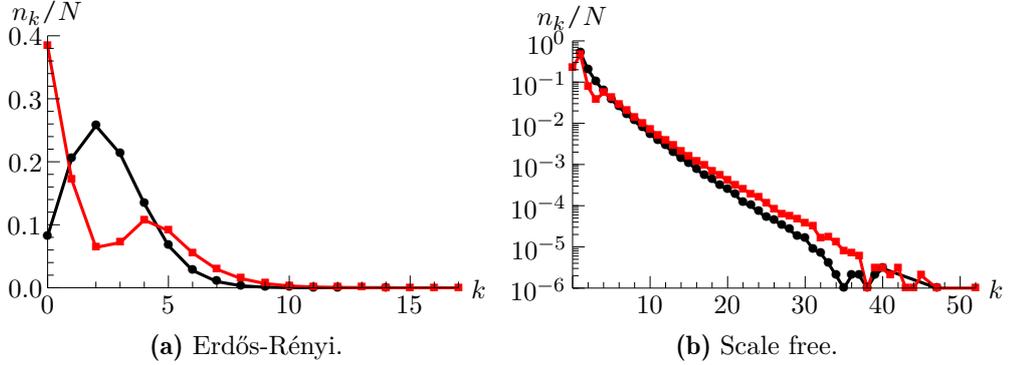


**Figure 3.3.**  $q$ -swap transformation, in this example on a target node of degree  $q = 5$ . Initially, the local network formed by the central node and its first-neighbors is a star of 5 leaves. After performing entanglement swapping on the qubits of the central node, the local network is transformed to a cycle of the 5 original leaves, and the central nodes is disconnected from the rest of the network.

swappings between successive pairs of neighbors of a central target node of degree  $q$ , thus changing an initial local  $q$ -star, with edges  $|\psi\rangle^{\otimes 2}$ , to a local  $q$ -cycle, with newborn edges that are in one of the outcomes of Eq. 3.10, while the central target node becomes disconnected from the network. As we said, the average singlet conversion probability of these newborn edges is  $\phi_1$  (lower than the one in the original edges,  $\phi_2$ ). One of the heuristic motivations for a transformation of this type is that, typically, a higher level of clustering leads to a lower percolation threshold (Newman, 2003a). However, this intuition cannot be elevated to a general statement since there are examples where a  $q$ -swap increases the percolation threshold (see eg Figures 3.11b, 3.12b and 3.12d).

For  $q \geq 4$ , there is a certain freedom in the actual implementation of the  $q$ -swap, as one can choose which nodes of the initial  $q$ -star will end up as neighbors in the new  $q$ -cycle. Had one access to other information beyond which are the neighbors of a specific node, a strategy based on more knowledge of the network would be possible. However, as we justified before, we restrict only to local knowledge of the network structure, and for simplicity we consider that the new neighbors are chosen at random.

For a given network topology (ie for some given global statistical properties), we will see that performing  $q$ -swaps on nodes with certain degrees improves the threshold. It is worth noting, however, that in some instances the application of particular  $q$ -swaps may be counterproductive, as mentioned above. In addition to the previously mentioned idea of increasing the clustering of the network, intuition of what type of transformations will decrease the threshold can be obtained by inspection of Eq. 2.54, that says that, for uncorrelated graphs, the percolation threshold is inversely proportional to  $g'_r(1)$ . Expanding it with Eq. 2.38, we can



**Figure 3.4.** Degree sequence  $n_k/N$  before (black) and after (red) 2, 3-swap in two networks of size  $N = 10^6$ . (a) Erdős-Rényi,  $\langle k \rangle = 2.5$ . (b) Power law with exponential cutoff,  $\tau = 1$  and  $\kappa = 4$ . In both cases, the increase in the number of isolated nodes is equal to the  $q$ -swaps performed, that turn a target node of degree  $q$  into one of degree 0.

see that

$$g'_r(1) = \sum_{k=1}^{\infty} k r_k = \sum_{k=2}^{\infty} \frac{k(k-1)}{\langle k \rangle} p_k = \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}. \quad (3.18)$$

Every  $q$ -swap transforms the target node into an isolated node with degree  $k = 0$ , but in turn the  $q$  neighbors of the target increase their degree by 1, so the mean degree  $\langle k \rangle$  of the network does not change. Hence, for a transformation that keeps  $\langle k \rangle$  fixed (like the  $q$ -swap does), one should go for an increase of  $\langle k^2 \rangle$ . As an example, Figure 3.4 shows the change in the degree sequence  $n_k$  of the 2, 3-swap in an Erdős-Rényi and a scale free network, where the number of nodes of degree 2 and 3 is deeply decreased and at the same time the number of nodes with high degrees is increased (note that even the maximum degree of the network is bigger). See also that the number of isolated nodes increases by a quantity equal to the number of  $q$ -swaps performed. Note, however, that this argument is partially flawed and is only given here to provide some intuition on the effect of  $q$ -swaps. Indeed, Eq. 3.18 is only valid for uncorrelated networks, while the transformed network after  $q$ -swap operations will exhibit degree-degree correlations. However, in the following sections we will see that even with these correlations the local action of the  $q$ -swaps leaves the transformed network with a structure amenable to analytical study.

### 3.2.4 Generating functions of the modified network

The main two figures of merit that we will use to compare between the direct strategy, that leaves the network structure unmodified, and the swapped strategy, that changes it, are the critical entanglement ( $\phi_1^*$  and  $\tilde{\phi}_1^*$  for the directed and swapped strategies, respectively) and the probability of connecting two random

nodes, related to the size of the giant components ( $S$  and  $\tilde{S}$ ). As we saw in Eq. 3.13, in direct entanglement percolation this probability is  $P_\infty = S^2$ . Since  $q$ -swaps disconnect vertices, which can always be chosen not to be the two corresponding to the parties that want to establish the maximally entangled pair, the probability of connecting two remote nodes in the swapped entanglement percolation is in fact

$$\tilde{P}_\infty = \hat{S}^2, \quad (3.19)$$

where  $\hat{S}$  is a normalized  $\hat{S} = \tilde{S}S_1/\tilde{S}_1$  and  $S_1$  ( $\tilde{S}_1$ ) is the value of  $S$  ( $\tilde{S}$ ) at  $\phi_1 = 1$ .

To compute these two values we will use the generating function formalism described in Section 2.3.2, which already gives us the solution for the original network,  $\phi_1^*$  and  $S$ . Recall that the method was to seek the generating function for the probabilities that, either following a random edge or starting from a random vertex, we get to a finite component of size  $s$ . These two probabilities are respectively  $R_s$  and  $P_s$ , and are generated by  $h_R(z)$  and  $h_P(z)$ . To find  $\tilde{\phi}_1^*$  and  $\tilde{S}$  of the modified network, we need to obtain expressions for the new generating functions  $\tilde{h}_R(z)$  and  $\tilde{h}_P(z)$ . Every particular  $q$ -swap can be implemented (or not) with probability  $\Pi_q$  (or  $1 - \Pi_q$ ) on nodes of degree  $q$ . Giving the values of the set  $\{\Pi_q\}$  for all  $q$  present in the network fully specifies our swapped strategy.

$q$ -swaps introduce cycles, so components are no longer treelike and generating functions cannot be directly used. Note however that, since newborn edges cannot be reused, those cycles do not overlap between each other, and can thus be treated as blocks of a treelike component by considering two steps in the branching process.

### Arriving to a $q$ -cycle: critical entanglement

We first compute the generating function for the probability  $\tilde{R}_s$  after  $q$ -swaps are done,  $\tilde{h}_R(z)$ . Let us first start with the case when we *would* arrive at a vertex of degree  $q$  that has undergone a  $q$ -swap transformation. Now, instead of arriving at a vertex of degree  $q$  connecting to other  $q - 1$  components, after a  $q$ -swap operation has been done we arrive at a cycle of  $q$  nodes (including the one we are coming from) connected via edges occupied with probability  $\phi_1$  (see Figure 3.5). When edges are converted into singlets, the accessible nodes of this new  $q$ -cycle form a path of length  $l$  with probability

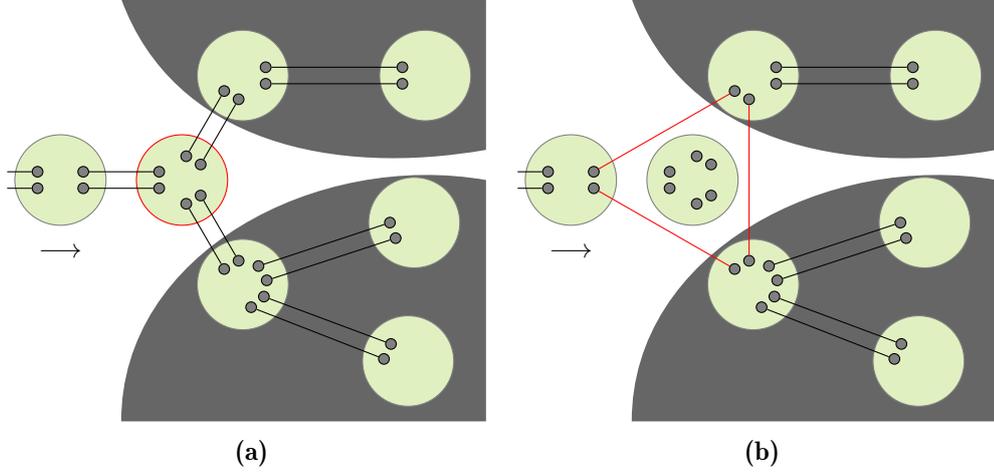
$$\begin{cases} \phi_1^q & \text{for } l = q, \\ q\phi_1^{q-1}(1 - \phi_1) & \text{for } l = q - 1, \\ (l + 1)\phi_1^l(1 - \phi_1)^2 & \text{for } l \leq q - 2. \end{cases} \quad (3.20a)$$

$$(3.20b)$$

$$(3.20c)$$

This probability is illustrated in Figure 3.6b.

Note that paths of length  $l < q$  will connect the “arriving node” (identified with a right arrow in Figure 3.5) to  $l$  unvisited nodes, while for  $l = q$ , the path will only connect  $q - 1$  unvisited nodes (the additional edge is used to close the  $q$ -cycle). The total number of edges emanating from this path is given by the sum of the



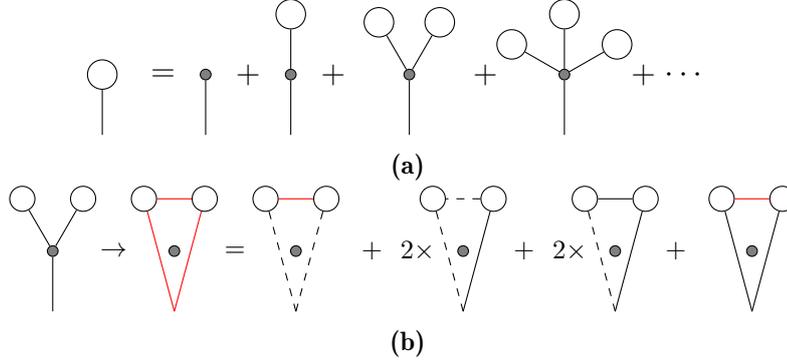
**Figure 3.5.** Example of the branching process before and after a 3-swap, starting at the leftmost node. (a) Before any operation the branching process arrives at a node of degree 3 (with a red border), leading to 2 components (in dark grey). (b) After the 3-swap, the branching process is already in a 3-cycle, each of its unvisited nodes belonging to one of the 2 components.

contributions of every unvisited node in the path. Each of these nodes has excess degree  $k$  with probability  $r_k$ , and thus leads to a component with a size generated by the function  $z g_r[\tilde{h}_R(z)]$ . (The size includes the node in the path, hence the  $z$  multiplying it.) Since these contributions are independent of each other, the distribution for the total number of edges will be simply given by the product of the corresponding generating functions. For  $l \leq q - 2$ ,  $l$  new components emerge, with total size (including all the vertices in the cycle, except the starting one) probability generated by  $\{z g_r[\tilde{h}_R(z)]\}^l$ . For  $l = q - 1$  and  $l = q$ ,  $q - 1$  components emerge, again with total size probability generated by  $\{z g_r[\tilde{h}_R(z)]\}^{q-1}$ . The total size of such cycle (excluding the starting vertex) and its emerging components is then generated by

$$C_q(z) = \sum_{l=0}^{q-2} (l+1) \phi_1^l (1 - \phi_1)^2 \{z g_r[\tilde{h}_R(z)]\}^l + [q \phi_1^{q-1} (1 - \phi_1) + \phi_1^q] \{z g_r[\tilde{h}_R(z)]\}^{q-1}. \quad (3.21)$$

Figure 3.6b shows a schematic representation of this function for a 3-swap.

This function can be used in the new  $\tilde{h}_R(z)$  if one also discounts the contribution of the target node, which is now of degree 0 and leads to no component. Therefore, the new  $\tilde{h}_R(z)$  is of the same form of Eq. 2.48 plus a term  $\tilde{h}_{R,q}(z)$  for



**Figure 3.6.** (a) Schematic representation of  $h_R(z)$  (see Eq. 2.48 on page 36). Small dots correspond to nodes, circles to components, edges are states with SCP equal to  $\phi_2$ . (b) In the new  $\tilde{h}_R(z)$  after a 3-swap, the fourth term in the sum (corresponding to  $r_k = r_2$ ), is changed according to  $C_q(z)$  (see Eq. 3.21). In this case, red edges correspond to newborn states with SCP equal to  $\phi_1$ , dashed edges to a failed conversion into a singlet (with probability  $1 - \phi_1$ ) and solid edges to a successful one.

each  $q$ -swap:

$$\tilde{h}_R(z) = 1 - \phi_2 + \phi_2 z g_r(\tilde{h}_R(z)) + \sum_{q \geq 2} \Pi_q \tilde{h}_{R,q}(z) \quad (3.22a)$$

$$\tilde{h}_{R,q}(z) = r_{q-1} \left\{ -(1 - \phi_2) - \phi_2 z \left[ \tilde{h}_R(z) \right]^{q-1} + C_q(z) \right\}. \quad (3.22b)$$

The latter Eq. 3.22b subtracts the original contribution of nodes with excess degree  $q - 1$  in the direct strategy and adds the contribution of the  $q$ -cycle.

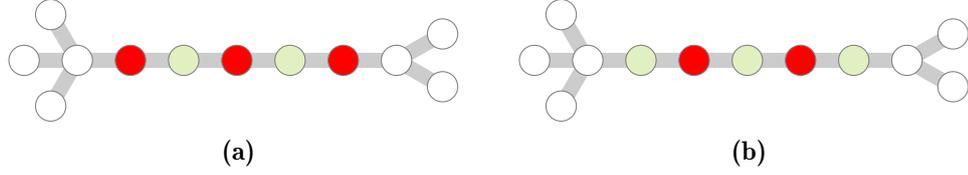
At this stage we can already calculate  $\tilde{\phi}_1^*$  as the smallest value of  $\phi_1$  for which there exists a positive solution

$$\tilde{u}^* = \tilde{h}_R(1) < 1 \quad (3.23)$$

to Eq. 3.22a at  $z = 1$ . It is easy to convince oneself that each separate contribution  $\tilde{h}_{R,q}(1)$  either increases or lowers the percolation threshold and therefore for the optimal strategy each  $\Pi_q$  is either 0 or 1.

### Starting at a $q$ -cycle: long-distance entanglement probability

For the new  $\tilde{h}_P(z)$  we need to consider that not all nodes of degree  $q$  are suitable targets of  $q$ -swaps, since they cannot be performed on adjacent nodes. Therefore, given a node of degree  $q$  there is a probability  $\eta_q$  that a  $q$ -swap can be performed on it. If the  $q$ -swap is performed on a node, then it changes its degree from  $q$  to zero and hence the contribution of reaching  $q$  components has to be substituted



**Figure 3.7.** Implementation of 2-swap in a cluster of 5 nodes with degree 2. Nodes are big circles: empty if their degree is different from 2, red if they are operated on, green if they are not. (a) Operations are done at nodes 1, 3 and 5, leading to  $\eta_2^{(\max)}$ . (b) Operations are done at nodes 2 and 4, leading to  $\eta_2^{(\min)}$ .

by that of being an isolated node:

$$\tilde{h}_P(z) = z g_p[\tilde{h}_R(z)] + z \sum_{q \geq 2} \Pi_q \eta_q p_q \left\{ 1 - [\tilde{h}_R(z)]^q \right\}. \quad (3.24)$$

By using the solution  $\tilde{u}^*$  of  $\tilde{u} = \tilde{h}_R(1)$  here, we can obtain the size of the giant connected component,

$$\tilde{S} = 1 - \tilde{h}_P(1) = 1 - z g_p(\tilde{u}^*) + z \sum_{q \geq 2} \Pi_q \eta_q p_q [1 - (\tilde{u}^*)^q]. \quad (3.25)$$

This gives the probability  $\tilde{P}_\infty = \tilde{S}^2$  that two distant nodes are connected by a path of singlets in the swapped entanglement percolation strategy.

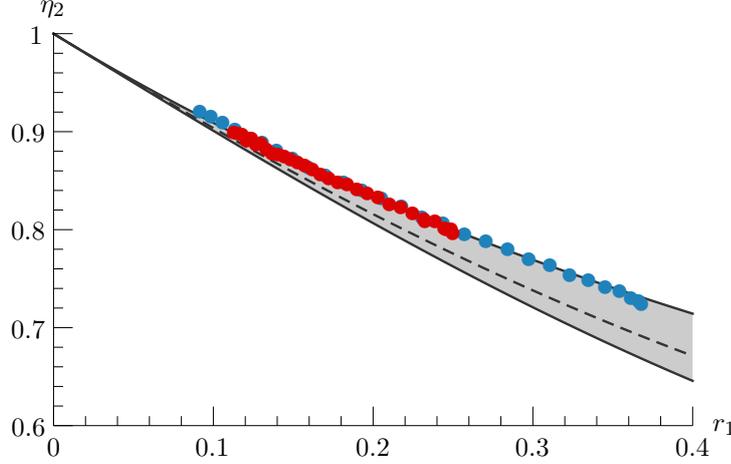
### 3.2.5 Strategies to implement $q$ -swaps

The probability  $\eta_q$  depends on which degrees are targets of  $q$ -swaps and on how the network is traversed to operate on the nodes. To compute its value we need to consider maximal clusters consisting of nodes where all vertices are of any target degree  $q$ —the border of such clusters is necessarily made of nodes of degree different from  $q$ , and hence operations can be done independently on every cluster.

Let us begin with the discussion of the simplest case in which only 2-swaps are performed. We consider that we select a random vertex of degree 2, perform a 2-swap on it, and then perform another 2-swap on every second node. Starting from a random vertex of degree 2, we find a cluster of vertices of same degree 2 whose size is  $s$  with probability  $s(1 - r_1)^2 r_1^{s-1}$ . By acting on every second node, there are two possible values for the number of operations done in each cluster,  $\lceil s/2 \rceil$  and  $\lfloor s/2 \rfloor$  (Figure 3.7), which coincide for  $s$  even. This gives a maximum and minimum value for  $\eta_2$ :

$$\eta_2^{(\max)} = \sum_{s \geq 1} s(1 - r_1)^2 r_1^{s-1} \frac{\lceil s/2 \rceil}{s} = \frac{1}{1 + r_1} \quad (3.26)$$

$$\eta_2^{(\min)} = (1 - r_1)^2 + \sum_{s \geq 2} s(1 - r_1)^2 r_1^{s-1} \frac{\lfloor s/2 \rfloor}{s} = \frac{1 - (1 - r_1)r_1^2}{1 + r_1}. \quad (3.27)$$



**Figure 3.8.** Probability  $\eta_2$  of performing a 2-swap, given a vertex of degree 2. Upper and lower solid lines correspond to Eqs. (3.26) and (3.27) respectively, and the dashed line to  $\eta_2^{(\text{rand})}$ . Blue dots are Erdős–Rényi network simulations, and red dots scale free network simulations. All the simulations were performed with  $N = 10^6$  and a Bread First Search traversal of the graph.

Note that in clusters of size  $s = 1$ , an operation is always done. Of course,  $\eta_2$  could be lower than  $\eta_2^{(\text{min})}$  if instead of acting on every two nodes, we just selected nodes at random.

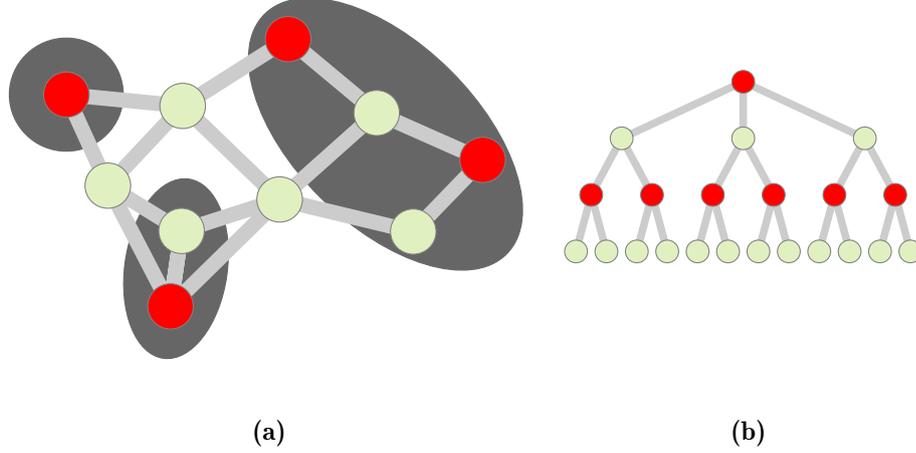
When operations are performed starting from a random vertex in each cluster of vertices with degree 2, one needs to take into account the number of vertices  $s$  and  $t$  at odd and even (including zero) distance from the first vertex: operations will be performed on a fraction  $t/(t+s)$  of the cluster. The probability  $\xi(s, t)$  of starting in a vertex of degree 2 such that it has  $s$  neighbors of degree 2 at odd distance and  $t$  at even distance is

$$\xi(s, t) = \begin{cases} \binom{2}{1+s-t} (1-r_1)^2 r_1^{s+t-1} t & \text{if } |s-t| \leq 1, \\ 0 & \text{otherwise.} \end{cases} \quad (3.28)$$

The derivation of this probability is easily seen by the method of generating functions explained later (see Eq. 3.34). Given the probability  $\xi(s, t)$ , then the value for  $\eta_2$  when operations are started at each cluster of target vertices is

$$\eta_2^{(\text{rand})} = \sum_{s,t} \frac{t}{t+s} \xi(s, t) = \frac{r_1 + (1-r_1)^2 \operatorname{atanh}(r_1)}{2r_1}. \quad (3.29)$$

This falls between the two probabilities in Eqs. 3.26 and 3.27,  $\eta_2^{(\text{min})} \leq \eta_2^{(\text{rand})} \leq \eta_2^{(\text{max})}$ . Figure 3.8 shows these three values together with numerical simulations performing 2-swaps by traversing the graph with a Breadth First Search: for each



**Figure 3.9.** (a) Example of a connected component with three clusters (in dark grey) of nodes of degree 2 and 3. Nodes in red denote a possible set of targets of 2, 3-swap. (b) Branching process in  $\eta_3^{(\text{rand})}$ , starting from a node inside the cluster of degree 3 and continuing to nodes of excess degree 2. 3-swaps are made on red nodes, which are the  $t$  nodes at even distance from the top one.

component, a starting node is selected at random, a  $q$ -swap is performed on it if it has the appropriate degree, and then neighbors are explored and  $q$ -swaps performed on them if they are adequate targets of  $q$ -swap (ie if they have the right degree *and* no  $q$ -swap has been performed on its neighbors yet). The numerical values for  $\eta_2$  are close to the maximum value because it is much more likely that the traversal of the graph started outside most of the degree 2 clusters (eg arriving through one of the white nodes in Figure 3.7), thus performing the maximum number of operations in them.

For general  $q$ , this probability can be found by generating functions similar to the ones used for percolation. As we said, the probability  $\eta_q$  depends on the target degrees  $\{q_i\}$  and on how the network is traversed. By  $\eta_q^{(\text{rand})}$  we denote the probability  $\eta_q$  when a  $q$ -swap is first done in a random vertex with target degree, and then the cluster of vertices with degree belonging to  $\{q_i\}$  is traversed by a Breadth First Search, performing  $q$ -swaps whenever possible (ie at every second step). After that, another vertex with target degree which has not yet been explored is selected, and its cluster traversed, until all target vertices have been checked. Such clusters consist of vertices of degree  $k \in \{q_i\}$  that are connected by at least one path whose vertices have also a degree in  $\{q_i\}$  and to which no more vertices of degree  $k$  can be added. Figure 3.9a shows an example of three of such clusters when the target degrees are 2 and 3.

A random vertex of degree  $k \in \{q_i\}$  belongs to a cluster with  $t$  vertices at even

distance (including itself) and  $s$  at odd distance with probability  $\xi(s, t)$ . In this cluster of size  $s + t$ ,  $t$   $q$ -swaps are made. The probability  $\eta_q^{(\text{rand})}$  is then

$$\eta_q^{(\text{rand})} = \sum_{t,s} \frac{t}{s+t} \xi(s, t). \quad (3.30)$$

The function generating  $\xi(s, t)$  can be computed similarly to Eqs. 2.48 and 2.50. In this case, it is a function of two variables:  $h_\xi(x, y) = \sum_{s,t \geq 0} \xi(s, t) y^s x^t$ . Two more distributions are needed:  $S(s, t)$  and  $T(s, t)$  are the probabilities of arriving at a vertex of the given degree (or degrees) which is at an odd or even distance from the starting vertex, respectively, and which belongs to a cluster of  $s$  extra vertices at odd distance, and  $t$  at even distance. The corresponding generating functions depend on each other:

$$h_S(x, y) = 1 - \sum_q \Pi_q r_q + y \sum_q \Pi_q r_q [h_T(x, y)]^{q-1}, \quad (3.31)$$

$$h_T(x, y) = 1 - \sum_q \Pi_q r_q + x \sum_q \Pi_q r_q [h_S(x, y)]^{q-1}, \quad (3.32)$$

and the function generating  $\xi(s, t)$  is

$$h_\xi(x, y) = x \sum_q \Pi_q [h_S(x, y)]^q. \quad (3.33)$$

This allows to compute  $\xi(s, t)$  by taking partial derivatives in  $x$  and  $y$ . As in the case of Eqs. 2.50 and 3.24,  $h_\xi(x, y)$  is in general a transcendental function and has to be solved numerically. However, in some cases it can be solved analytically. In the case of 2-swap only ( $\Pi_2 = 1$ ,  $\Pi_{q \neq 2} = 0$ ), Eq. 3.33 simplifies to the closed form

$$h_\xi(x, y) = \frac{x(1-r_1)^2(1+r_1y)^2}{(1-r_1^2xy)^2}. \quad (3.34)$$

The probability  $\xi(s, t)$  in Eq. 3.28 is then

$$\xi(s, t) = \frac{1}{s!t!} \left. \frac{\partial^s \partial^t h_\xi(x, y)}{\partial y^s \partial x^t} \right|_{x,y=0} = \binom{2}{1+s-t} (1-r_1)^2 r_1^{s+t-1} t \quad (3.35)$$

if  $|s-t| \leq 1$  and 0 otherwise.

Alternatively, for the case of a single target degree,  $\eta_q^{(\text{rand})}$  can also be computed exactly up to the  $n$ -th order in  $r_{q-1}$  by the branching process depicted in Figure 3.9b. The process begins at step 0, with  $k_0 = 1$  vertices of degree  $q$ . At step 1,  $k_1$  vertices out of  $qk_0 = q$  are of degree  $q$  with binomial probability

$$\binom{q}{k_1} r_{q-1}^{k_1} (1-r_{q-1})^{q-k_1}. \quad (3.36)$$

At following steps  $i \geq 2$  in the branching process, there are  $(q - 1)$  new vertices for each previous vertex of degree  $q$ . Thus, in every step,  $k_i$  vertices are of degree  $q$  with probability

$$\binom{(q-1)k_{i-1}}{k_i} r_{q-1}^{k_i} (1 - r_{q-1})^{(q-1)k_{i-1} - k_i}. \quad (3.37)$$

Operations are made on vertices at even steps. Note that every new step in the branching process involves higher orders in  $r_{q-1}$ . Therefore, the expansion of  $\eta_q$  up to order  $n$  is obtained by summing the contributions of the first  $n$  steps:

$$\eta_q^{(\text{rand})} = \sum_{\{k_i\}} \frac{\sum_{i=0}^{\lfloor n/2 \rfloor} k_{2i}}{\sum_{i=0}^n k_i} \binom{q}{k_1} \times \prod_{i=2}^n \binom{(q-1)k_{i-1}}{k_i} r_{q-1}^{\sum_{i=1}^n k_i} (1 - r_{q-1})^{q+(q-2)(\sum_{i=1}^{n-1} k_i) - k_n}, \quad (3.38)$$

where the sum in  $\{k_i\}$  sums for  $k_0 = 1$ ,  $k_1 = 0, 1, \dots, qk_0$  and  $k_{i \geq 2} = 0, 1, \dots, (q - 1)k_{i-1}$ .

### 3.2.6 Network examples

In this Section we present and discuss the results of the critical entanglement and long-distance entanglement probability for different complex networks. This networks have already been introduced in Chapter 2: the Bethe lattice (p. 11), the Erdős-Rényi (p. 17), an example of the configuration model, namely a scale-free network with an exponential cutoff (p. 21), and the Watts-Strogatz small world model with added shortcuts (p. 23). We will also present the results that would be obtained if one could implement the swapped strategy in a real-world network. For this example, we will use the World Wide Web snapshot of sites in the [nd.edu](http://nd.edu) domain (Albert et al., 1999).

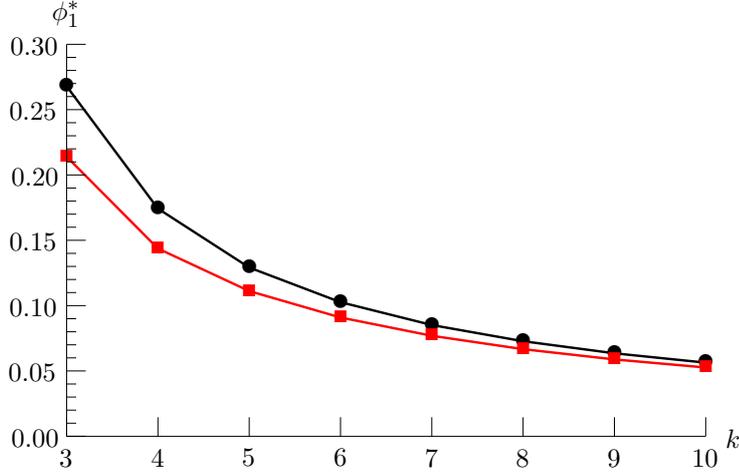
#### Bethe lattice

The degree distribution generating functions of the Bethe lattice with coordination number  $k$  are simply  $g_p(z) = z^k$  and  $g_r(z) = z^{k-1}$ . In this network, the phase transition occurs at  $\phi_2^* = (k - 1)^{-1}$ , ie is the solution  $\phi^* \leq 1$  of

$$\frac{1}{k-1} = 2\phi_1 - \frac{\phi_1^2}{2}. \quad (3.39)$$

In the Bethe lattice all vertices have the same degree, so it is only possible to apply one type of  $q$ -swap, that for  $q = k$ . The critical entanglement of the Bethe lattice modified by the  $k$ -swap is the solution of

$$\frac{1}{k-1} = \frac{2\phi_1 + \phi_1^k [\phi_1(k-1) - (k+1)]}{1 - \phi_1}. \quad (3.40)$$



**Figure 3.10.** Critical entanglement  $\phi_1^*$  of the Bethe lattice (in black) and of the Bethe lattice modified by a  $k$ -swap (in red), for coordination numbers  $k$  from 3 to 10. The quantum preprocessing gives always a better (lower) critical entanglement, except for the special case  $k = 2$  corresponding to an infinite one-dimensional chain (not shown in the plot).

The Bethe lattice with  $k = 2$  corresponds to an infinite one-dimensional chain. For this special case, the deterministic conversion of  $|\psi\rangle^{\otimes 2}$  into a maximally entangled state when  $\phi_1 \geq 2 - \sqrt{2}$ , possible in the direct strategy, beats the swapped one, which has the percolation threshold at  $\phi_1 = 1$ . In all the other cases  $k \geq 3$ , the quantum preprocessing via  $k$ -swap gives always a better threshold. Of course, as  $k$  grows the threshold goes to zero in both cases. Figure 3.10 shows the differences in the threshold for the first values of  $k$ .

### Erdős-Rényi

The Poisson degree distribution of the Erdős-Rényi network gives equal generating functions for the degree and excess degree,  $g_p(z) = g_r(z) = e^{c(z-1)}$ . This fact simplifies many calculations. For example, from Eq. 2.48 with  $h_R(1) = u$ , and using Eq. 2.51, the relation between the probability  $u$  and the size of the giant component is

$$u - 1 = -\phi_2 S. \quad (3.41)$$

Plugging it back to Eq. 2.51, the size of the giant connected component is the solution of the transcendental equation

$$S = 1 - e^{-c\phi_2 S}. \quad (3.42)$$

In this case this solution can be expressed in terms of the Lambert  $W$  function<sup>2</sup>,

$$S = 1 + \frac{1}{c\phi_2} W\left(-c\phi_2 e^{-c\phi_2}\right). \quad (3.43)$$

The percolation phase transition happens at the well-known point  $\phi_2^* = 1/c$ . In terms of the entanglement of  $|\psi\rangle$  (recall Eq. 3.16,  $\phi_2 = \min\{1, 2\phi_1 - \phi_1^2/2\}$ ), this means a critical entanglement for the original Erdős-Rényi network given by the solution of

$$\frac{1}{c} = 2\phi_1 - \frac{\phi_1^2}{2}. \quad (3.44)$$

The critical entanglement after a set of  $q$ -swap transformations can be found using the generating functions  $g_p(z)$  and  $g_r(z)$  of the Erdős-Rényi model in Eqs. 3.22 and 3.23. As an example, after the 2-swap and 3-swap operations, the critical entanglement is given by

$$\frac{1}{c} = \phi_2 + e^{-c}[-\phi_2 + c(2\phi_1 - \phi_1^2)] \quad (3.45)$$

and

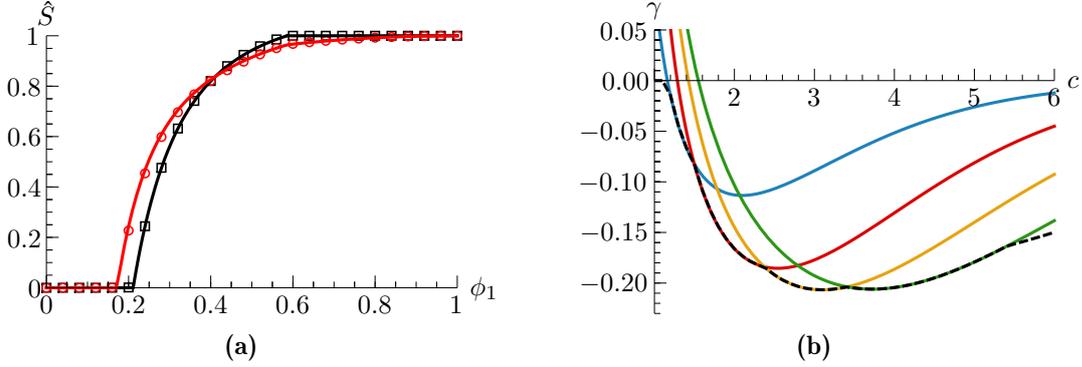
$$\frac{1}{c} = \phi_2 + ce^{-c}[-\phi_2 + c(1 + \phi_1 - \phi_1^2)], \quad (3.46)$$

respectively. Similarly, the probability of establishing long-distance entanglement in the large network limit  $\tilde{P}_\infty = \hat{S}^2$  (Eq. 3.19) depends on  $\tilde{S}$ , the solution of Eq. 3.25. These probabilities are plotted in Figure 3.11a for an Erdős-Rényi network of mean degree  $c = 2.5$  and 2,3-swap transformations, showing a perfect agreement between analytical and numerical results. As it can be seen, the gain

$$\gamma = \frac{\tilde{\phi}_1^* - \phi_1^*}{\phi_1^*} \quad (3.47)$$

in the critical entanglement is quite high. Figure 3.11b shows this gain depending on the  $q$ -swap that is performed, and also the optimal swapped strategy given this family of transformations. Depending on the network parameters (the mean degree) and on the transformation that is performed, the gain can be in some situations higher than 20%. The performance of different  $q$ -swaps depends on the mean degree  $c$ , usually improving the threshold those operations which act on nodes whose degree is around  $c$ .

<sup>2</sup>The Lambert  $W(z)$  function is defined by the equation  $z = W(z)e^{W(z)}$  for any complex number  $z$ .



**Figure 3.11.** Erdős–Rényi network. (a) Normalized size  $\hat{S}$  of the giant component as a function of  $\phi_1$  for  $c = 2.5$ , before (black squares) and after (red circles) 2,3-swap. Lines are analytical results, dots are simulations of a single network of size  $N = 10^6$ . (b) Gain  $\gamma$  as a function of the mean degree  $c$  after 2,3-swap (blue line), 2,3,4-swap (yellow line), 2,3,4,5-swap (green line) and optimal  $q$ -swaps (dashed black line).

### Scale free

As a model of a scale-free network we consider the configuration model with a power-law degree distribution with exponential cutoff (Eq. 2.19):

$$p_k = Ck^{-\tau} e^{-k/\kappa}, \quad (3.48)$$

where  $C$  is a normalization constant. The generating functions of the degree and excess degree in this case are

$$g_p(z) = \frac{\text{Li}_\tau(e^{-1/\kappa}z)}{\text{Li}_\tau(e^{-1/\kappa})} \quad (3.49)$$

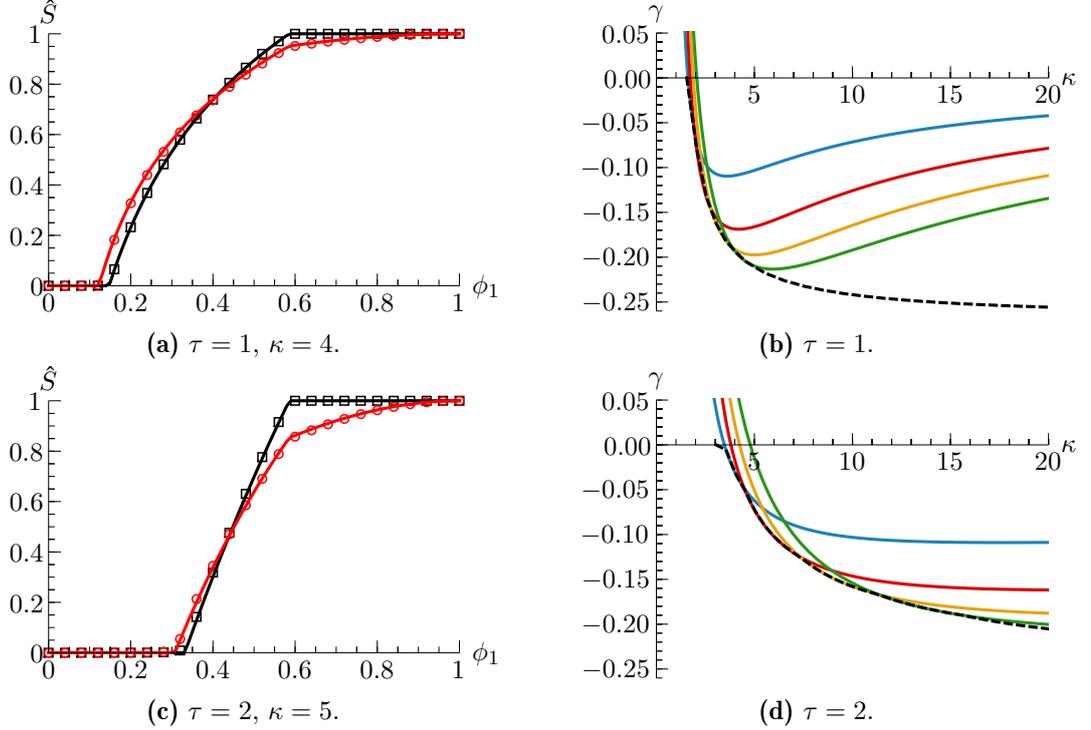
and

$$g_r(z) = \frac{\text{Li}_{\tau-1}(e^{-1/\kappa}z)}{z\text{Li}_{\tau-1}(e^{-1/\kappa})}, \quad (3.50)$$

respectively. Here  $\text{Li}_n(z)$  is the  $n$ -th polylogarithm<sup>3</sup> of  $z$ . The critical entanglement of the direct strategy is given by

$$\phi_2^* = \left[ \frac{\text{Li}_{\tau-2}(e^{-1/\kappa})}{\text{Li}_{\tau-1}(e^{-1/\kappa})} - 1 \right]^{-1}. \quad (3.51)$$

<sup>3</sup>The polylogarithm  $\text{Li}_n(z)$  is defined as the sum  $\text{Li}_n(z) = \sum_{k=1}^{\infty} \frac{z^k}{k^n}$ . The case  $n = 1$  is related to the natural logarithm,  $\text{Li}_1(z) = -\log(1 - z)$ . It also has an alternative definition via repeated integrals,  $\text{Li}_{n+1}(z) = \int_0^z \frac{\text{Li}_n(t)}{t} dt$ .

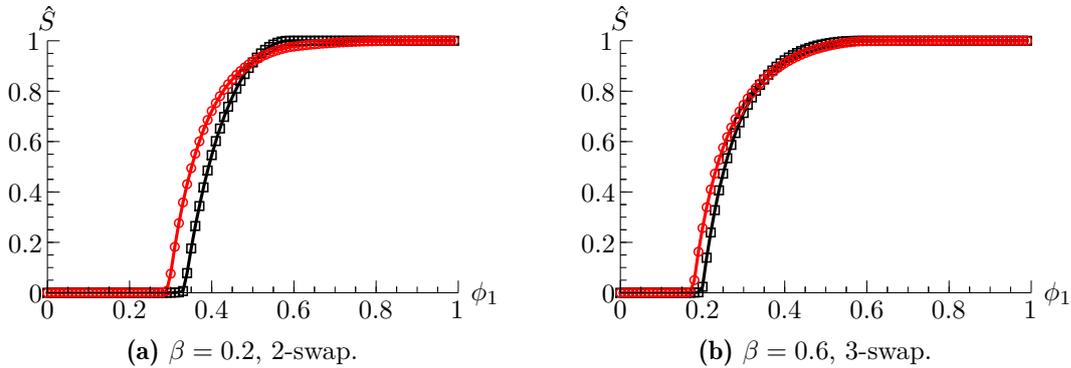


**Figure 3.12.** Scale-free network, with a power-law degree distribution with an exponential cutoff, see Eq. 3.48. (a) and (c) are the normalized size  $\hat{S}$  of the giant component as a function of  $\phi_1$ , before (black squares) and after (red circles) 2, 3-swap. Lines are analytical results, dots are simulations of a single network of size  $N = 10^6$ . (b) and (d) are the gain  $\gamma$  as a function of the cutoff  $\kappa$  after 2-swap (blue line), 2, 3-swap (red line), 2, 3, 4-swap (yellow line), 2, 3, 4, 5-swap (green line) and optimal  $q$ -swaps (dashed black line).

We do not give the explicit equations for the critical entanglement for the swapped strategy, but again they can be obtained by using the generating functions  $g_p(z)$  and  $g_r(z)$  in Eqs. 3.22 and 3.23. Figure 3.12 show the results for the giant connected component evolution and the gain in scale free networks with  $\tau = 1$  and  $\tau = 2$ . Note that in this case the gain can be of around 25%.

### Watts-Strogatz small world model with added shortcuts

The Watts-Strogatz model has high clustering and a correlated degree distribution. The tree-like assumption does not hold, and the above analytic approach cannot be used. However, numerical simulations show that  $q$ -swaps can also provide an improvement in the percolation threshold,  $\tilde{\phi}_1^* < \phi_1^*$ . These results are summarized in Figures 3.13 and 3.14a. The former shows the critical entanglement and the size of the giant connected component for different shortcut probabilities  $\beta$  and different  $q$ -swaps, with a behavior that is qualitatively equivalent to that of networks with



**Figure 3.13.** Watts-Strogatz small world model ( $k = 1$ ) with added shortcuts. Network size is  $N = 10^6$ . Normalized size  $\hat{S}$  of the giant component as a function of  $\phi_1$  in a network where shortcuts are added with probability  $\beta$ , before (black squares) and after (red circles)  $q$ -swap. Solid lines join dots and are a guide to the eye.

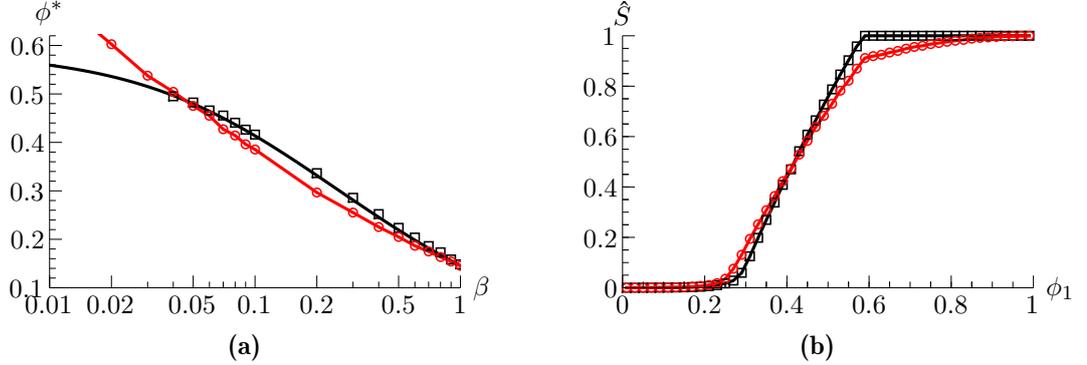
uncorrelated degree distribution. The latter shows the comparison between the critical entanglement of an unmodified and a modified network as a function of the shortcut probability. In some cases, however, the direct strategy is superior to the swapped strategy for small values of  $\beta$  (in the case shown,  $k = 1$  and 2-swap, for  $\beta$  below  $\approx 0.05$ ).

#### A real-world example: the World Wide Web

As a real-world network we take the example of a scale-free network consisting of *World Wide Web* sites in the [nd.edu](http://www.nd.edu) domain (Albert et al., 1999)<sup>4</sup>. In this case we introduce an artificial cutoff by neglecting nodes with degree  $k \geq 15$ , leaving a graph with 142 192 nodes and 170 352 edges. Results after a 2, 3-swap are shown in Figure 3.14b, showing an enhancement. However, in this case finite-size effects make it difficult to appreciate the exact critical entanglement.

As a final remark, note that, for all networks in general, it may be counterproductive to perform  $q$ -swaps. In the above figures we see that for some values of  $\phi_1$  the giant connected component fraction  $S$  without preprocessing is larger than  $\hat{S}$ . This often happens around  $\phi_1 = 2 - \sqrt{2}$ . This is precisely the point where the edges in the unmodified network can be directly converted into singlets with  $\phi_2 = 1$ , ie all connections become ideal channels and  $S$  attains its maximal value  $S = S_1$ . Obviously at this stage any preprocessing cannot further increase the size of the connected component, and it will most likely decrease it.

<sup>4</sup>The network dataset is available at <http://www.nd.edu/~networks/resources.htm>.

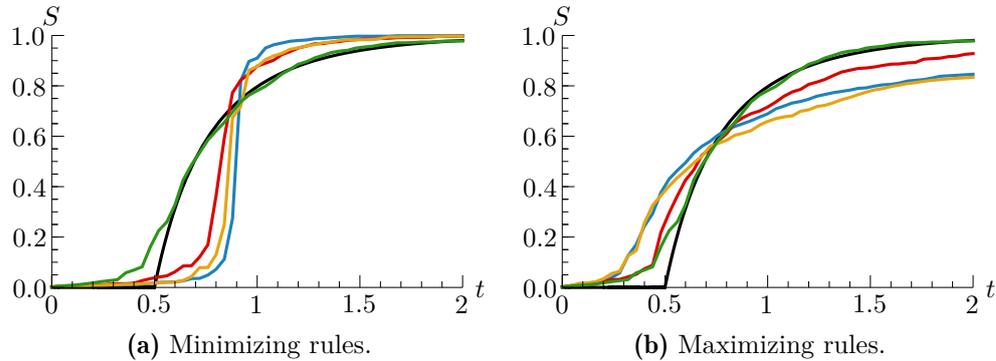


**Figure 3.14.** (a) Watts-Strogatz small world model,  $N = 10^6$ . Critical entanglement  $\phi_1^*$  as a function of shortcut probability  $\beta$  before (black squares) and after (red circles) 2-swap. The black line is the analytic result from Moore and Newman (2000), the red line joins dots and is a guide to the eye. (b) World Wide Web network (Albert et al., 1999), with a cutoff at  $k = 15$ , before (black squares) and after (red circles) 2,3-swap. Solid lines join dots and are a guide to the eye.

### 3.2.7 Explosive percolation: advance and delay of the transition

Up to now, we have considered entanglement percolation in networks where connections are built on pure, non-maximal bipartite entangled states, and have studied a local quantum preprocessing of the network that can significantly decrease the percolation threshold and therefore allow quantum communication for a lower level of entanglement. Decreasing the percolation threshold was of course a goal motivated by the need to allow quantum communication with less resources. In a different context, however, the goal can be instead to increase the value of the critical point (Bohman and Frieze, 2001). A clear example is the spread of a disease in a community. In this case, the threshold marks the point of the outbreak of an epidemic, and hence one is interested in a threshold as high as possible.

Recently, a stimulating paper by Achlioptas et al. (2009) has triggered an intense research into an abrupt percolation phase transition, that has been termed **explosive percolation**. Percolation processes are typically second-order phase transitions: the order parameter (the size of the giant component, in this case), changes *continuously* but its derivative (the mean component size) diverges at the critical point. In this thesis we have considered percolation as a process where edges have a certain occupancy probability. Alternatively, one can consider a different approach where edges from a given set are added one at a time (and hence introducing a sense of “time” in the evolution of the graph). This is analogous to the difference between the Gilbert and the Erdős-Rényi models. With this different approach to percolation, the standard, continuous percolation phase transition of the Erdős-Rényi model corresponds to a random choice of the added edge at each step.



**Figure 3.15.** Evolution of the relative size  $S$  of the largest cluster of networks with  $N = 500$  as a function of a normalized “time”  $t = m/N$  ( $m$  is the number of edges added). Blue, red and yellow lines correspond to the product, adjacent edge and triangular rules (Achlioptas et al., 2009; D’Souza and Mitzenmacher, 2010). The green line is the simulation of a standard Erdős-Rényi network with no selection rule. The black line is the analytic value for this standard Erdős-Rényi network. (a) In explosive percolation, edges are selected if they minimize these rules, showing a delay of the threshold and an abrupt change in  $S$ . (b) If edges are selected to maximize the rules, the effect is similar to that of a  $q$ -swap transformation.

Achlioptas et al. (2009) proposed a different rule in the selection of the added edge: two of the possible edges should be selected at random, say  $(u_1, v_1)$  and  $(u_2, v_2)$ , but only the one that minimizes the product of the sizes of components it merges is actually added, while the other one is discarded. With this **product rule**, the appearance of the giant component was delayed (the critical point was higher), but the phase transition appeared to be discontinuous (Achlioptas et al., 2009), an effect also observed on scale-free networks (Cho et al., 2009; Radicchi and Fortunato, 2009). Although the phase transition related to this selection rule was later shown to be actually continuous (da Costa et al., 2010; Riordan and Warnke, 2011), the transition is still very sharp for networks of very large sizes (up to order  $10^{18}$ ). Recently, a modified rule has been introduced that gives rise to a truly discontinuous phase transition (Panagiotou et al., 2011).

The idea behind explosive percolation is that the appearance of the giant component is *delayed* by choosing the edge  $(u_i, v_i)$  that *minimizes* the product rule. This keeps largest components similar in size in the subcritical regime and for a longer time, until a very small (presumably finite) number of additional edges joins them. Figure 3.15a shows this process for three different selection rules: the already presented product rule (Achlioptas et al., 2009), and the **adjacent edge** and **triangular rules** of D’Souza and Mitzenmacher (2010), which follow a similar protocol of choosing an edge which minimizes the growth of components. Other rules have also been presented (see eg Araújo and Herrmann, 2010).

This idea can be turned around. If, instead, one chooses the edge that *max-*

*imizes* the product rule (or the other two rules, in case they are used), the giant component appears *before* the standard threshold (Figure 3.15b), and the similarity with entanglement percolation is clear. One of the drawbacks of the product rule is that it is a rather artificial process—edges are selected one at a time, and the decision to actually add them or not is clearly nonlocal. Moreover, the process of adding one edge at a time is radically different from that on which entanglement percolation is based, with a probability associated to every edge. However, a better understanding of entanglement percolation could be obtained by investigating the effect of the delay or acceleration of component growth. This could lead to a better control of the critical entanglement needed to establish a long-distance singlet.

### 3.3 Entanglement in mixed states

A network with nodes connected by pure states is an abstraction that gives insight into the possibilities of long-distance entanglement in complex networks, enabling perfect teleportation between distant parties when at least a path of maximally entangled states is created. In realistic implementations, however, networks are noisy (Cirac et al., 1997; Duan et al., 2001; Kraus and Cirac, 2004; Kimble, 2008; Duan and Monroe, 2010; Ritter et al., 2012). In the present and the following Section we move to the study of entanglement distribution in networks with noise. As in the pure-state scenario, we consider a network where neighbors are connected by bipartite states  $\rho$ , which now are mixed. We begin in Section 3.3.1 with a brief presentation entanglement transformation and swapping with rank-two mixed states, that have been considered in the context of entanglement percolation (Broadfoot et al., 2009), and discuss how they could be used in a  $q$ -swap transformation of a mixed-state network. Then in Section 3.3.2 we turn to full-rank states, and present how the fidelity of these states decreases with the number of swappings. This will lead us to a different approach to entanglement distribution in a mixed-state scenario, where noise limits the maximum number of steps through which information can be repeated. We present this limited-path-length percolation in the following Section 3.4.

#### 3.3.1 States from amplitude-damping channels

If one assumes perfect local operations, so once the mixed-state connections in the network are established no more noise is introduced, a natural extension of entanglement percolation from pure-state to mixed-state networks is to purify the entanglement of each edge in the network into a pure, maximally-entangled state, and then use this state for perfect entanglement swapping, as in the ideal scenario. That is, to transform  $\rho^{\otimes n} \rightarrow |\Psi\rangle$  via LOCC, where  $n$  is the number of bipartite mixed states shared between two neighbors. Jané (2002) showed that this can be

done with a certain nonzero probability if and only if the states  $\rho$  are of the form

$$\rho_{\text{AD}} \equiv \rho(\lambda_0, \gamma, \lambda) = \lambda |\psi\rangle \langle \psi| + (1 - \lambda) |01\rangle \langle 01|, \quad (3.52)$$

where  $|\psi\rangle = \sqrt{\lambda_0} |00\rangle + \sqrt{1 - \lambda_0 - \gamma} |11\rangle + \sqrt{\gamma} |01\rangle$  and  $0 \leq \lambda \leq 1$ , and there are at least  $n = 2$  two copies of them (possibly with different parameters). States like these occur as the result of sending  $|\psi\rangle$  through an amplitude damping channel (Nielsen and Chuang, 2000). Given two states  $\rho(\lambda_0, \gamma, \lambda)$  and  $\rho'(\lambda'_0, \gamma', \lambda')$ , the singlet conversion probability is in this case

$$\phi_{\text{cla}} = 2\lambda\lambda' \min\{\lambda_0(1 - \lambda'_0 - \gamma'), \lambda'_0(1 - \lambda_0 - \gamma)\}, \quad (3.53)$$

and for  $\lambda = 1$  and  $\gamma = 0$  it reduces to the pure-state case described in Section 3.1.

Broadfoot et al. (2009, 2010b) considered networks with nodes connected by bipartite mixed states arising from an amplitude damping process, and showed that any two nodes in the network can generate a perfect singlet if and only if they are joined by at least two paths and the mixed states are like that of Eq. 3.52. So in particular, if one considers (possibly complex) networks as the ones in Section 3.2 but with edges  $\rho_{\text{AD}}^{\otimes 2}$  instead of  $|\psi\rangle^{\otimes 2}$ , the entanglement percolation results of the unmodified network hold, with the new edge probability defined by Eq. 3.53.

In the same work, Broadfoot et al. (2009, 2010b) also considered three different implementations of entanglement swapping that can be used to change the structure of the network, so again a quantum preprocessing can enhance the percolation properties. The first type of swapping is just that corresponding to the classical strategy, in which each edge is probabilistically converted to a singlet and then the swapping propagates it so distant nodes can be perfectly connected if they are in the same component. The second one, which they call **hybrid swapping**, first converts each edge to a pure state (but not necessarily maximally entangled) and then implements the swapping, which produces a state with some SCP  $\phi_{\text{hyb}}$ . Finally, the third one directly performs entanglement swapping for each of the copies of  $\rho$  in an edge, and then purifies the resulting distributed states into a singlet with probability  $\phi_{\text{dir}}$ . This last implementation is called **direct swapping**. The hybrid swapping gives always a probability greater or equal than the other implementations, and is thus the one that should be used. These ideas have been extended to networks with certain rank-three connections (states resulting from an amplitude damping channel can undergo phase errors) by combining error correction and percolation (Broadfoot et al., 2010a).

Broadfoot et al. apply their results to networks with a regular topology, but in principle one could also use the hybrid swapping to implement the  $q$ -swap transformation in a complex network with amplitude damping noise and at least four states per edge, and consider the probabilities  $\phi_{\text{hyb}}$  and  $\phi_{\text{cla}}$  instead of  $\phi_1$  and  $\phi_2$  in Eqs. 3.22a and 3.24 to find the percolation threshold and the giant component size of the modified networks.

### 3.3.2 General states

A general, rank-four two-qubit state connecting two nodes can be characterized by its maximal singlet fraction (or fidelity)  $F$ :

$$F(\rho) = \max_{|\Phi\rangle} \langle \Phi | \rho | \Phi \rangle, \quad (3.54)$$

where  $\Phi$  is a maximally entangled state. The singlet fraction  $F$  determines how good the state  $\rho$  is for teleportation: the optimal fidelity of teleportation  $f$  is related to the singlet fraction (Horodecki et al., 1999) by

$$f = \frac{2F + 1}{3}. \quad (3.55)$$

Using only LOCC, any general state can be depolarized into a Werner state by performing random local unitaries (Bennett et al., 1996c,b). The resulting state,

$$\rho_F = F |\Phi_{00}\rangle \langle \Phi_{00}| + \frac{1-F}{3} (|\Phi_{01}\rangle \langle \Phi_{01}| + |\Phi_{10}\rangle \langle \Phi_{10}| + |\Phi_{11}\rangle \langle \Phi_{11}|), \quad (3.56)$$

is diagonal in the Bell basis and has the same singlet fraction  $F$  of the original state. This state is entangled if  $F > 1/2$ . In this case, it can be used for teleportation, exceeding the classical limit  $f = 2/3$ . Note that, as long as the original  $\rho$  is entangled, this limit can be achieved even if  $F < 1/2$  by locally increasing the singlet fidelity through non-trace-preserving (Horodecki et al., 1997) or trace-preserving LOCC (Verstraete and Verschelde, 2003).

The Werner state  $\rho_F$  can also be rewritten as a combination of the maximally entangled state  $|\Phi_{00}\rangle$  and the maximally mixed state of two qubits  $\mathbb{1}/4$ ,

$$\rho_\alpha = \alpha |\Phi_{00}\rangle \langle \Phi_{00}| + (1-\alpha) \frac{\mathbb{1}}{4}, \quad (3.57)$$

with  $\alpha = (4F - 1)/3$ , so it is entangled if  $\alpha > 1/3$ .  $\rho_\alpha$  can be interpreted as the result of transmitting a pure  $|\Phi_{00}\rangle$  through a depolarizing channel (Nielsen and Chuang, 2000, p. 378).

As in the case of pure states, one can propagate entanglement between two nodes  $A$  and  $C$  to a third node  $B$  via entanglement swapping if  $B$  and  $C$  share also an entangled state (see Fig. 3.1 on page 55). If the states between  $A$  and  $C$ , and  $B$  and  $C$ , are Werner states  $\rho_{\alpha_1}$  and  $\rho_{\alpha_2}$ , respectively, then with probability  $\alpha_0\alpha_1$  two Bell states  $|\Phi_{00}\rangle_{AC} |\Phi_{00}\rangle_{CB}$  are obtained and the swapping protocol is successful, resulting in  $|\Phi_{00}\rangle_{AB}$ . In all other cases the protocol fails, resulting in the completely mixed case. That is, after swapping  $AB$  become entangled with a Werner state  $\rho_\alpha$  of parameter  $\alpha = \alpha_1\alpha_2$ . This means that the singlet fraction will decrease exponentially in a chain of swaps, and the resulting state will become useless for quantum teleportation: when a state is teleported through  $l$  identical

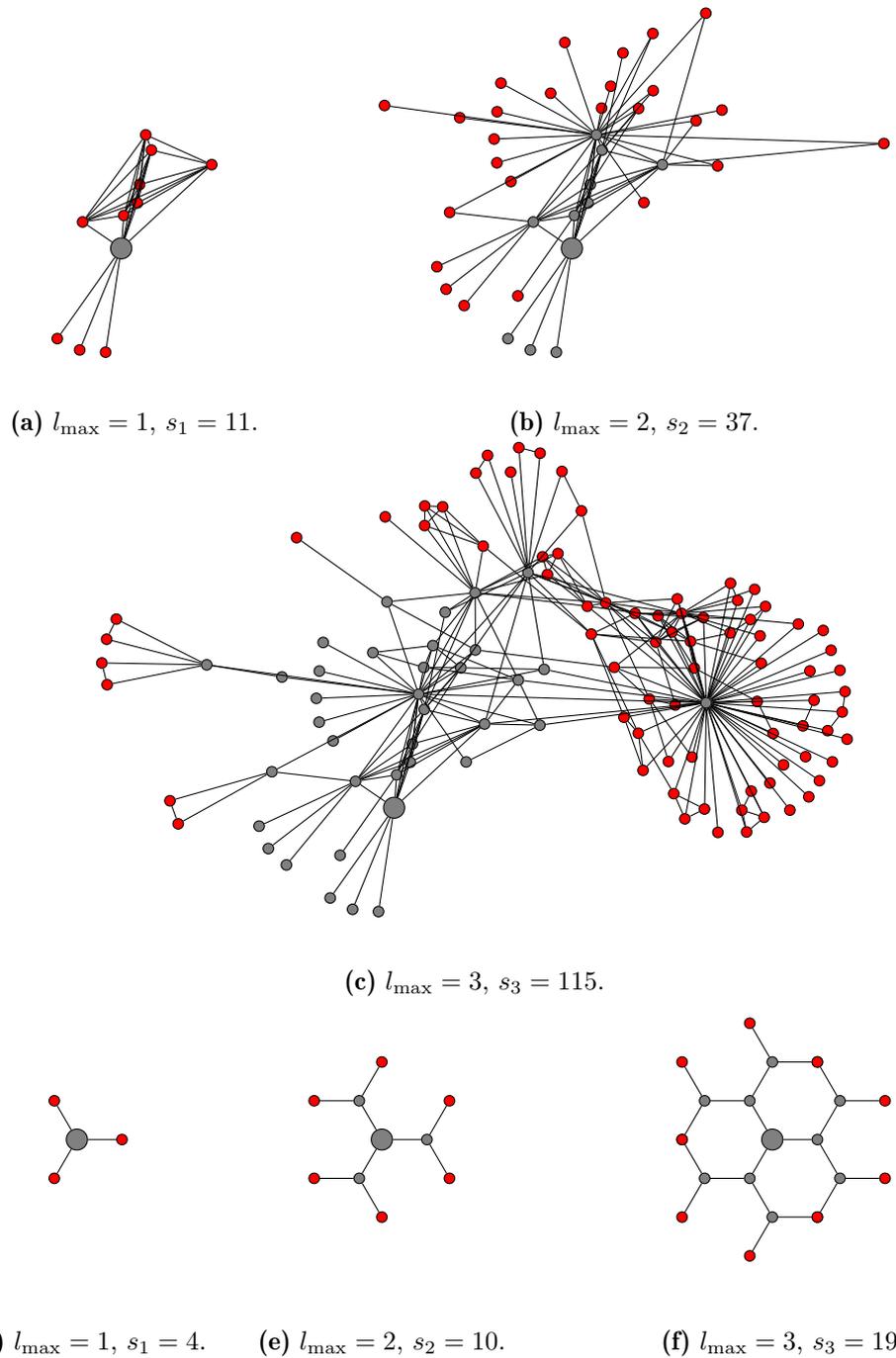
states  $\rho_\alpha$  (Briegel et al., 1998; Dür et al., 1999; Sen, De), its fidelity is 1 with probability  $\alpha^l$  and 1/2 otherwise, so its final fidelity is

$$f_l = \frac{1 + \alpha^l}{2}. \quad (3.58)$$

To overcome this difficulty, one can try to purify the entanglement resulting after each swap, and then use the resulting state in the following swap. This is the idea behind quantum repeaters (Briegel et al., 1998; Dür et al., 1999), where the goal is to produce a highly entangled state between two distant nodes in a one-dimensional network (see also Duan et al., 2001; Hartmann et al., 2007b; Sangouard et al., 2011). The idea to combine entanglement swapping and purification has been recently extended to general networks with an arbitrary architecture by Lapeyre Jr. et al. (2012). In another approach, Perseguers (2010) gives a fidelity threshold for the links above which long-distance quantum communication in the presence of noise is possible for an infinite cubic lattice, by mapping the problem to a three-dimensional noisy cluster state (Raussendorf et al., 2005). This approach strongly relies on the high symmetry of the lattice, and is hence not suited for complex networks.

### 3.4 Limited-path-length percolation in mixed-state networks

Communication in noisy networks can be considered from another perspective: the noise in the connections fixes a limit  $l_{\max}$  in the maximum number of nodes through which the information can be repeated before it becomes too corrupted (Dür et al., 1999), and one may ask if still a significant fraction of the network can be reached. In this limited-path-length scenario, the total number of vertices that a given node can communicate to (ie the nodes to which it can teleport information with fidelity above a threshold  $f_{\min}$ , or similarly the nodes at a distance  $l \leq l_{\max}$ , see Figure 3.16) also depends strongly on the structure of the communication network. This quantity is in fact related to the average path length  $l_{\text{av}}$  that we introduced in Section 2.1.4: all nodes within this distance constitute a significant fraction of the network. Therefore, for a path length limit  $l_{\max}$  above the average  $l_{\text{av}}$ , communication will be possible among an important number of nodes. Since the limiting  $l_{\max}$  is finite, the giant connected component appears only in models where  $l_{\text{av}}$  is also finite. In general, this only happens if the network size is finite too. The question then is whether a small  $l_{\max}$  will suffice to cover a significant fraction of the network. In finite  $d$ -dimensional networks, the average path length scales as  $l_{\text{av}} \sim N^{1/d}$ . However, the average path length of many complex networks scales *logarithmically* with the size of the network. This property is known as the small world effect, and appears also in many real world communication networks such as the Internet. In this case, to access a significant fraction of nodes, only



**Figure 3.16.** For a limited path length, cluster growth depends on the network topology. Here, clusters of limited path length  $l_{\max} = 1, 2, 3$  and size  $s_{l_{\max}}$  in the Web of Trust (a)–(c) and the honeycomb lattice (d)–(f). In the Web of Trust, the central node (bigger) corresponds to a key of who writes this. For each figure, nodes in red are those at the maximum distance 1, 2 or 3.

a small number of edges needs to be traversed. Small world models are therefore the first candidates, where losses by noise can be balanced by a short path length.

The problem of *limited path percolation* was also addressed in a different approach by López et al. (2007). In their model, they calculate the percolation phase transition under the assumption that, after deleting a number of edges, communication is only effective if the new minimum path length between two nodes does not exceed a multiple of the original path length between them (before edges are removed). Thus, in their study the limitation in the path length comes from the topology of the network and not from the nature of channels connecting nodes, which fixes a constant limit of nodes through which the information can be repeated.

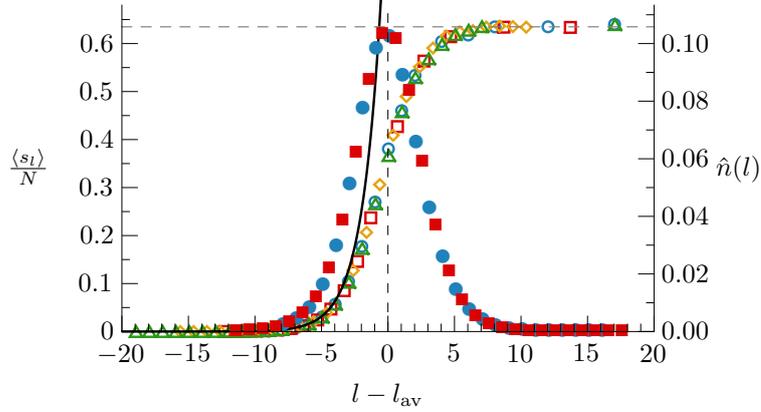
Here, we are interested in the number of nodes that can exchange quantum information with a given node above some fixed minimum fidelity  $f_{\min}$ , or similarly with what probability two random nodes can reliably communicate between them. We will consider a similar scenario as in the previous Sections, but replacing pure-state connections with generic entangled mixed states. Here, no quantum preprocessing will be possible. However, we will find that the complex network structure (in particular the small world effect) allows to interconnect a large number of nodes using the direct entanglement percolation strategy. We start with some numerical simulations and then derive the generating functions for limited path percolation and compute the limited average size in non-correlated networks and the Watts-Strogatz model.

### 3.4.1 Network model, minimum fidelity and maximum path length

We begin by simulating different models of networks. For simplicity we consider that edges hold a single copy of a two-qubit state  $\rho$  with maximum singlet fidelity  $F > 1/2$  so that the classical limit of  $f = 2/3$  in the teleportation fidelity can be exceeded. As we saw in Eq. 3.58, the teleportation fidelity  $f_l$  decreases exponentially with the distance  $l$  and makes such communication scheme useless in networks such as linear chains or regular lattices, where the typical distance between two nodes scales as the size of the network. However, as we discussed, the typical distance in many complex networks scales only logarithmically. The maximum distance  $l_{\max}$  that information can travel is fixed by the minimum fidelity  $f_{\min}$  required at the end point and by the purity  $\alpha$  of the channels:

$$l_{\max} = \left\lfloor \frac{\ln(2f_{\min} - 1)}{\ln \alpha} \right\rfloor. \quad (3.59)$$

This means that, even if there exists a path between a sender and a receiver in a network, it will only be useful if the length of this path is below a certain threshold.



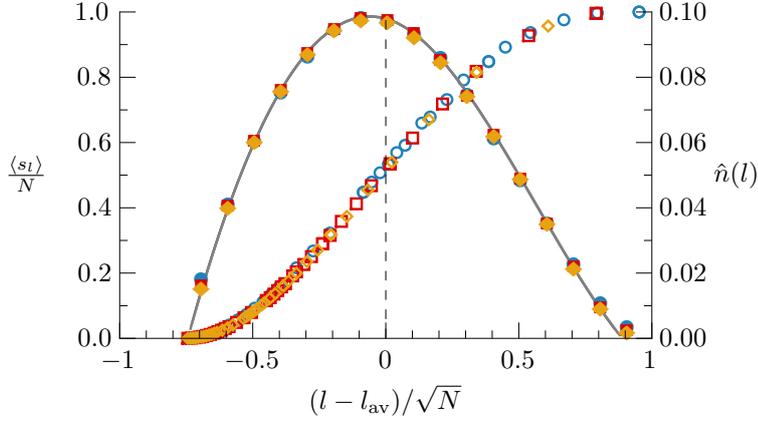
**Figure 3.17.** Normalized  $l$ -limited mean component size  $\langle s_l \rangle / N$  as a function of  $l - l_{av}$  of the Erdős-Rényi network with  $c = 2$  and network sizes  $N = 10^3, 10^4, 10^5, 10^6$  (empty blue circles, red squares, yellow diamonds and green triangles). Superposed filled markers is the shape of the path length distribution normalized with the total number of possible vertex pairs,  $\hat{n}(l) = 2n(l)/N(N - 1)$ , for  $N = 10^3$  (blue squares) and  $10^4$  (red circles). Solid black line is Eq. 3.65, horizontal dashed line is the square of the giant component at  $\phi_1 = 1$ , see Eq. 3.42.

### 3.4.2 Path length distribution and limited components

We have performed extensive simulations of networks where neighboring nodes share a state like the one in Eq. 3.57, and considered the classical limit as the minimum required fidelity,  $f_{\min} = 2/3$ . For small networks (up to  $N \sim 10^4$ ) we have performed the calculations over several network realizations and then averaged the results. For bigger networks, a single network realization is usually enough due to self-averaging. The  $l$ -limited average cluster size  $\langle s_l \rangle$  is a specially relevant parameter, which relates to the probability that two nodes can communicate with fidelity  $f > f_{\min}$ :

$$P_{AB}^{(l)} = \frac{\langle s_l \rangle - 1}{N - 1}. \quad (3.60)$$

We have thus calculated  $\langle s_l \rangle$  for different network models and sizes. In Figure 3.17 we plot the normalized size  $\langle s_l \rangle / N$  as a function of  $l - l_{av}$  for the Erdős-Rényi model, with average path length  $l_{av} \sim \ln N / \ln z$ . For different network sizes the curves collapse, supporting a linear  $N$ -dependence  $\langle s_l \rangle \sim N$  for fixed  $l$ . Similar results have been recently found for the average number of nodes at exact distance  $l$  from a random central node (Dorogovtsev et al., 2003; Shao et al., 2008). Regarding the dependence in  $l$ , our results show that the average size grows exponentially with  $l$  for  $l \ll l_{av}$ , but deviates from this behavior when  $l$  is close to the average path length, saturating to the maximum component size shortly after  $l_{av}$ . This deviation is due to the depletion of nodes at distance  $l > l_{av}$ . In the same Figure 3.17 we plot the path length distribution, ie the number of pairs  $n(l)$



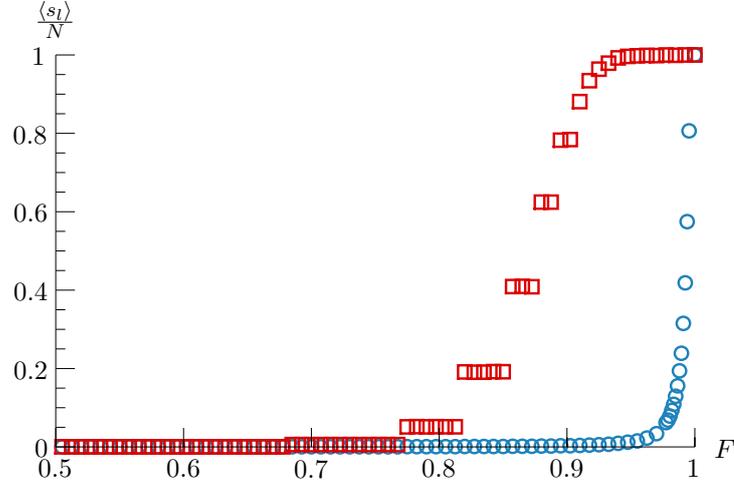
**Figure 3.18.** Normalized  $l$ -limited mean component size  $\langle s_l \rangle / N$  as a function of  $(l - l_{av})N^{-1/2}$  of the Honeycomb 2-dimensional network with network sizes  $N = 1014, 5046, 10086$  (blue circles, red squares and yellow diamonds). Superposed filled markers is the normalized histogram of the path length distribution.

separated by a distance  $l$ , normalized by the total number of pairs  $N(N - 1)/2$ . Again, both curves  $N = 10^3$  and  $N = 10^4$  collapse, thus supporting a dependence  $n(l) \sim N^2$ . We also found similar results for the scale free and the Watts-Strogatz models. It is interesting to note that, while  $l_{av}$  grows (logarithmically) with the size of the network, the width of the path length distribution remains constant. Thus, for large networks a small increase in the purity of the channel  $\alpha$  near  $l_{av}$  leads to an abrupt change in  $\langle s_l \rangle / N$ . This is in stark contrast to regular lattices, where both the mean and the width scale as  $N^{1/d}$ . For instance, in Figure 3.18 we plot  $\langle s_l \rangle / N$  and the path length distribution of the Honeycomb 2-dimensional lattice as a function of  $(l - l_{av})N^{-1/2}$ . The collapse of the curves confirms the  $N^{1/d}$  length-scale dependence.

As an example of a real work network, we considered the OpenPGP Web of Trust. Figure 3.19 shows the probability that two arbitrary nodes can communicate with fidelity  $f > 2/3$  as a function of the singlet fraction  $F$ . Again, the comparison with a Honeycomb lattice of the same size shows that the small world property of the complex networks allows for faithful communication between most of the nodes in the network for reasonable values of the noise, while in regular lattices this is only possible for nearly pure states.

### 3.4.3 Generating functions of the limited components

We will now formalize these observations by deriving analytical results for the relevant quantities. For this purpose, we now proceed to derive the generating functions for the limited path percolation problem. In this case, we are interested in the distribution of sizes  $s$  of the components that can be reached by only  $l$  steps



**Figure 3.19.** Normalized  $l$ -limited mean component size  $\langle s_l \rangle / N$  as a function of singlet fidelity  $F$  in the biggest component OpenPGP Web of Trust (red squares) and a honeycomb 2-dimensional lattice (blue circles),  $N \sim 3 \cdot 10^4$ .

through edges that are always occupied. As in the non limited case, there are two different distributions  $P_s^{(l)}$  and  $R_s^{(l)}$  for the cases where a random vertex or a random edge are selected. The two corresponding generating functions,  $h_P^{(l)}(z)$  and  $h_R^{(l)}(z)$ , read as

$$h_P^{(l)}(z) = \begin{cases} z & \text{for } l = 0, \\ z g_p [h_R^{(l-1)}(z)] & \text{for } l \geq 1, \end{cases} \quad (3.61)$$

and

$$h_R^{(l)}(z) = \begin{cases} z & \text{for } l = 0, \\ z g_r [h_R^{(l-1)}(z)] & \text{for } l \geq 1. \end{cases} \quad (3.62)$$

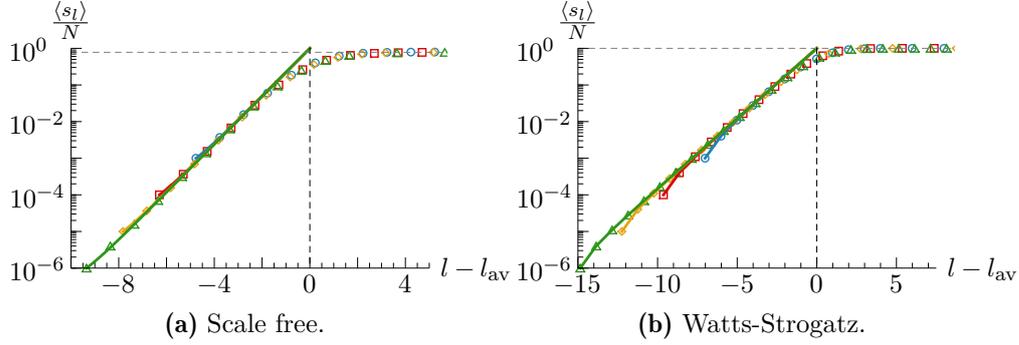
Note that all edges are occupied with probability one. This could be easily generalized to a different occupancy probability, by a change of  $h_R^{(l)}(z)$  in Eq. 3.62 to

$$\begin{cases} z & \text{for } l = 0, \\ 1 - \phi + \phi z g_r [h_R^{(l-1)}(z)] & \text{for } l \geq 1, \end{cases} \quad (3.63)$$

but this is not needed here.

As before, we are now ready to obtain the  $l$ -limited average size,

$$\langle s_l \rangle = \left. \frac{d h_P^{(l)}(z)}{d z} \right|_{z=1} = 1 + g_p'(1) h_R^{(l-1)}(1). \quad (3.64)$$



**Figure 3.20.** Normalized  $l$ -limited mean component size  $\langle s_l \rangle / N$  as a function of  $l - l_{av}$  for (a) the scale-free network and (b) the Watts-Strogatz network. Points are simulation results for network sizes  $N = 10^3, 10^4, 10^5, 10^6$  (blue circles, red squares, yellow diamonds, green triangles), solid lines correspond to Eq. 3.65 and 3.69, horizontal dashed lines are the values of  $S_1^2$ .

By solving the recurrence equation given by  $h_R^{(l)}(1)$  with the boundary condition  $h_R^{(0)}(1) = 1$  we find

$$\langle s_l \rangle = \begin{cases} 1 & \text{for } l = 0, \\ 1 + g_p'(1) \frac{1 - [g_r'(1)]^l}{1 - g_r'(1)} & \text{for } l \geq 1. \end{cases} \quad (3.65)$$

This equals to the probability that any two nodes will be able to communicate with fidelity above  $f_{\min}$ . Figure 3.20a shows this result for the scale-free model, with very good agreement between theoretical and numerical results below  $l_{av}$ .

As we discussed above, the derivation leading to this exponential growth of  $\langle s_l \rangle$  is valid for  $l$  well below  $l_{av}$ . Our numerical simulations, however, show that the validity of this approximation can be extended to values near  $l_{av}$ . Figure 3.17 shows that the path length distribution is very peaked around  $l_{av}$ , and its width is independent of  $N$ . This suggests, on one hand that our analytical approach holds true for values of  $l$  that fall out of this finite width (approaching from below)—see Figure 3.20. On the other hand, the finite width implies that if  $l$  is a few steps beyond  $l_{av}$  then most of the nodes in the components will be reached before the limit distance is attained. In this situation, Eqs. 3.61 and 3.62 approach the non-limited case of Eqs. 2.50 and 2.48 with  $\phi_1 = 1$ , and the size of the giant component  $S_l$  tends to the non-limited size  $S_1$ . Therefore, for networks with the small world property, ie  $l_{av} \sim \log N$ , one can interconnect with a threshold fidelity (say, the classical benchmark  $f = 2/3$ ) any arbitrary pair of nodes in the network provided that the singlet fraction of the edges scales as  $F = 1 - O(1/\log N)$  with the size of the network, which is clearly less stringent than the analogous constrain for  $d$ -dimensional networks  $F = 1 - O(N^{-1/d})$ .

We also consider the Watts-Strogatz model, which has a base circular lattice of size  $N$  with an average of  $\beta N$  randomly added shortcuts. In this case the derivation of

the probability that a random vertex belongs to an  $l$ -limited cluster of size  $s$ ,  $P_s^{(l)}$ , and its generating function  $h_P^{(l)}(z)$  uses the formalism of “local clusters” introduced in [Moore and Newman \(2000\)](#). These “local clusters” are clusters in the base lattice (without considering the shortcuts). For a given  $l$ , the “local cluster” is always of size  $2l + 1$ . Then, a shortcut at distance  $\lambda - 1$  from the starting vertex leads to a (global) cluster of size  $s'$  with probability  $P_{s'}^{(l-\lambda)}$ . A random shortcut emerges from the starting vertex with probability  $1/N$ , from a vertex at distance  $\lambda$  with probability  $2/N$ , and lies outside the local cluster with probability  $(N - 2l + 1)/N$ . Hence, that shortcut will lead to a cluster of size  $s$  with a probability given by the generating function:

$$f(z) = 1 - \frac{1}{N} \left( 2l - 1 - h_P^{(l-1)}(z) - 2 \sum_{\lambda=2}^l h_P^{(l-\lambda)}(z) \right) \quad (3.66)$$

There are  $2\beta N$  shortcut end-points that can similarly contribute to the total size of the cluster. Recalling that the generating function of the sum of sizes is the product of the generating function of each size, we find

$$h_P^{(l)}(z) = z^{l+1} f(z)^{2\beta N} \quad (3.67)$$

where  $z^{(l+1)}$  is the generating function corresponding to the starting “local” cluster. In the limit of large  $N$  this can be simplified to

$$h_P^{(l)}(z) = z^{1+2l} e^{-2\beta \left[ 2l-1 - h_P^{(l-1)}(z) - 2 \sum_{\lambda=2}^l h_P^{(l-\lambda)}(z) \right]}. \quad (3.68)$$

Again, we can obtain the limited average size by taking the first derivative at  $z = 1$ . For  $l = 0$ ,  $\langle s_0 \rangle = 1$ . For  $l \geq 1$ , this results in the recurrence equation

$$\langle s_l \rangle = 1 + 2l + 2\beta \left( \langle s_{l-1} \rangle + 2 \sum_{\lambda=0}^{l-2} \langle s_\lambda \rangle \right) = \langle s_{l-1} \rangle + 2 + 2\beta (\langle s_{l-1} \rangle + \langle s_{l-2} \rangle), \quad (3.69)$$

which can be exactly solved (in the latter expression, one has to use  $\langle s_1 \rangle = 3 + 2\beta$ ). [Figure 3.20b](#) shows this result. We want to stress the fact that from these generating functions, [Eqs. 3.61](#) and [3.68](#), one can also calculate the probability  $P_s^{(l)}$  up to any  $s$  by solving  $s + 1$  iterations of them and using [Eq. 2.36](#).

As a final remark, we want to emphasize that in complex networks the transition from the regime with small limited components to that of a component that spans the full non-limited scenario is abrupt. This means that any increase in the channel purity  $\alpha$ , even if small, can lead to a critical effect if it is around the region  $l \sim l_{av}$ .

## CHAPTER 4

---

### Distribution of multipartite entanglement

---

As we have said and exemplified already several times, the structure of a network strongly affects its functionality. Be it a real-world communication network or an abstract model capturing the working of a complex system, not only the type of the relation between the elements of the system but—and sometimes even more importantly—the shape and organization of such relations define the way it behaves. In the previous Chapter, we have seen that quantum information is no exception, and the distribution of bipartite entanglement in a quantum network depends on its structure. Quantum mechanics offer in addition the possibility to enhance the functionality of a network by acting on its structure. In that case, this was done by means of quantum operations that rewire the connections with no need to “physically” modify the underlying communication network. Interestingly, this can be done even when only partial information about this structure is known, and with only restricted control over the local structure. We studied the case of complex networks, that model a wide spectra of real-world systems and include most relevant classical communication networks.

Bipartite entanglement is indeed a key ingredient for many quantum information applications. In particular, if one is able to share a perfect, maximally entangled state with another party it amounts to a perfect quantum channel—even if only for a single use. Maybe one of the reasons of this ubiquity of bipartite entanglement in most quantum protocols is that it is already well understood (Plenio and Virmani, 2007; Horodecki et al., 2009). On the contrary, understanding of multipartite entanglement is still in its infancy, even for very small systems of few qubits. Advancing in this understanding could shed some light on the design of new multipartite applications.

An effective way to do that is to examine some classes of states. In this sense,

a relevant and large family of multipartite entangled states that has been studied is that of graph states (Hein et al., 2004). For instance, they include paradigmatic states like Greenberger-Horne-Zeilinger states (Greenberger et al., 1989), cluster states (Briegel and Raussendorf, 2001), and codewords of error-correcting codes (Schlingemann and Werner, 2001), they provide some novel quantum communication applications, like secret entangled-state distribution (Dür et al., 2005a) or can be used to implement existing protocols like quantum repeaters (Zwerger et al., 2012), they are useful in the study of non-locality (Gühne et al., 2005), and most importantly they include states which are universal resources (Van den Nest et al., 2006) for measurement-based quantum computation (Raussendorf and Briegel, 2001; Raussendorf et al., 2003; Briegel et al., 2009). These states are the ones that arise from bipartite entangling interactions within a number of qubits that are in an initial separable state. Their name comes from the fact that they can be associated to a graph in which nodes correspond to those qubits, and edges to the interactions that entangle them. This allows an efficient description of such states with relatively few parameters, but at the same time they still have a rich variety of features. Current experiments (Lu et al., 2007; Monz et al., 2011) have succeeded in the experimental implementation of GHZ and cluster states up to 14 qubits (Monz et al., 2011). Graph states can also be extended to states arising from interactions with different phase, which are then associated to a graph with weighted edges, each weight depending on the phase of the interaction (Dür et al., 2005b; Calsamiglia et al., 2005; Hartmann et al., 2005). And quite recently they have also been extended to states arising from multisite—instead of bisite—interactions. These are the so-called locally maximally entangleable (LME) states (Kruszynska and Kraus, 2009).

Following the motivation of the previous Chapter, here we turn now to study the distribution of multipartite entanglement in the form of graph states. We will address how noise affects the resulting state, and how can we deal with it. To do so, in this Chapter we consider the realistic scenario of noisy network channels and a small, but non-negligible, amount of noise in the local operations and measurements. Again, we are interested in networks with a complex structure. The global structure can be unknown and only local information may be available to every node. Our goal here is to propose a protocol to create a large graph using this underlying network of noisy channels. The protocol should tolerate channels and operations with errors, scale efficiently with the size of the network, and work for any network topology, and in particular for complex networks.

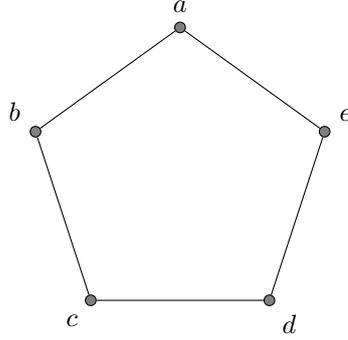
Channels linking separate nodes in a network are typically noisy, and pose the main caveat to the creation of distributed multipartite entangled states with high fidelity. To overcome this, there exist bipartite (Bennett et al., 1996b; Deutsch et al., 1996) and multipartite (Dür et al., 2003; Aschauer et al., 2005; Kruszynska et al., 2006; Glancy et al., 2006; Kay and Pachos, 2007) entanglement purification protocols that allow either to generate highly purified Bell pairs, which can latter be used to teleport an arbitrary graph state, or to directly purify the desired graph

state. These recursive protocols tolerate a reasonable amount of noise in local operations, but require a number of initial copies that grows exponentially with the size of the state. Other proposals do not use postselection, making the purification efficient in terms of the size of the graph state, but come at the expense of a stricter noise threshold (Goyal et al., 2006). Another option is to use entanglement pumping without postselection to obtain efficient purification when constructing the graph state edge by edge (Campbell, 2007). In different approaches, the graph state is created by a probabilistic growth using non-deterministic entangling operations (Kieling et al., 2007a; Rohde and Barrett, 2007; Campbell et al., 2007a,b; Matsuzaki et al., 2010). Recently, Carle et al. (2012) have also proposed a purification protocol for the bigger class of LME states.

Generating a graph state in a large complex network poses two main restrictions. First, one does not necessarily know the exact structure of the network, so one needs a protocol that acts only on small subgraphs, independently of the rest of the network and without need to coordinate. This means, for example, that a postselection protocol, that depends on the colorability of the graph, does not satisfy this demand. The same happens for other protocols like the bandaid (Goyal et al., 2006). Second, the large size of the network makes it unfeasible to purify a full graph state as in the postselection protocol. Instead, one needs a protocol that is efficient for large networks. Again, a possible solution is to purify only small subgraphs whose size do not depend on  $N$ .

Here, we investigate the advantages of generating and purifying small GHZs that reproduce the local structure of the network, and merge them in order to distribute a network-wide graph state. GHZs have a fixed size that depends on the degree of each node and is thus independent of the size of the network. Hence, this protocol is efficient in the size while it still maintains the high thresholds of the recurrence schemes. We benchmark this protocol with two other protocols that generate high-fidelity bipartite states between a node and the rest of the network, which are then used to distribute a locally generated graph state. We use the fidelity of the graph state as a figure of merit to compare the three protocols. The fidelity decays exponentially with the size of the network for a constant level of noise, so we also use its decay rate. It turns out that both quantities can be understood as the partition function and free energy of a thermodynamic system, respectively, and thus standard methods of statistical classical mechanics are readily used.

The Chapter is structured as follows. First, in Section 4.1 we introduce graph states and their formalism. In Section 4.2 we present the network and the noise model we consider. We then review in Section 4.3 the multipartite purification protocol of Dür et al. (2003) that we will use later, and give the first-order expression of the fixed point of the purification of a GHZ. Then, in Section 4.4 we present the protocol that we propose and two other protocols based on bipartite purification that we use as a comparison. In Section 4.5 we present the two figures of merit—



**Figure 4.1.** Graph state of 5 qubits. Dots represent qubits, initially in state  $|+\rangle$ . Edges represent entangling interactions between qubits. Alternatively, in the stabilizer formalism, there is an operator, eg  $K_b = X_b Z_a Z_c$ , associated to each qubit.

the fidelity of the graph state and its decay rate—, and relate them with an analog partition function and free energy of a classical Ising-type Hamiltonian. Finally, in Sections 4.6 and 4.7, we apply the protocols to the creation of a linear cluster state, for which we obtain exact results, and of a graph state associated to a complex network.

## 4.1 Graph states

A graph state (Hein et al., 2004) is a quantum state associated to a graph  $G$ . There exist two equivalent ways to establish this correspondence (see Figure 4.1). One has a clearer physical meaning, where the graph state is constructed by applying successive interactions between qubits initially prepared in a separable state. The other one describes the graph state in terms of its stabilizer, a subgroup of the Pauli group on  $N = |V|$  qubits. This second description turns out to be very convenient, as the group can be compactly described by the  $N$  generators of the stabilizer group. In this Section, we first introduce these two formalisms, and then present mixed graph states and operations under these descriptions.

### 4.1.1 Graph states in the interaction picture

The interaction picture describes graph states as prepared from an initial separable state by a successive application of entangling interactions. Let  $X$ ,  $Y$  and  $Z$  be the **Pauli operators** (often denoted as  $\sigma_x$ ,  $\sigma_y$  and  $\sigma_z$ , or  $\sigma_1$ ,  $\sigma_2$  and  $\sigma_3$ ). These operators, together with the identity  $I$  (also denoted  $\mathbb{1}$  or  $\sigma_0$ ) and the multiplicative factors  $\pm 1$ ,  $\pm i$ , form the **Pauli group** for a single qubit,

$$\mathcal{P}_1 = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}. \quad (4.1)$$

In the **interaction picture**, a graph state is set initially in the product of  $N$  eigenstates of  $X$ ,  $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ . Since there are no interactions yet, the associated graph is the empty graph of size  $N$ . Then, a graph state  $|\mathbf{0}\rangle_G$  associated to an arbitrary graph  $G$  is created by applying a two-qubit unitary  $U_{u,v}$  for every edge  $(u, v) \in E$ ,

$$|\mathbf{0}\rangle_G = \prod_{(u,v) \in E} U_{u,v} |+\rangle^{\otimes N}. \quad (4.2)$$

The state is completely characterized by the simple, undirected graph  $G$  if these unitaries commute  $[U_{u,v}, U_{v,w}] = 0$ , are symmetric  $U_{u,v} = U_{v,u}$  and all qubits interact through the same unitary (Hein et al., 2006). In this thesis, we deal exclusively with unweighted graph states, for which unitaries  $U_{u,v}$  are the controlled phase gate, or CPHASE:

$$\begin{aligned} U_{u,v} &\equiv |0\rangle_u \langle 0| \otimes I_v + |1\rangle_u \langle 1| \otimes Z_v \\ &= \frac{1}{2} (I_u \otimes I_v + Z_u \otimes I_v + I_u \otimes Z_v - Z_u \otimes Z_v). \end{aligned} \quad (4.3)$$

The subindex in the operators denote on which qubit they operate. In general, by  $O_u$  we denote an operator acting non-trivially on qubit (vertex)  $u$  and as the identity on the rest of the graph. Note that  $U_{u,v}^2 = I \otimes I$ , so applying a CPHASE  $U_{u,v}$  to a graph state with an already existing edge  $(u, v)$  has the effect of deleting this edge. Let us also define here another important two-qubit gate, the controlled-NOT or CNOT, that will be later used. A CNOT with control qubit  $u$  and target qubit  $v$  is defined as

$$\begin{aligned} \text{CNOT}_{u \rightarrow v} &= |0\rangle_u \langle 0| \otimes I_v + |1\rangle_u \langle 1| \otimes X_v \\ &= \frac{1}{2} (I_u \otimes I_v + Z_u \otimes I_v + I_u \otimes X_v - Z_u \otimes X_v). \end{aligned} \quad (4.4)$$

Note that the CNOT is not symmetric. To keep expressions more compact, we will omit in the following the symbol  $\otimes$  when it does not induce confusion.

### 4.1.2 Graph states in the stabilizer formalism

A common, alternative description of graph states is the stabilizer formalism (Nielsen and Chuang, 2000, p. 453). Within this formalism, a stabilizer  $\mathcal{S}$  is a subgroup of the Pauli group on  $N$  qubits whose elements commute and that does not contain the element  $-I$ . A state  $|\psi\rangle$  is said to be **stabilized** by  $\mathcal{S}$  if it is a fixed point under the action of every  $S \in \mathcal{S}$ :

$$S |\psi\rangle = |\psi\rangle \quad \forall S \in \mathcal{S}. \quad (4.5)$$

For a graph state associated to  $G$ , we define  $N$  stabilizer operators  $K_u^G$ , one for each vertex  $u$ , as

$$K_u^G = X_u \prod_{v \in \mathcal{N}_u(G)} Z_v. \quad (4.6)$$

A pure graph state  $|\boldsymbol{\mu}\rangle_G$ , with  $\boldsymbol{\mu} \in \{0,1\}^N$  a binary vector of length  $N$ , is a common eigenstate of all stabilizer operators with

$$K_u^G |\boldsymbol{\mu}\rangle_G = (-1)^{\mu_u} |\boldsymbol{\mu}\rangle_G \quad \forall u \in V, \quad (4.7)$$

where  $\mu_u$  is the  $u$ -th component of  $\boldsymbol{\mu}$ . The set  $\{|\boldsymbol{\mu}\rangle_G\}$  form the graph state basis. Since  $Z_u$  anticommutes with  $K_u$  and commutes with the rest of  $K_v$ ,  $v \neq u$ , any graph-basis element can be expressed as

$$|\boldsymbol{\mu}\rangle_G = \prod_{u \in V} (Z_u)^{\mu_u} |\mathbf{0}\rangle_G. \quad (4.8)$$

In the following, we will omit  $G$  when the graph is clear by context.

The stabilizer formalism is specially interesting because it gives a compact description of graph states. In particular, one does not need to resort to the full group, but just to its generators: a set of elements  $\{g\}$  from which any other element of the group can be obtained by multiplication. This means that one has to check only this generators to see if a given state is stabilized by some  $\mathcal{S}$ . The fact that there are at most  $\log |\mathcal{S}|$  generators makes this formalism very efficient. In addition, one can see that the effect of unitaries  $U$  on a state  $|\psi\rangle$  can be tracked by the transformation of the generator  $UgU^\dagger$  (Nielsen and Chuang, 2000).

### 4.1.3 Mixed graph states

Due to decoherence, a graph state will be in general in a mixed state. This can be understood as a pure graph state that undergoes a certain noise process, modeled as the action of Pauli matrices acting on both sides of the pure state  $|\mathbf{0}\rangle_G \langle \mathbf{0}|$ . This leaves the graph state in a general form  $\rho_G$ , with a certain fidelity

$$F = \langle \mathbf{0} | \rho_G | \mathbf{0} \rangle_G \quad (4.9)$$

with the original pure state. As  $\{|\boldsymbol{\mu}\rangle\}$  form a basis,  $\rho_G$  can be expressed as

$$\rho_G = \sum_{\boldsymbol{\mu}, \boldsymbol{\nu}} \lambda_{\boldsymbol{\mu}, \boldsymbol{\nu}} |\boldsymbol{\mu}\rangle \langle \boldsymbol{\nu}|, \quad (4.10)$$

where the sum is over all possible  $\boldsymbol{\mu}$  and  $\boldsymbol{\nu}$ . A state like this can be depolarized into a diagonal form by randomly applying local Pauli operations (Aschauer et al., 2005; Dür et al., 2005c). The random local Pauli operations correspond to the stabilizer operators  $K_u$ . Using Eq. 4.6, one sees that the action of  $K_u$  on  $|\boldsymbol{\mu}\rangle \langle \boldsymbol{\nu}|$  is

$$K_u |\boldsymbol{\mu}\rangle \langle \boldsymbol{\nu}| K_u = (-1)^{\mu_u + \nu_u} |\boldsymbol{\mu}\rangle \langle \boldsymbol{\nu}|. \quad (4.11)$$

Hence, applying  $K_u$  with probability 1/2 and otherwise leaving  $\rho_G$  untouched, one gets rid of the elements  $\lambda_{\boldsymbol{\mu}, \boldsymbol{\nu}}$  with  $\mu_u \neq \nu_u$ . To fully depolarize  $\rho_G$ , this has to be done for all  $K_u$ . Note also that the diagonal elements are left unchanged, and in

particular the fidelity, equal to  $\lambda_{\mathbf{0},\mathbf{0}}$ , remains the same. In the remaining of the Chapter we use  $\lambda_{\boldsymbol{\mu}} \equiv \lambda_{\boldsymbol{\mu},\boldsymbol{\mu}}$  and consider mixed graph states diagonal in the graph state basis:

$$\rho_G = \sum_{\boldsymbol{\mu}} \lambda_{\boldsymbol{\mu}} |\boldsymbol{\mu}\rangle \langle \boldsymbol{\mu}|. \quad (4.12)$$

Instead of using the standard graph state basis, for our purpose, it will more convenient to work in the operator basis. Let us define the stabilizer operator  $\mathbf{K}_x^G$ :

$$\mathbf{K}_x^G = \prod_{u \in V} (K_u^G)^{x_u}. \quad (4.13)$$

Here  $\mathbf{x}$  is a binary vector of  $N$  elements (either 0 or 1),  $\mathbf{x} = (x_1 x_2 \dots x_N) \in \{0, 1\}^N$ , that indicates which stabilizer is included in the product  $\mathbf{K}_x^G$ . The  $2^N$  different products of stabilizers  $\mathbf{K}_x^G$  are a subset of the  $N$ -qubit Pauli operators (which are  $4^N$ ), and are sufficient to generate graph diagonal states. Using Eqs. 4.7 and 4.12, a state in this basis takes the form

$$\rho_G = \frac{1}{2^N} \sum_{\mathbf{x}} \langle \mathbf{K}_x^G \rangle \mathbf{K}_x^G, \quad (4.14)$$

where  $\langle \mathbf{K}_x^G \rangle = \sum_{\boldsymbol{\mu}} \lambda_{\boldsymbol{\mu}} (-1)^{\boldsymbol{\mu} \cdot \mathbf{x}}$  and, similarly as before, the sum is over all possible  $\mathbf{x}$ . The fidelity of a mixed graph state  $\rho_G$  with respect to  $|\mathbf{0}\rangle_G$  is thus the sum of these expected values:

$$F_G = {}_G \langle \mathbf{0} | \rho_G | \mathbf{0} \rangle_G = \lambda_{\mathbf{0}} = \frac{1}{2^N} \sum_{\mathbf{x}} \langle \mathbf{K}_x^G \rangle. \quad (4.15)$$

#### 4.1.4 Operations and measurements on graph states

Using their commutation relations, the effect of the Pauli operators on  $\mathbf{K}_x$  is

$$Z_u \mathbf{K}_x Z_u = (-1)^{x_u} \mathbf{K}_x, \quad (4.16)$$

$$X_u \mathbf{K}_x X_u = \prod_{v \in \mathcal{N}_u} (-1)^{x_v} \mathbf{K}_x, \quad (4.17)$$

$$Y_u \mathbf{K}_x Y_u = (-1)^{x_u} \prod_{v \in \mathcal{N}_u} (-1)^{x_v} \mathbf{K}_x, \quad (4.18)$$

so these unitaries map diagonal graph states into diagonal graph states. A CPHASE  $U_{u,v}$  adds an edge between  $u$  and  $v$ , if they were not connected, or removes it, if the edge already existed. The effect on  $\mathbf{K}_x$  is

$$U_{u,v} \mathbf{K}_x U_{u,v} = \mathbf{K}_x (Z_v)^{x_u} (Z_u)^{x_v}. \quad (4.19)$$

The action of Pauli measurements can also be easily described in this formalism as a transformation of the graph (up to some local unitaries). Measurement of  $Z$

simply disconnects the measured qubit from the rest of the graph, while  $X$  and  $Y$  transform the neighborhood of the measured qubit and then disconnect it. In terms of the stabilizer operators, the measurement of  $Z_u$  commutes with all  $K_v$ , for  $v \neq u$ , and anticommutes with  $K_u$ . Thus,

$$[I + (-1)^m Z_u] \mathbf{K}_x [I + (-1)^m Z_u] = [I + (-1)^m Z_u] \mathbf{K}_x \delta_{0,x_u}, \quad (4.20)$$

where  $m = \{0, 1\}$  labels the measurement outcome  $\{+1, -1\}$  respectively. After tracing out qubit  $u$ , the new stabilizer is

$$(-1)^{m \cdot \sum_{v \in \mathcal{N}_u} x_v} \mathbf{K}'_x \delta_{0,x_u}, \quad (4.21)$$

where the new  $\mathbf{K}'_x$  corresponds to a new graph  $G'$  obtained from  $G$  by removing vertex  $u$  and its attached edges. The spurious phase factor in Eq. 4.21 can be cancelled by applying a unitary  $(\prod_{v \in \mathcal{N}_u} Z_v)^m$ . Similarly, measurements of  $X_u$  or  $Y_u$  also result in the disconnection of the measured qubit, but in these cases the remaining graph is transformed by local complementations of the neighborhood of  $u$ , as described in (Hein et al., 2006). We will be more explicit in the concrete cases where we apply these measurements.

## 4.2 Network and noise model

We consider a network where nodes are spatially separated. Some of the nodes are connected through quantum links, which we model as a noisy depolarizing channel on one qubit with error parameter  $p_c$ ,

$$T_c^{(u)} = (1 - p_c)[I] + \frac{p_c}{4} \sum_{i=0}^3 [\sigma_i^{(u)}]. \quad (4.22)$$

The square brackets  $[A]$  denote that  $A$  acts on both sides of the state, as  $[A]\rho = A\rho A^\dagger$ , and  $\sigma_i$  are  $I, X, Y, Z$  for  $i = 0, 1, 2, 3$ , respectively. A Pauli measurement on  $u$  is modeled as a perfect measurement preceded by a depolarizing channel  $T_1^{(u)}$  with error probability  $p_1$  on that qubit,

$$T_1^{(u)} = (1 - p_1)[I] + \frac{p_1}{4} \sum_{i=0}^3 [\sigma_i^{(u)}]. \quad (4.23)$$

A noisy two-qubit gate (eg a CPHASE or a CNOT) on qubits  $u$  and  $v$  is modeled as an ideal gate followed by the two-qubit depolarizing channel on  $u, v$  with error parameter  $p_2$ ,

$$T_2^{(u,v)} = (1 - p_2)[I] + \frac{p_2}{16} \sum_{i,j=0}^3 [\sigma_i^{(u)} \otimes \sigma_j^{(v)}] \quad (4.24)$$

Gates can only be applied locally, ie on qubits within the same node.

In the stabilizer basis, the effect of each noise source is easily tracked: it multiplies each stabilizer element by a coefficient, and keeps the graph state in diagonal form. Let us consider for example the effect of noise  $T_2^{(u,v)}$  on  $\mathbf{K}_x$ . The first part of Eq. 4.24, that proportional to  $1 - p_2$ , does not affect  $\mathbf{K}_x$ . Each term in the second part, that proportional to  $p_2$ , will at most induce a change of sign. Using Eqs. 4.16 to 4.18, one easily finds that the sign change can only occur if  $\mathbf{K}_x$  acts non-trivially on  $u$  or  $v$ , ie if exists at least one  $x_a = 1$  for  $a \in u \cup v \cup \mathcal{N}_u \cup \mathcal{N}_v$ . In addition one sees that half of the 16 terms in the sum induce a sign change which cancel the contribution of the other half. Hence, the noise  $T_2^{(u,v)}$  does not alter  $\mathbf{K}_x$  unless  $x$  has support in  $u$  or  $v$ , in which case it gets multiplied by a factor  $(1 - p_2)$ :

$$T_2^{(u,v)}(\mathbf{K}_x) = (1 - p_2)^{\theta(x_u, x_v, x_{\mathcal{N}_u}, x_{\mathcal{N}_v})} \mathbf{K}_x \quad (4.25)$$

where  $\theta(x) \equiv 1 - \delta(\mathbf{0}, x)$  and  $x_{\mathcal{N}_u} = (x_{v_1} \dots x_{v_k})$  for all  $v_a \in \mathcal{N}_u$ . Noise  $T_1^{(u)}$  behaves similarly. In this case, the sign is changed with probability  $p_1/2$  unless  $x_u = 0$  and  $\bigoplus_a x_a = 0$  for  $a \in \mathcal{N}_u$ , so the multiplying coefficient is

$$(1 - p_1)^{\theta(x_u, \bigoplus_{v \in \mathcal{N}_u} x_v)}. \quad (4.26)$$

Note thus that these noise sources affect the qubits on which the gates act, plus their neighbors.

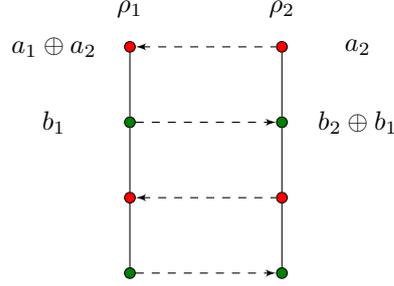
Finally let us point that we do not associate any noise to the local ‘‘correcting’’ unitaries performed in order to bring the post-measurement states to a standard graph form. We refer to these unitaries in our protocols to simplify bookkeeping, but their action can be pushed forward (or commuted) till the end of the protocol, and hence the resulting state is exactly equivalent as a resource of entanglement.

### 4.3 Multipartite purification protocol

The recursive protocol of Dür et al. (2003) (see also Aschauer et al., 2005) is an entanglement purification protocol for multipartite, two-colorable graph states, which has also been extended to general graph states (Kruszynska et al., 2006). It operates on two identical copies

$$\rho_1 \otimes \rho_2 = \frac{1}{2^{2N}} \sum_{x_1, x_2} \langle \mathbf{K}_{x_1}^{(1)} \rangle \langle \mathbf{K}_{x_2}^{(2)} \rangle \mathbf{K}_{x_1}^{(1)} \mathbf{K}_{x_2}^{(2)} \quad (4.27)$$

of a two-colorable graph state of size  $N$  (with colors  $A$  and  $B$ ), and consists of two subprotocols (P1 and P2), each of which purifies one of the two colors. Here  $\mathbf{K}_{x_i}^{(i)}$  is a stabilizer element of state  $\rho_i$ . In each subprotocol, information about  $\rho_1$  is transferred to  $\rho_2$ . Then,  $\rho_1$  is kept or discarded depending on the outcomes of measurements on  $\rho_2$ .



**Figure 4.2.** Subprotocol P1. In this example, two copies  $\rho_1$  and  $\rho_2$  of a linear cluster state of four qubits. Qubits in color  $A$  and  $B$  are colored as red and green, respectively. A dashed arrow from qubit  $u$  to  $v$  corresponds to a  $\text{CNOT}_{u \rightarrow v}$  ( $u$  is the control qubit). The effect in the indices of the stabilizer elements is shown. After the CNOTs, nodes in  $\rho_2$  are measured in  $X$  (those belonging to  $A$ ) and in  $Z$  (those in  $B$ ).

Let us briefly review subprotocol P1, depicted in Figure 4.2, in the basis  $\mathbf{K}_x$ . In this subprotocol, a  $\text{CNOT}_{2 \rightarrow 1}$  is applied to every node in  $A$ , and a  $\text{CNOT}_{1 \rightarrow 2}$  to every node in  $B$ . This transforms the state to

$$\frac{1}{2^{2N}} \sum_{x_1, x_2} \langle \mathbf{K}_{x_1}^{(1)} \rangle \langle \mathbf{K}_{x_2}^{(2)} \rangle \mathbf{K}_{a_1 \oplus a_2}^{(A_1)} \mathbf{K}_{b_1}^{(B_1)} \mathbf{K}_{a_2}^{(A_2)} \mathbf{K}_{b_2 \oplus b_1}^{(B_2)}, \quad (4.28)$$

where  $\mathbf{a}$  and  $\mathbf{b}$  are the elements of  $\mathbf{x}$  that correspond to colors  $A$  and  $B$ . The stabilizer  $\mathbf{K}_{a_1 \oplus a_2}^{(A_1)}$  corresponds to qubits of  $\rho_1$  in color  $A$ , and the modulo 2 summation of the index  $\mathbf{a}_1 \oplus \mathbf{a}_2$  is made elementwise. To keep notation short, we contract  $\mathbf{K}_{a_1 \oplus a_2}^{(A_1)} \mathbf{K}_{b_2}^{(B_1)} = \mathbf{K}_{a_1 \oplus a_2, b_2}^{(1)}$ , where the first subindex corresponds to color  $A$  and the second to color  $B$ . Then, every node in  $\rho_2$  is measured:  $X$  in nodes of color  $A$  (outcomes  $\xi$ ) and  $Z$  in nodes of color  $B$  (outcomes  $\zeta$ ). This gives the (unnormalized) state

$$\begin{aligned} & \sum_{x_1, x_2} (-1)^{(\xi \oplus \zeta) \cdot \mathbf{a}_2} \langle \mathbf{K}_{x_1}^{(1)} \rangle \langle \mathbf{K}_{x_2}^{(2)} \rangle \mathbf{K}_{a_1 \oplus a_2, b_1}^{(1)} \delta_{\mathbf{0}, b_2 \oplus b_1} \\ &= \sum_{x_1, a_2} (-1)^{(\xi \oplus \zeta) \cdot \mathbf{a}_2} \langle \mathbf{K}_{a_1, b_1}^{(1)} \rangle \langle \mathbf{K}_{a_2, b_1}^{(2)} \rangle \mathbf{K}_{a_1 \oplus a_2, b_1}^{(1)}, \end{aligned} \quad (4.29)$$

where  $\xi \oplus \zeta$  is a binary vector with components  $\xi_u \oplus \zeta_v$  for all  $u \in A$ . The state is selected if  $\xi \oplus \zeta = \mathbf{0}$ . Summing over all the possible outcomes, the final (post-selected) state after P1 is

$$\frac{1}{2^{N|A|}} \sum_{x_1, a_2} \langle \mathbf{K}_{a_1}^{(1)} \mathbf{K}_{b_1}^{(1)} \rangle_1 \langle \mathbf{K}_{a_2}^{(2)} \mathbf{K}_{b_1}^{(2)} \rangle_2 \mathbf{K}_{a_1 \oplus a_2}^{(A_1)} \mathbf{K}_{b_1}^{(B_1)}. \quad (4.30)$$

Subprotocol P2 is equivalent, with colors  $A$  and  $B$  interchanged. The purification protocol is the successive application of subprotocols P1 and P2. It can be seen that this map has a fixed point that has a larger fidelity than the unpurified states.

For our protocol, we need the fixed point of a GHZ of size  $j + 1$ , with a central node colored as  $A$  and  $j$  leaves colored as  $B$  (ie  $|A| = 1$  and  $|B| = j$ ). Noise can come from CNOTs as  $1 - p_2$  and from measurements in state  $\rho_2$  as  $1 - p_1$ :

$$\begin{aligned} \langle K_a \mathbf{K}_b \rangle^{(P1)} &= \frac{1}{2} \sum_{a_2=0}^1 \langle K_{a \oplus a_2}^{(1)} \mathbf{K}_b^{(1)} \rangle \langle K_{a_2}^{(2)} \mathbf{K}_b^{(2)} \rangle \\ &\quad (1 - p_2)^{\theta(a, a_2, \bigoplus_{b \in B} b)} \prod_{b \in B} (1 - p_2)^{\theta(a, a_2, b)} \\ &\quad (1 - p_1)^{a_2} \prod_{b \in B} (1 - p_1)^{a_2} \end{aligned} \quad (4.31a)$$

$$\begin{aligned} \langle K_a \mathbf{K}_b \rangle^{(P2)} &= \frac{1}{2^d} \sum_{b_2=\mathbf{0}}^1 \langle K_a^{(1)} \mathbf{K}_{b \oplus b_2}^{(1)} \rangle \langle K_a^{(2)} \mathbf{K}_{b_2}^{(2)} \rangle \\ &\quad (1 - p_2)^{\theta(a, \bigoplus_{b \in B} b_2, \bigoplus_{b \in B} b)} \prod_{b \in B} (1 - p_2)^{\theta(a, b_2, b)} \\ &\quad (1 - p_1)^{\bigoplus_{b \in B} b_2} \prod_{b \in B} (1 - p_1)^{b_2}. \end{aligned} \quad (4.31b)$$

For simplicity (see also the discussion in Section 4.5), we take a single error parameter  $p = p_1 = p_2$  and approximate  $\langle K_a \mathbf{K}_b \rangle$  at first order in  $p$ . Let the unnormalized  $\langle K_a \mathbf{K}_b \rangle^{(P1)} \sim 1 - \tilde{\beta}_{a,|b|} p$  and  $\langle K_a \mathbf{K}_b \rangle^{(P2)} \sim 1 - \tilde{\alpha}_{a,|b|} p$ . Composing P1 and P2 we can find the fixed point at first order in  $p$ . In P1, each  $\tilde{\beta}_{a,|b|}$  equals  $\tilde{\alpha}_{0,|b|} + \tilde{\alpha}_{1,|b|}$  plus a constant term:

$$\tilde{\beta}_{0,|b|} = \alpha_{0,|b|} + \alpha_{1,|b|} + j + 1 + \left\lceil \frac{|b|}{2} \right\rceil, \quad (4.32a)$$

$$\tilde{\beta}_{1,|b|} = \alpha_{0,|b|} + \alpha_{1,|b|} + \frac{3}{2}(j + 1). \quad (4.32b)$$

These terms are normalized dividing them by  $\langle \mathbb{1} \rangle = \langle K_0 \mathbf{K}_0 \rangle$ , so the normalized first-order coefficients for P1 read  $\beta_{a,|b|} = \tilde{\beta}_{a,|b|} - \tilde{\beta}_{0,0}$ . Similarly, for P2:

$$\tilde{\alpha}_{0,|b|} = \frac{1}{2^j} \sum_{b_2} (\beta_{0, b \oplus b_2} + \beta_{0, b_2}) + j + 1 + \left\lceil \frac{|b|}{2} \right\rceil \quad (4.33a)$$

$$\tilde{\alpha}_{1,|b|} = \frac{1}{2^j} \sum_{b_2} (\beta_{1, b \oplus b_2} + \beta_{1, b_2}) + \frac{3}{2}(j + 1) \quad (4.33b)$$

After normalization (dividing by  $\langle K_0 \mathbf{K}_0 \rangle$ ),

$$\langle K_0 \mathbf{K}_b \rangle \sim 1 - \left\lceil \frac{|b|}{2} \right\rceil p, \quad (4.34a)$$

$$\langle K_1 \mathbf{K}_b \rangle \sim 1 - (j + 1)p. \quad (4.34b)$$

The fidelity is

$$F \sim 1 - \frac{1}{2^{j+1}} \sum_{b=0}^j \binom{j}{b} \left[ \left\lceil \frac{b}{2} \right\rceil + (j+1) \right] p = 1 - \frac{5}{8}(j+1)p. \quad (4.35)$$

This fixed point is only reached if the initial states have a fidelity above a certain minimum fidelity that depends on the size of the state and on the noise of operations  $p$  (Aschauer et al., 2005). There is also a threshold in this noise, above which the purification protocol cannot increase the fidelity. In linear cluster states, this threshold is independent of the size of the state. However, in the purification of GHZ states the threshold decreases with  $N$ . This is because the number of errors that can affect a given node depends on its degree, and in GHZ the central node has a scaling degree  $N - 1$ . Typically, the error threshold depends on the maximum degree of the graph state. Here we will assume that the GHZ subgraphs are small enough so as to guarantee that the noise of the operations falls below this threshold. Also, we consider that the channel noise is within the working parameters of the purification protocol. Although the noise parameters of the channel do not enter explicitly in the final fidelity, it is their presence that imposes the use of purification which in turn introduces noise.

## 4.4 Protocols

We propose a protocol to distribute a graph state with the structure of a general quantum communication network, of arbitrary topology, associated to graph  $G$ . Noise  $p_c$  in the communication channels is considered to be relatively high, so some sort of purification or error correction is in order. The protocol uses the structure of the network to distribute several copies of small subgraphs between neighbors, and purifies them by means of multipartite purification. These subgraphs are GHZ states, which are associated to a star graph with a central node of degree  $j$ , connected to  $j$  leaves of degree 1. In order to benchmark our protocol, we also consider two reference protocols that distribute bipartite states between a central node, which locally creates the desired graph state, and the rest of the network. The bipartite states are then used to teleport the locally created graph state. Since this central node may not be directly connected to the rest of the network, quantum repeaters (Briegel et al., 1998) are used to establish purified bipartite states between this node and all the network's nodes.

In all cases, we consider the same multipartite purification protocol for 2-colorable graph states (Aschauer et al., 2005) described in the previous section. As we discussed, the noise threshold of this purification scheme depends on the degree of the central qubit of the state that is purified, and goes from  $\approx 0.06$  in a bipartite state (ie a GHZ with central degree 1) to  $\approx 0.02$  for a GHZ of central degree 9. This poses a limit in the maximum degree of a network for which the protocols can be used. Hence, we work under the assumption that the errors in

the local operations are low—as compared to the (finite) degree of the subgraphs,  $jp \ll 1$ —and that the channel noise can be significant, but also low enough so that the minimum threshold fidelity required for the purification protocol to succeed can be attained. As we have seen in Section 4.3, at first order in  $p = p_1 = p_2$ , the output state fixed point of this purification scheme is given by

$$\left\langle K_a^{x_a} \prod_{b \in \mathcal{N}_a} K_b^{x_b} \right\rangle = 1 - p \left[ \overline{x_a} \left\lceil \frac{\sum_b x_b}{2} \right\rceil + x_a(j+1) \right], \quad (4.36)$$

where  $a$  is the central node and  $b$  the leaves of the GHZ,  $j$  is the number of leaves and the overline in  $x_a$  represents the bit-complement,  $\overline{x_a} = x_a \oplus 1$ .

Note that the extensive use of quantum repeaters in the first two protocols renders them extremely inefficient. Nevertheless, we will find that the performance in terms of attainable fidelity is still comparable to that of our more efficient subgraph protocol.

#### 4.4.1 Bipartite A protocol

In the first protocol (depicted in Figure 4.3a), purified, but still noisy, entangled states are created between a central node and all the others by means of quantum repeaters. This central node then teleports a locally generated state to all the other nodes. For simplicity, in protocols *A* and *B* we assume that the central node is sending *all* qubits through purified channels, including his own. This adds a source of noise that would not be strictly necessary, but its effect is small for large enough networks. In cases, like complex networks, where the network is not necessarily connected, it is understood that there is a central node for each connected component distributing the corresponding graph state.

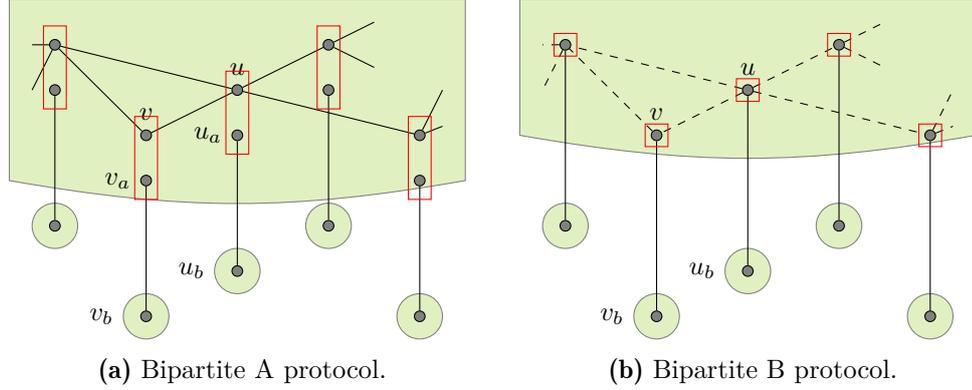
The local state  $\rho_G$ , which mimics the structure of the network, is created by initializing  $N$  qubits in the state stabilized by  $X$  and then applying CPHASES between neighbors. Then, the noisy Bell states between the central node and all the others are used to teleport the corresponding qubits (see Figure 4.3a). We label qubits in the local graph as  $u \in V(G)$ . For each node in the network, there is a bipartite state  $\rho_{g_u}$  of two qubits,  $u_a, u_b \in V(g_u)$  and one edge  $(u_a, u_b) \in E(g_u)$ . Qubit  $u_a$  belongs to the central node, while  $u_b$  is in the corresponding node in the network. In order to account for the errors in the teleportation Bell-measurement, we implement it by a CPHASE on  $(u, u_a)$  followed by  $X$  measurements on  $u$  and  $u_a$ .

To simplify the explanation, we will start analysing the generation of the graph state in a noiseless scenario. The effect of noise will be accounted for latter. The state before teleportation is

$$\rho_G \otimes \bigotimes_{u \in V(G)} \rho_{g_u}, \quad (4.37)$$

with

$$\rho_G = \frac{1}{2^N} \sum_x \mathbf{K}_x^G \quad (4.38a)$$



**Figure 4.3.** Bipartite A and B protocols. The upper green area represents the “central” node, where the local graph state with the structure of the network is generated. The lower green circles are nodes in the network. Small, gray dots correspond to qubits, and lines connect neighbors. (a) Rectangles in red indicate Bell measurements involved in the teleportation of the local graph states. (b) Dashed lines, corresponding to the edges of the local graph, are created after solid lines, corresponding to the distributed bipartite states. Squares in red indicate  $X$  measurements used to teleport the local graph states.

and

$$\rho_{g_u} = \frac{1}{4} \sum_{x_{u_a}, x_{u_b}} (K_{u_a}^{g_u})^{x_{u_a}} (K_{u_b}^{g_u})^{x_{u_b}}. \quad (4.38b)$$

Graphs  $g_u$  correspond to the bipartite states with qubits  $u_a$  and  $u_b$  used for teleportation, ie  $K_{u_a}^{g_u} = X_{u_a} Z_{u_b}$ . The action of the CPHASE  $U_{u, u_a}$  affects only the stabilizers  $K_u^G \rightarrow K_u^G Z_{u_a}$  and  $K_{u_a}^{g_u} \rightarrow K_{u_a}^{g_u} Z_u$ . The measurement of  $X_u$  anticommutes with all  $K_v^G$ ,  $v \in \mathcal{N}_u(G)$ , and with  $K_{u_a}^{g_u} Z_u$ , while that of  $X_{u_a}$  anticommutes only with  $K_u^G Z_{u_a}$  and  $K_{u_b}^{g_u}$ . Thus, each term changes to

$$[1 + (-1)^{m_{u_a}} X_{u_a}] [1 + (-1)^{m_u} X_u] \mathbf{K}_x (Z_{u_a})^{x_u} (K_{u_a}^{g_u} Z_u)^{x_{u_a}} (K_{u_b}^{g_u})^{x_{u_b}} \times \delta_{0, x_{u_a} \oplus \bigoplus_{v \in \mathcal{N}_u} x_v} \delta_{0, x_u \oplus x_{u_b}}, \quad (4.39)$$

where  $m_u$  and  $m_{u_a}$  are the measurement outcomes. Tracing out qubits  $u$  and  $u_a$  and correcting the state with  $(Z_{u_b})^{m_u} (X_{u_b})^{m_{u_a}}$  we obtain

$$\mathbf{K}'_x \delta_{0, x_{u_a} \oplus \bigoplus_{v \in \mathcal{N}_u} x_v} \delta_{0, x_u \oplus x_{u_b}}, \quad (4.40)$$

with  $\mathbf{K}'_x$  associated to a new graph  $G'$  where vertex  $u$  has been substituted by vertex  $u_b$ . Teleportation of all local qubits results in the desired distributed state.

Noise can now be introduced as the multiplicative factors in front of the stabilizer elements of Eq. 4.37. Here, the order of the CPHASE gates used to generate

the local graph matters, as qubits receive noise from gates performed at neighbors which are already connected to them. There is thus a noise  $(1 - p_1)^{x_u}$  corresponding to the preparation of each node in  $X_u$ , and a  $(1 - p_2)^{\theta(x_u, x_v, x_{\tilde{\mathcal{N}}_u}, x_{\tilde{\mathcal{N}}_v})}$  for each edge in the local state, where the tilde in  $\tilde{\mathcal{N}}_u$  labels that we consider the neighborhood of  $u$  at the moment the  $\text{CPHASE}_{u,v}$  is performed. Additionally, and using that  $x_{u_a} = \bigoplus_{v \in \mathcal{N}_u} x_v$  and  $x_{u_b} = x_u$ , in each teleportation the  $\text{CPHASE}$  introduces noise as  $(1 - p_2)^{\theta(x_u, x_{\mathcal{N}_u})}$ , and the measurements as  $(1 - p_1)^{x_u + \bigoplus_{v \in \mathcal{N}_u} x_v}$ . Finally, there is a  $\langle (K_{u_a}^{g_u})^{\bigoplus_{v \in \mathcal{N}_u} x_v} (K_{u_b}^{g_u})^{x_u} \rangle$  term from the purified Bell states. This results in a final distributed state  $\rho = \frac{1}{2^N} \sum_{\mathbf{x}} \langle \mathbf{K}_{\mathbf{x}} \rangle \mathbf{K}_{\mathbf{x}}$  with

$$\begin{aligned} \langle \mathbf{K}_{\mathbf{x}} \rangle &= \prod_{u \in V} \left[ (1 - p_1)^{x_u} \left\langle (K_{u_a}^{g_u})^{\bigoplus_{v \in \mathcal{N}_u} x_v} (K_{u_b}^{g_u})^{x_u} \right\rangle \right. \\ &\quad \left. (1 - p_2)^{\theta(x_u, x_{\mathcal{N}_u})} (1 - p_1)^{x_u + \bigoplus_{v \in \mathcal{N}_u} x_v} \right] \\ &\quad \prod_{(u,v) \in E} (1 - p_2)^{\theta(x_u, x_v, x_{\tilde{\mathcal{N}}_u}, x_{\tilde{\mathcal{N}}_v})}, \end{aligned} \quad (4.41)$$

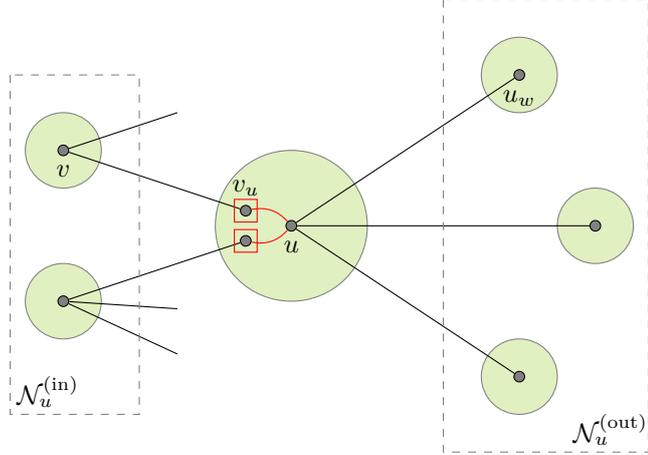
where the tilde over  $\Pi$  denotes that the edges are introduced in a certain order. An explicit expression for the Bell state correlators for the postselection purification protocol at first order in  $p$  can be obtained from Eq. 4.36:

$$\langle (K_{u_a}^{g_u})^{x_{u_a}} (K_{u_b}^{g_u})^{x_{u_b}} \rangle \sim 1 - (\bar{x}_{u_a} x_{u_b} + 2x_{u_a})p. \quad (4.42)$$

#### 4.4.2 Bipartite B protocol

The previous strategy can be improved by directly connecting the local ends of Bell pairs by means of  $\text{CPHASE}$ s, and then teleporting the local graph state performing only one  $X$  measurement per node—instead of the various sources of noise induced by the Bell-measurement ( $\text{CPHASE}$  and two  $X$  measurements). The initial state is now  $\bigotimes_{u \in V(G)} \rho_{g_u}$ , where qubits are labeled as  $u$  if they belong to the central node and  $u_b$  if they are in the distributed nodes (see Figure 4.3b). The sources of noise are the non-unit purity of Bell pairs,  $\text{CPHASE}$ s used in the preparation of the local state and the measurement involved in each teleportation. As in the previous case, the order of  $\text{CPHASE}$ s is important: each contributes to a noise with  $(1 - p_2)^{\theta(x_u, x_v, x_{u_b}, x_{v_b}, x_{\tilde{\mathcal{N}}_u}, x_{\tilde{\mathcal{N}}_v})}$ . Note that the  $\text{CPHASE}$ s are performed on qubits that are already connected to nodes in the network and thus affect the distributed qubits  $u_b$ . The calculation is similar as before and results in a final distributed state with correlators

$$\begin{aligned} \langle \mathbf{K}_{\mathbf{x}} \rangle &= \prod_{u \in V} \left[ \left\langle (K_u^{g_u})^{x_u} (K_{u_b}^{g_u})^{\bigoplus_{v \in \mathcal{N}_u} x_v} \right\rangle (1 - p_1)^{x_u} \right] \\ &\quad \prod_{(u,v) \in E} (1 - p_2)^{\theta(x_u, x_v, \bigoplus_{w \in \mathcal{N}_u} x_w, \bigoplus_{w \in \mathcal{N}_v} x_w, x_{\tilde{\mathcal{N}}_u}, x_{\tilde{\mathcal{N}}_v})}. \end{aligned} \quad (4.43)$$



**Figure 4.4.** Purify subgraphs and merge. Green circles are nodes in the network. Nodes on the right are the outgoing neighborhood of  $u$ , while those on the left are the incoming neighborhood. Small, gray dots correspond to qubits, and lines connect neighbors. Lines and squares in red indicate CPHASES and measurements involved in the connection of subgraphs.

### 4.4.3 Purify subgraph and merge

In this strategy, subgraph states of small size ( $N$  independent) are distributed and purified, and then interconnected locally at each node to form the desired structure. The protocol follows two steps (see Figure 4.4). To each node of degree  $k$  we assign an outgoing neighborhood, with  $j$  nodes, and an incoming neighborhood, with  $i$  nodes, so  $k = j + i$ . First, each node prepares a GHZ of size  $j + 1$ . A GHZ graph state with  $j + 1$  qubits is associated to a star graph  $g_u$ , ie a graph with a central node and  $j$  external nodes, called leaves. Then, each qubit corresponding to a leaf of the GHZ is sent to one of the outgoing neighboring nodes through the depolarizing channels. Several copies of this distributed state are created and then purified using the bicolorable graph state protocol of Dür et al. (2003). The final purified subgraph state with central qubit  $u$  is

$$\rho_u = \frac{1}{2^j} \sum_{x_u, \mathbf{x}_{\mathcal{N}_u^{(out)}}} \left\langle (K_u^{g_u})^{x_u} \prod_{u_w \in \mathcal{N}_u^{(out)}} (K_{u_w}^{g_u})^{x_{u_w}} \right\rangle (K_u^{g_u})^{x_u} \prod_{u_w \in \mathcal{N}_u^{(out)}} (K_{u_w}^{g_u})^{x_{u_w}}, \quad (4.44)$$

where the correlators of the fixed point of the purification scheme are given in Eq. 4.36. Here  $u_w$  denotes the leaf qubit that has been sent to  $w \in \mathcal{N}_u^{(out)}(G)$ . At the same time, each node receives  $i = k - j$  leaves corresponding to the GHZ states created at its incoming neighborhood.

In the second step, each node connects the central qubits of their GHZ state with the leaves they have received. The connection is made by performing a CPHASE between the two qubits  $u$  and  $v_u$  and a  $Y$  measurement of the received

qubit  $v_u$ . The effect of the  $Y$  measurement is to add an edge between the two central nodes  $u$  and  $v$  and destroy the measured qubit  $v_u$ .

The action on the stabilizers can be seen in a single connection example between a central qubit  $u$  of a subgraph and a leaf qubit  $v_u$  of an incoming subgraph (see Figure 4.4). We focus on the stabilizer elements  $(K_v^{g_v})^{x_v} (K_{v_u}^{g_{v_u}})^{x_{v_u}} (K_u^{g_u})^{x_u}$ , as all the others remain unchanged. The action of a CPHASE between  $u$  and  $v_u$  is  $K_{v_u}^{g_{v_u}} \rightarrow K_{v_u}^{g_{v_u}} Z_u$  and  $K_u^{g_u} \rightarrow K_u^{g_u} Z_{v_u}$ . The measurement of  $Y_{v_u}$  anticommutes with the new stabilizer operators at  $v$ ,  $v_u$ , and  $u$ , so each term changes to

$$[1 + (-1)^m Y_{v_u}] (K_v^{g_v})^{x_v} (K_{v_u}^{g_{v_u}} Z_u)^{x_{v_u}} (K_u^{g_u} Z_{v_u})^{x_u} \delta_{0, x_v \oplus x_{v_u} \oplus x_u}. \quad (4.45)$$

Tracing out  $v_u$  and correcting the state depending on the measurement outcome  $m$  with  $\exp[(-1)^m i \frac{\pi}{4} Z_v] \exp[(-1)^m i \frac{\pi}{4} Z_u]$ , we obtain

$$(K'_v)^{x_v} (K_u^{g_u} Z_v)^{x_u} \delta_{0, x_v \oplus x_{v_u} \oplus x_u}, \quad (4.46)$$

where  $K'_v$  is the stabilizer of  $v$  with qubit  $v_u$  changed to  $u$ .

Noise added by the measurement enters as a coefficient  $(1 - p_1)^{x_u \oplus x_v}$ , where we used Eq. 4.26 together with  $x_{v_u} = x_u \oplus x_v$ . Noise by a CPHASE between one of the incoming leaves  $v_u$  and  $u$  adds

$$(1 - p_2)^{\theta(x_u, x_v, \mathbf{x}_{\mathcal{N}_u^{(\text{out})}}, \mathbf{x}_{\tilde{\mathcal{N}}_u^{(\text{in})}})}, \quad (4.47)$$

where  $\tilde{\mathcal{N}}_u^{(\text{in})}$  is the incoming neighborhood that has already been connected to  $u$ .

Finally, note that if a node has  $j = 0$  it does not need to prepare any GHZ state. In this case, one of the incoming leaves is used as the qubit to which all the other leaves are connected. If there are no incoming leaves, then the node is isolated (and is thus prepared in the state stabilized by  $X$ ). This means that the contribution of the GHZ distribution and purification at a node  $u$  is

$$\langle \mathbf{K}_x^{g_u} \rangle \begin{cases} \left\langle (K_u^{g_u})^{x_u} \prod_{w \in \mathcal{N}_u^{(\text{out})}} (K_{u_w}^{g_{u_w}})^{x_u \oplus x_w} \right\rangle & \text{if } j > 0, \end{cases} \quad (4.48a)$$

$$\begin{cases} 1 & \text{if } j = 0, i > 0, \end{cases} \quad (4.48b)$$

$$\begin{cases} (1 - p_1)^{x_u} & \text{if } j = i = 0. \end{cases} \quad (4.48c)$$

All in all, the final correlators for the subgraph protocol can be written as,

$$\langle \mathbf{K}_x \rangle = \prod_{u \in V} \left\{ \langle \mathbf{K}_x^{g_u} \rangle \prod_{v_a \in \mathcal{N}'_u} \left[ (1 - p_1)^{x_u \oplus x_{v_a}} (1 - p_2)^{\theta(x_u, \mathbf{x}_{\mathcal{N}_u^{\text{out}}, x_{v_1}, \dots, x_{v_a}})} \right] \right\}. \quad (4.49)$$

Here,  $\mathcal{N}'_u$  denotes the incoming neighborhood  $\mathcal{N}_u^{\text{in}}$  if  $j > 0$ , or the incoming neighborhood minus the first incoming neighbor,  $\mathcal{N}_u^{\text{in}} \setminus v_1$ , in the case  $j = 0$  where the first neighbor  $v_1$  is used as the qubit to which all remaining edges are connected.

## 4.5 Fidelity

The fidelity for a distributed graph  $G$  is  $F_N^G = \frac{1}{2^N} \sum_{\mathbf{x}} \langle \mathbf{K}_{\mathbf{x}}^G \rangle$ , and is in general hard to calculate. In order to simplify the analysis, here we will use a single error parameter  $p = p_1 = p_2$ . Then, in the case of the linear cluster state, which has a simpler form, we will differentiate again between the two error parameters and compare their effect in the protocol. Most of the results presented here can be extended to more general dependencies between the CPHASE and measurement errors, which strongly depend on the physical implementation. With this particular parametrization, the effect of errors enter as factors

$$(1-p)^{h(\mathbf{x})} = e^{-\beta h(\mathbf{x})}, \quad (4.50)$$

where we have defined  $\beta = -\log(1-p)$ . To first order in  $p$ , the correlators of the purified graph states, Eq. (4.36), can also be written in the same form,

$$1 - h(\mathbf{x})p \sim e^{-ph(\mathbf{x})} \sim e^{-\beta h(\mathbf{x})}. \quad (4.51)$$

Hence, the correlators of the generated graph state can be written as

$$\langle \mathbf{K}_{\mathbf{x}} \rangle = \exp[-\beta H^G(\mathbf{x})], \quad (4.52)$$

where  $H^G(\mathbf{x})$  is the sum of the different noise terms  $h(\mathbf{x})$ . The fidelity thus resembles the partition function of a system with Hamiltonian  $H^G(\mathbf{x})$  and inverse temperature  $\beta$ :

$$F_N^G = \frac{1}{2^N} \sum_{\mathbf{x}} e^{-\beta H^G(\mathbf{x})}. \quad (4.53)$$

The ‘‘Hamiltonian’’  $H^G(\mathbf{x})$  can be expressed as the sum of many-body local Hamiltonians of the form

$$\theta(x_1, x_2, \dots, x_n) = 1 - \bar{x}_1 \bar{x}_2 \dots \bar{x}_n \quad (4.54a)$$

$$x_u \oplus x_v = x_u \bar{x}_v + \bar{x}_u x_v \quad (4.54b)$$

$$\bar{x}_u \left[ \frac{\sum_{b=1}^j x_b}{2} \right] = \frac{1}{2} \left( \bar{x}_u \sum_{b=1}^j x_b + \bar{x}_u \bigoplus_{b=1}^j x_b \right). \quad (4.54c)$$

The last term  $\bigoplus_{b=1}^j x_b$  in Eq. 4.54c is a  $j$ -body interaction term. By recursively using Eq. 4.54b, this term can be seen to be equal to the sum over all index permutations of  $\sum_{a \text{ odd}} x_1 \dots x_a \bar{x}_{a+1} \dots \bar{x}_j$ . That is, we have rephrased our problem of computing the fidelity of a distributed large graph state as that of computing the thermal properties of a classical many-body Ising-type system, where the indices  $x$  take the role of classic spins (with possible values 0 and 1). The corresponding Hamiltonian will inherit the topology of the underlying graph and its precise expression will depend on the graph-growth protocol used.

We are interested in the rate at which  $F_N^G$  decays,

$$\beta f_N^G = -\frac{1}{N} \log F_N^G, \quad (4.55)$$

where  $Nf_N^G$  is the analog of the free energy of the system. A good reason to study this quantity is that in statistical systems such as a complex network, modeled as an ensemble, the partition function (here, the fidelity) in itself is not an extensive quantity, while the free energy is typically extensive and self-averaging (see, for example, the discussion in [Amit, 1989](#), p. 188).

We can further exploit the statistical physics analogy and apply the known methods and understanding to compute the rate at which the fidelity decays for the different proposed protocols. We are interested in a regime where the noise in the operations is low, which corresponds to the high temperature limit. In addition, in the cases under study, each spin (or node) is effectively coupled to several spins, either as nearest- or second-nearest neighbors. These are conditions for which mean-field approximation is very well suited: Eqs. 4.54a to 4.54c can be linearized using the standard mean-field approximation to express the Hamiltonian as the sum of a constant term plus linear terms in  $x_u$ . For this purpose we take  $x_u \rightarrow s + \delta_u$  (and  $\bar{x}_u \rightarrow 1 - s - \delta_u$ ) where  $s$  is the value of the mean-field and  $\delta_u$  are the arguably small fluctuations of  $x_u$  around its mean value. Keeping only the linear terms in the fluctuations, the different terms present in the Hamiltonian become:

$$\theta(x_1, x_2, \dots, x_n) = 1 - (1 - s)^n + (1 - s)^{n-1} \sum_{a=1}^n \delta_a \quad (4.56a)$$

$$x_u \oplus x_v = 2s(1 - s) + (1 - 2s)(\delta_u + \delta_v) \quad (4.56b)$$

$$\begin{aligned} \bar{x}_u \left[ \frac{\sum_b x_b}{2} \right] &= \frac{1 - s}{2} \left( js + \sum_{b=1}^j \delta_b \right) - \frac{\delta_u}{2} js \\ &+ \frac{1 - s}{2} \left\{ \frac{1}{2} [1 - (1 - 2s)^j] + (1 - 2s)^{j-1} \sum_{b=1}^j \delta_b \right\} \\ &- \frac{\delta_u}{2} \frac{1}{2} [1 - (1 - 2s)^j]. \end{aligned} \quad (4.56c)$$

With this linearization, the new mean-field Hamiltonian takes the form  $H^{\text{MF}} = \sum_u A_u + \sum_u B_u x_u$ , where  $A_u$  and  $B_u$  are functions of  $s$ . Hence, the sum over  $\mathbf{x}$  (ie “configurations”) in the fidelity can now be carried out with ease,

$$F^{\text{MF}} = \frac{1}{2^N} \sum_{\mathbf{x}} \prod_{u \in V} e^{-\beta A_u} e^{-\beta B_u x_u} = \prod_{u \in V} e^{-\beta A_u} e^{-\beta B_u / 2} \left( \cosh \frac{\beta B_u}{2} \right), \quad (4.57)$$

and the decay rate becomes  $\beta f^{\text{MF}}$ , with

$$f^{\text{MF}} = \frac{1}{N} \sum_{u \in V} A_u + \frac{1}{N} \sum_{u \in V} \frac{B_u}{2} - \frac{1}{\beta N} \sum_{u \in V} \log \cosh \frac{\beta B_u}{2}. \quad (4.58)$$

The value of  $s$  (the magnetization) is found by adding an artificial linear term to the Hamiltonian (playing the role of an external magnetic field), changing  $B_u \rightarrow B_u + \xi$ , and requiring consistency in the definition

$$s = \langle x_u \rangle = \left. \frac{\partial f^{\text{MF}}}{\partial \xi} \right|_{\xi \rightarrow 0}.$$

One is hence left with the transcendental equation for  $s$ ,

$$s = \frac{1}{2} - \frac{1}{2N} \sum_u \tanh \frac{\beta B_u}{2}. \quad (4.59)$$

Its solution  $s^*$  can be substituted back in the expression for  $f^{\text{MF}}$  (Eq. 4.58) to obtain the desired result. In the cases of interest here,  $B_u$  is some polynomial which remains bounded for all values of  $s \in [0, 1]$ . Hence, to leading order in  $p$  we can approximate  $\tanh \frac{\beta B_u}{2} \approx \frac{p B_u}{2}$ , arriving at

$$s \approx \frac{1}{2} - \frac{1}{2} p B_u|_{s=1/2} \quad (4.60)$$

for small enough  $p$ .

## 4.6 Distribution of a closed linear cluster

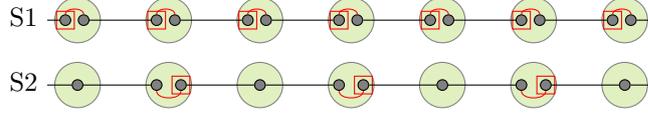
We first study the case of a one-dimensional network, in which a linear cluster state is created. This network is remarkable because we can compute the exact fidelity for any cluster size. In this case, we will keep error parameters  $p_1$  and  $p_2$  independent and study which is their effect in the different protocols. Moreover, we will first consider an ideal situation where all the purified graphs are perfect, and later turn to the noisy purification scenario.

For symmetry, we consider a closed linear cluster state, where all nodes have degree 2. The order of the CPHASES in the creation of the local graph in protocols Bipartite A and B, and the size of subgraphs in protocol Subgraphs, gives different results for the fidelity. For simplicity, in Bipartite A and B, we consider CPHASES applied to successive nodes. Except for the noise of the first gate (which affects only two nodes) and of the last one (which affects four nodes), all the gates contribute to the noise of three nodes. Each of the correlators is thus

$$\begin{aligned} \langle \mathbf{K}_x \rangle &\simeq \prod_{u \in V} (1 - p_1)^{x_u} (1 - p_1)^{x_u + x_{u-1} \oplus x_{u+1}} (1 - p_2)^{2\theta(x_{u-1}, x_u, x_{u+1})} \\ &\quad \left\langle (K_{u_a}^{g_u})^{x_{u-1} \oplus x_{u+1}} (K_{u_b}^{g_u})^{x_u} \right\rangle \end{aligned} \quad (4.61)$$

for the Bipartite A protocol and

$$\langle \mathbf{K}_x \rangle \simeq \prod_{u \in V} (1 - p_1)^{x_u} (1 - p_2)^{\theta(x_{u-1}, x_u, x_{u+1}, x_{u+2})} \left\langle (K_u^{g_u})^{x_u} (K_{u_b}^{g_u})^{x_{u-1} \oplus x_{u+1}} \right\rangle \quad (4.62)$$



**Figure 4.5.** Subgraph protocol in a linear cluster. Top: S1, using subgraphs of degree 1. Bottom: S2, using subgraphs of degree 2.

for the Bipartite B protocol. Recall that the correlator of the purified subgraphs is given in Eq. 4.36.

The performance of the subgraph protocol, on the other hand, depends on the choice of size of the subgraphs. There are two extreme strategies, depicted in Figure 4.5: create subgraphs of degree 1, and connect them successively (S1), or create subgraphs of degree 2 every second node, and connect them at their common neighbor (S2). One could also adopt an intermediate strategy, in which the degree of each GHZ is selected at random, resulting in a mix of subgraphs of degree 1 and 2 —its performance, which will not be reported here, falls in between the two extreme strategies. In the first case (S1), Eq. 4.49 reads

$$\langle \mathbf{K}_x \rangle = \prod_{u \in V} (1 - p_2)^{\theta(x_u, x_{u+1}, x_{u+2})} (1 - p_1)^{x_u \oplus x_{u+1}} \left\langle (K_u^{g_u})^{x_u} (K_{u+1}^{g_{u+1}})^{x_u \oplus x_{u+1}} \right\rangle, \quad (4.63)$$

while in the second (S2) it is

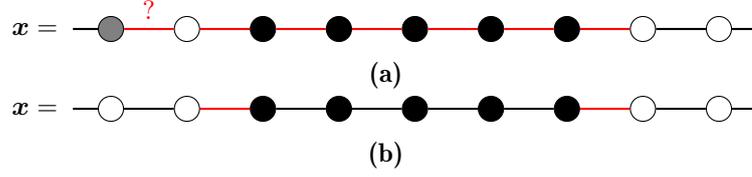
$$\langle \mathbf{K}_x \rangle = \prod_{u \in V_{\text{even}}} (1 - p_2)^{\theta(x_u, x_{u+1}, x_{u+2})} (1 - p_1)^{x_{u+1} \oplus x_{u+2}} \left\langle (K_u^{g_u})^{x_u} (K_{u-1}^{g_{u-1}})^{x_u \oplus x_{u-1}} (K_{u+1}^{g_{u+1}})^{x_{u+1}} \right\rangle. \quad (4.64)$$

Note that in the latter the product is over  $u = 2, 4, \dots \in V_{\text{even}}$ .

### 4.6.1 Exact solution via generating functions of the domains

Let us now define a domain of  $\mathbf{x} = (x_1 x_2 \dots x_N)$  as a sequence  $(x_u x_{u+1} \dots x_{u+l})$  where all  $x_v$ , for  $v = u, \dots, u+l$ , have the same value (either 0 or 1), and where  $x_{u-1}$  and  $x_{u+l+1}$  have a different value. The noise functions in the exponents of correlators  $\langle \mathbf{K}_x \rangle$  depend on certain parameters of the domains of  $\mathbf{x}$ . Hence, one can express  $\langle \mathbf{K}_x \rangle$  as a function of these parameters, and reduce the complexity of the sum of the fidelity in Eq. 4.15.

Let us take Eq. 4.63 as an example. We first define  $n$  as total number of ones in the sequence  $\mathbf{x}$ ,  $n = |\mathbf{x}|$ ;  $c_1$  as the number of domains of only one zero; and  $c_2^*$  as the number of domains with two or more zeros. The function  $\theta(x_u, x_{u+1}, x_{u+2})$  that accompanies the term with  $p_2$  is equal to 1 in all the red edges of Figure 4.6a. That is, the domain of ones depicted in the Figure contributes to the number of  $(1 - p_2)$  noises with its size plus two. Note, however, that if the  $x$  in gray is actually



**Figure 4.6.** Schematic representation of the vector  $\mathbf{x}$  and the noise effect on  $\langle \mathbf{K}_x \rangle$  of Eq. 4.63. Dots are components  $x$  of the vector: black if  $x = 1$  and white if  $x = 0$ . Lines correspond to edges between the qubits associated to every  $x$ . (a) Red lines mark edges that are affected by noise  $1 - p_2$ . The line with a question mark is affected by noise, but should not be counted if the gray  $x$  is actually equal to 1. (b) Same, with noise  $1 - p_1$ .

equal to 1 (ie the domain of ones is preceded by a domain of only a 0), the edge with the question mark should not be counted, as it will be included in that other domain. Summing over all domains, there are  $n + c_1 + 2c_{2^*}$  terms of the noise  $(1 - p_2)$ . The function  $x_u \oplus x_{u+1}$  that accompanies the term with  $p_1$  is equal to 1 only in the domain walls, as shown in Figure 4.6b. Hence, there are  $2(c_1 + c_{2^*})$  terms of the noise  $(1 - p_1)$ . The terms  $\langle (K_u^{g_u})^{x_u} (K_{u+1}^{g_{u+1}})^{x_u \oplus x_{u+1}} \rangle$  behave similarly. In all, Eq. 4.63 can be expressed in terms of the domain parameters of  $\mathbf{x}$  as

$$\langle \mathbf{K}_x \rangle = (1 - p_2)^{c_1 + 2c_{2^*} + n} (1 - p_1)^{2(c_1 + c_{2^*})} \langle K_{u_v}^{g_u} \rangle^{c_1 + c_{2^*}} \langle K_u^{g_u} \rangle^{n - c_1 - c_{2^*}} \langle K_u^{g_u} K_{u_v}^{g_u} \rangle^{c_1 + c_{2^*}}. \quad (4.65)$$

Using Eq. 4.15, the fidelity is then

$$F_N = \frac{1}{2^N} \sum_{c_1, c_{2^*}, n} g(c_1, c_{2^*}, n, N) \langle \mathbf{K}_x(c_1, c_{2^*}, n) \rangle, \quad (4.66)$$

where  $g(c_1, c_{2^*}, n, N)$  is the number of  $\langle \mathbf{K}_x(c_1, c_{2^*}, n) \rangle$  elements with these parameters in a graph of  $N$  vertices. This sum can be computed exactly by turning to generating functions. The function generating  $g(c_1, c_{2^*}, n, N)$  is

$$G(x, y_1, y_2, z) = \sum_{c_1, c_{2^*}, n, N} g(c_1, c_{2^*}, n, N) x^n y_1^{c_1} y_2^{c_{2^*}} z^N. \quad (4.67)$$

The fidelity for a state of size  $N$  corresponds to the  $N$ -th term of  $G$ ,

$$F = \frac{1}{2^N} \frac{1}{N!} \partial_z^N G(x, y_1, y_2, z), \quad (4.68)$$

evaluated at some specific values of  $x$ ,  $y_1$ ,  $y_2$ , and  $z = 1$  according to Eq. 4.65. The factorization of  $\langle \mathbf{K}_x \rangle$  for Subgraphs S2 and Bipartite protocols depend on slightly different parameters. We derive them in the following four subsections. In all cases, errors from CPHASES and measurements are tracked by error parameters  $p_2$  and  $p_1$ , while errors in the purification step enters in the correlators  $\langle K \rangle$ . For simplicity, we will consider the first order approximation of these purification errors (tracked by error parameter  $p$ ), from which the perfect purification case can be obtained setting  $p = 0$ .

Subgraphs S1:  $g(n, c_1, c_{2^*}, N)$

As we already introduced, in a chain of  $N$  nodes, let  $n$  be the total number of ones,  $c_1$  the number of domains of ones preceded by a domain of only a zero, and  $c_{2^*}$  the number of domains of ones preceded by a domain of at least two zeros ( $c_1 + c_{2^*}$  is the total number of domains of ones). Recall Eq. 4.65, which gives the correlator of the protocol Subgraph S1 in terms of  $n$ ,  $c_1$  and  $c_{2^*}$ . Then,  $g(n, c_1, c_{2^*}, N)$  is the number of different configurations of that chain with given parameters, and  $G(x, y_1, y_2, z)$  its generating function, as defined in Eq. 4.67. Each variable  $x$ ,  $y_1$ ,  $y_2$ , and  $z$  “counts” the number of ones, domains of one zero, domains of two or more zeros and the total size, respectively.

The function  $G$  can be found by joining simpler distributions. We can think of the linear chain as a construction of domains of zeros and ones joined together. Consider the set of domains of zeros,  $\{0, 00, 000, \dots\}$ , each domain of a given size appearing only once. The number of domains of size  $N$  in this set is  $d_N = 1$ , which is generated by

$$D(z) = \sum_{N \geq 1} d_N z^N = \frac{z}{1-z}. \quad (4.69)$$

We can differentiate between the set of domains of only a zero,  $\{0\}$ , and that of two or more zeros,  $\{00, 000, \dots\}$ . In this case, the generating functions are respectively  $z$  and  $D(z) - z$ . The set of domains of ones,  $\{1, 11, 111, \dots\}$ , is generated by the same function, but here each element contributes to the total size of the chain and to the number of ones. Its generating function is thus  $D(xz)$ .

The function generating the set of pairs of domains, the first of zeros and the second of ones, is

$$P \equiv P(x, y_1, y_2, z) = \underbrace{\{y_1 z + y_2 [D(z) - z]\}}_{\text{zeros}} \underbrace{D(xz)}_{\text{ones}}. \quad (4.70)$$

To clarify this expression let us recall that  $y_1$  keeps track or “counts” the number of domains of a single zero,  $y_2$  the number of domains of more than one zero,  $x$  the number of ones and  $z$  the total size of the chain. And again, the coefficients in the power expansion of  $P(x, y_1, y_2, z)$  are the number of configurations of a pair of domains with the specified parameters (by different variables), starting with a domain of zeroes followed by a domain of ones. Now, the generating function corresponding to a chain formed by a sequence of such pairs (including no pair at all) is

$$1 + P + P^2 + \dots = \frac{1}{1-P} = 1 + \{y_1 z + y_2 [D(z) - z]\} \frac{D(xz)}{1-P}. \quad (4.71)$$

Here, the first element (1) counts the case where there are no pairs at all, and the second,  $P/(1-P)$ , to that where there is at least one pair. Now, there are several ways we can complete the sequence of domain pairs to construct a closed ring. We can close the ring as it is (a), we can add a possible domain of zeros at the end (b),

a possible domain of ones at the beginning (c), or both (d). The final generating function is then

$$G = \left\{ 1 + [y_1 z + y_2 (D(z) - z)] \frac{D(xz)}{1-P} \right\} \left[ \overbrace{1}^{(a)} + \overbrace{D(xz)}^{(c)} + \overbrace{P}^{(d)} \right] + \left[ 1 + y_2 D(z) D(xz) \frac{1}{1-P} \right] \underbrace{D(z)}_{(b)}. \quad (4.72)$$

Note that when we added a domain of zeros (the term b), we changed the variable  $y_1$  to  $y_2$ , to take into account that the domain of zeros is now of size greater than one (because we are considering a closed linear chain). Simplifying, we obtain

$$G = \frac{1 - xz^2(1 - y_1 + 2(y_1 - y_2)z)}{1 - (1+x)z + x(1-y_1)z^2 + x(y_1 - y_2)z^3}. \quad (4.73)$$

The fidelity of a linear cluster of size  $N$  is thus given by Eq. 4.68 using

$$x = (1 - p_2) \langle K_u^{g_u} \rangle, \quad (4.74a)$$

$$y_1 = (1 - p_2)(1 - p_1)^2 \langle K_{uv}^{g_u} \rangle \langle K_u^{g_u} \rangle^{-1} \langle K_u^{g_u} K_{uv}^{g_u} \rangle, \quad (4.74b)$$

$$y_2 = (1 - p_2)^2 (1 - p_1)^2 \langle K_{uv}^{g_u} \rangle \langle K_u^{g_u} \rangle^{-1} \langle K_u^{g_u} K_{uv}^{g_u} \rangle, \quad (4.74c)$$

and  $z = 0$ . Setting all  $\langle \dots \rangle = 1$ , we have the perfect purification case.

**Bipartite B:**  $g(n, c_1, c_2, c_{3^*}, N)$

In Bipartite B protocol, the correlator depends on similar parameters. For simplicity, and since later we will use the first order approximation of Eq. 4.36, here we have to differentiate between the number of domains of zeros with one element ( $c_1$ ), two elements ( $c_2$ ) and three or more elements ( $c_{3^*}$ ):

$$\langle \mathbf{K}_x \rangle = (1 - p_1)^n (1 - p_2)^{c_1 + 2c_2 + 3c_{3^*} + n} (1 - 2p)^n (1 - p)^{2c_2 + 2c_{3^*}}. \quad (4.75)$$

This can be achieved by a small modification of the previous generating function. Now,

$$P \equiv P(x, y_1, y_2, y_3, z) = \left\{ y_1 z + y_2 z^2 + y_3 [D(z) - z - z^2] \right\} D(xz), \quad (4.76)$$

and

$$G = \frac{(1 + xz^2(-1 + y_1 - 2y_1 z + 2y_2 z + 3(-y_2 + y_3)z^2))}{(1 + z(-1 + x(-1 + z(1 + y_1(-1 + z) + z(y_2(-1 + z) - y_3 z))))). \quad (4.77)$$

Bipartite A:  $g(n, c_1, c_2^*, \bar{c}_2^*, N)$

In Bipartite A protocol, we also need to count the number  $\bar{c}_2^*$  of domains of two or more ones. The correlator in this case reads

$$\langle \mathbf{K}_x \rangle = (1-p_1)^{2n+2c_2^*+2\bar{c}_2^*} (1-p_2)^{2c_1+4c_2^*+2n} (1-2p)^{2c_2^*+2\bar{c}_2^*} (1-p)^{n-2\bar{c}_2^*}. \quad (4.78)$$

We count  $\bar{c}_2^*$  using variable  $w_2$ , and differentiating between the sets  $\{1\}$  and  $\{11, 111, \dots\}$ , which are generated by  $xz$  and  $D(xz) - xz$ , respectively. The extended function generating the set of pairs of domains is now

$$P \equiv P(x, y_1, y_2, w_2, z) = \{y_1 z + y_2 [D(z) - z]\} \{xz + w_2 [D(xz) - xz]\}. \quad (4.79)$$

Proceeding as in the previous case, we obtain

$$G = \frac{1 + xz^2(-1 + y_1 - 2y_1 z + 2y_2 z) + (-1 + w_2)x^2 z^3(y_1(2 - 3z) + 3y_2 z)}{1 + z(-1 + x(-1 + z(1 + (1 + (-1 + w_2)xz)(y_1(-1 + z) - y_2 z)))}. \quad (4.80)$$

Subgraphs S2:  $g(n_{01}, n_{10}, n_{11}, c_l, c_r, N)$

In the Subgraphs S2 protocol, the sum is performed over even nodes. In this case, it is convenient to express  $\mathbf{x}$  as a sequence of elements 00, 01, 10 and 11 (the first digit corresponding to an odd node, and the second to an even node). Each 01, 10 and 11 contribute to one CPHASE noise, as well as each domain of 00 which is preceded by a 01 or a 11. Moreover, each 01 and 10 contribute to one  $Y$  measurement noise. Finally, each 01 and 11 contribute to a  $1 - 3p$  noise of the purified subgraph, each 10 to a  $1 - p$  and each domain of 00 followed by a 10 or a 11 also to a  $1 - p$ . Thus, we need the number of configurations with  $n_{01}$ ,  $n_{10}$ , and  $n_{11}$  number of 01, 10 and 11 elements, and  $c_l$  and  $c_r$  domains of 00 preceded by 01 or 11 and followed by 10 or 11, respectively. The correlator is

$$\langle \mathbf{K}_x \rangle = (1-p_2)^{n_{01}+n_{10}+n_{11}+c_l} (1-p_1)^{n_{01}+n_{10}} (1-3p)^{n_{01}+n_{11}} (1-p)^{n_{10}+c_r}. \quad (4.81)$$

Now, each element contributes with  $z^2$  to the size of the chain. The domains  $\{00, 00 00, \dots\}$  are generated by  $\frac{z^2}{1-z^2}$ . A domain made of elements 01, 10 and 11 of any size (including 0) is generated by

$$\frac{1}{1 - (x_{01} + x_{10} + x_{11})z^2}, \quad (4.82)$$

and one which ends (or begins) with, say, element 01 (and thus is of size at least 2) is generated by

$$\frac{1}{1 - (x_{01} + x_{10} + x_{11})z^2} x_{01} z^2. \quad (4.83)$$

Proceeding as in the previous cases, the generating function is

$$G = \left[ 1 + \frac{(x_{01} + x_{10} + x_{11})z^2}{1 - (x_{01} + x_{10} + x_{11})z^2} + \frac{z^2}{1 - z^2} + G \right] \frac{1}{1 - P}, \quad (4.84)$$

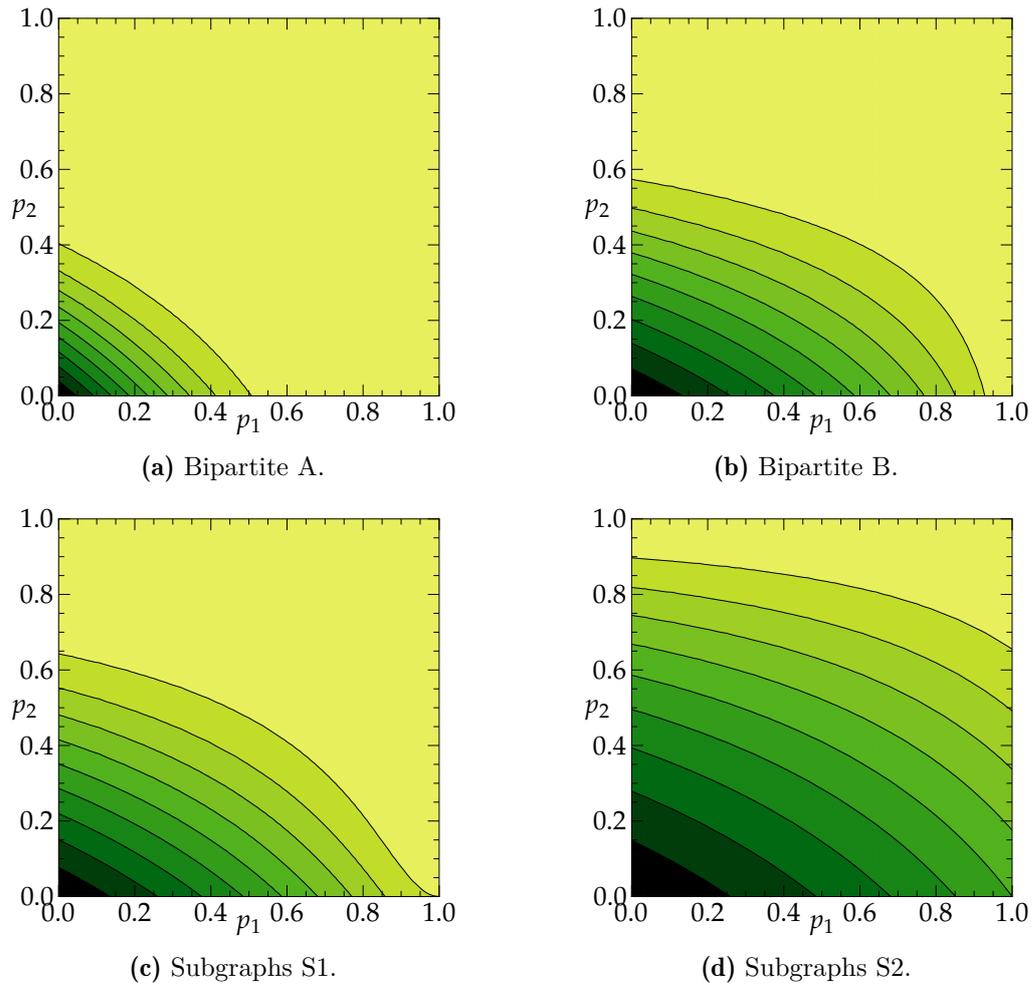
where  $P$  is equal to

$$\left\{ (x_{01}y_l + x_{10}y_r + x_{11}y_ly_r)z^2 + \frac{1}{1 - (x_{01} + x_{10} + x_{11})z^2} z^4 [(x_{01} + x_{11})y_l + x_{10}] [(x_{10} + x_{11})y_r + x_{01}] \right\} \frac{z^2}{1 - z^2}. \quad (4.85)$$

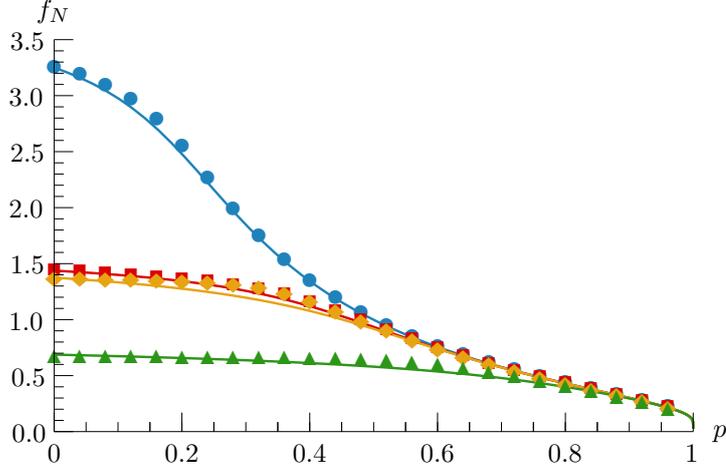
### Comparison of the fidelity

We will first consider the ideal case of perfect purification, and later tackle the noisy case. In the ideal scenario, we can substitute all subgraph correlators by 1 and Eq. 4.68 gives the exact fidelity for all values of  $p_1$  and  $p_2$  (here taken to be independent parameters). The fidelities of the four protocols are plotted in Figure 4.7, the best protocol being S2 (d). The fidelity decays at first exponentially in the error probabilities but as probability of error increases the decay becomes slower and the fidelity reaches a non-zero minimum value. In general, the fidelity is more sensible to noise  $p_2$ , as this is the one corresponding to CPHASES, which affects more qubits. The dependence on  $p_1$  varies a lot from one protocol to another. This is reflected in the asymptotic minimum value: for  $p_2 = 0$  and  $p_1 = 1$  (ie perfect two-qubit gates, but maximally noisy single-qubit measurements), the fidelity in the S1 and S2 protocols is higher than the minimum value  $2^{-N}$ . At this regime all terms in the sums Eq. 4.63 and 4.64 vanish except for those where the exponent of  $(1 - p_1)^{f(\mathbf{x})}$  is zero (ie the corresponding  $\mathbf{K}_{\mathbf{x}}$  is not affected by the noise). Of course, this is true in all protocols for the identity,  $\mathbf{x} = \mathbf{0}$ , but in the S1 it is also true for  $\mathbf{x} = \mathbf{1}$ , so the minimum fidelity is  $2^{-N+1}$ . In protocol S2 the effect is even bigger, as the product is over  $V_{\text{even}}$  and hence any  $\mathbf{x}$  where pairs  $x_{2n+1} \oplus x_{2n+2} = 0$  for  $n = 0, \dots, N/2 - 1$  translates in a  $\mathbf{K}_{\mathbf{x}}$  that is not affected by this noise. In this case, the minimum fidelity is  $2^{-N/2}$ .

To further compare the four protocols, in Figure 4.8 we plot the corresponding decay exponent  $f_N$  for  $p = p_1 = p_2$ . As expected, the behavior of Bipartite A protocol is the most sensible to noise, because it involves more operations and measurements. One can also observe that S1 and Bipartite B have a similar decay rate. Note that, except for the use of a central node in Bipartite B, while S1 protocol is distributed, both use bipartite resources, and in particular both implement the same number of CPHASES and measurements. For this reason, noise comes in a very similar way, the main difference being that CPHASES in Bipartite B affect a larger number of qubits, and therefore the associated  $\theta$  function has a larger support in  $\mathbf{x}$ . This holds even if one considers non-perfect purification, as



**Figure 4.7.** Fidelity of a linear cluster of size  $N = 10$  created by the (a) Bipartite A, (b) Bipartite B, (c) Subgraph S1 and (d) Subgraphs S2 protocols, as a function of noise parameters  $p_1$  and  $p_2$ . Lines correspond to fidelities  $2^{-1}, 2^{-2}, \dots, 2^{-9}$ , starting from the bottom left corner.



**Figure 4.8.** Rescaled decay rate  $f_N$  of a closed linear cluster of  $N = 10$  nodes with perfect purification. Blue circles: Bipartite A; red squares: Bipartite B; yellow diamonds: Subgraphs S1; and green triangles: Subgraphs S2. Dots correspond to the mean-field approximation (see Section 4.6.2), lines to the exact result using generating functions.

both protocols rely on bipartite purification. On the other hand, S2 performs the merging of subgraphs at every second node, so it has much less sources of noise. In fact, for  $p \rightarrow 0$  its decay rate is exactly half the one of S1.

#### 4.6.2 Comparison with the mean-field approximation

To give insight into the methods that are available for general networks, we show here how the common mean-field approximation can be used. As we introduced in Section 4.5,  $\langle \mathbf{K}_x \rangle \sim \exp[-\beta H(\mathbf{x})]$ . For the Subgraph S1 protocol,  $H(\mathbf{x}) = \sum_{u \in V} h_u(\mathbf{x})$ , with

$$h_u(\mathbf{x}) = \theta(x_u, x_{u+1}, x_{u+2}) + x_u \oplus x_{u+1}. \quad (4.86)$$

Recall that we are still considering perfect purification. Substituting  $x_u \rightarrow s + \delta_u$  and keeping only linear terms in  $\delta_u$ , this takes the form of  $h_u^{\text{MF}}(\mathbf{x}) = a_u + b_u x_u + c_u x_{u+1} + d_u x_{u+2}$ , with

$$a_u = (5 - 2s)s^2 \quad (4.87a)$$

$$b_u = 2 - 4s + s^2 \quad (4.87b)$$

$$c_u = 2 - 4s + s^2 \quad (4.87c)$$

$$d_u = (1 - s)^2. \quad (4.87d)$$

All qubits  $u$  are equivalent, so the coefficients  $a, b, c, d$  are the same for every node and the total Hamiltonian is  $H^{\text{MF}} = AN + B \sum_u x_u$ , with  $A = a$  and  $B = b + c + d$ .

Hence, the fidelity in the mean-field approximation is

$$F_N^{\text{MF}} = e^{-\beta AN} e^{-\beta BN/2} \left( \cosh \frac{\beta B}{2} \right)^N, \quad (4.88)$$

and the decay rate

$$f^{\text{MF}} = A + \frac{B}{2} - \frac{1}{\beta} \log \cosh \frac{\beta B}{2}. \quad (4.89)$$

The same approximation can be made for the other protocols. The local Hamiltonian of Subgraphs S2 is

$$h_u(\mathbf{x}) = \theta(x_u, x_{u+1}, x_{u+2}) + x_{u+1} \oplus x_{u+2}. \quad (4.90)$$

Note that, in this case, the total Hamiltonian is the sum for  $u$  even. In the case of Bipartite A and B, the local Hamiltonians are

$$h_u(\mathbf{x}) = 2x_u + x_{u-1} \oplus x_{u+1} + 2\theta(x_{u-1}, x_u, x_{u+1}) \quad (4.91)$$

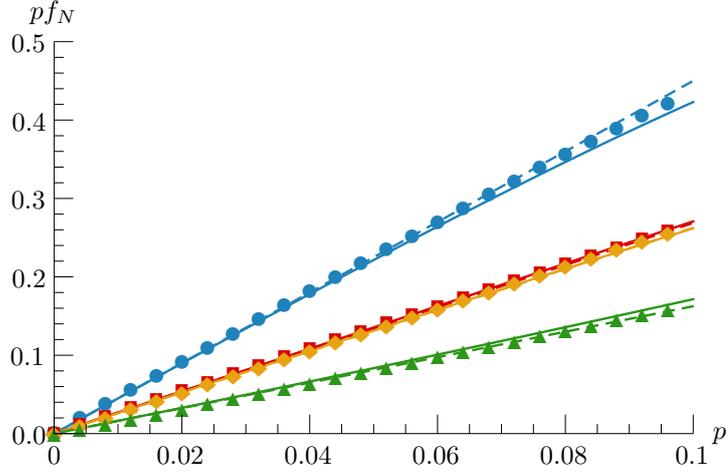
and

$$h_u(\mathbf{x}) = x_u + \theta(x_{u-1}, x_u, x_{u+1}, x_{u+2}), \quad (4.92)$$

respectively. The mean-field results are shown in Figure 4.8 together with the exact result. The agreement is remarkably good, specially considering that this is a one-dimensional network configuration. We observe that the decay rate has a linear dependence in  $p$  but soon higher order (non-linear) terms start to kick in. Here, we have studied and plotted the solution for a wide range of  $p$ . This has been done for completeness and to check the validity of the mean-field approximation, but recall that the aim in this Chapter is solely to compute the decay rate to leading order in  $p$ . The reason is that in realistic scenarios  $p$  will be strongly limited by threshold values required for subgraph (or bipartite) purification.

We now consider the realistic scenario following the same procedure as before but taking into account the corrections due the noisy purified subgraphs. This can be done by approximating the correlators by their linear correction around unity (ie Eq. 4.36). Consequently, an additional term (Eq. 4.51) must be added to the local Hamiltonians (Eqs. 4.86, 4.90, 4.91, and 4.92) for each of the different protocols. It is important to bear in mind that all subsequent results have to be taken consistently up to leading order in  $p$ . This leading order is the one that enters in the features that we seek: the linear dependence of the fidelity decay rate around a small values  $p \sim 0$  (and for arbitrarily large  $N$ ). From Eqs. 4.58 and 4.59 it is immediate to see that at this order the whole contribution to the decay rate is given by the first two terms in Eq. 4.58 evaluated at  $s = 1/2$ , ie

$$f_N = A + \frac{B}{2} \Big|_{s=1/2} = \frac{1}{N2^N} \sum_{\mathbf{x}} H(\mathbf{x}). \quad (4.93)$$



**Figure 4.9.** Fidelity decay rate  $pf_N$  of a closed linear cluster of  $N = 100$  nodes. From upper to lower lines: Bipartite A (blue, circles), Bipartite B (red, squares), Subgraph S1 (yellow, diamonds) and Subgraph S2 (green, triangles). Solid lines correspond to the generating function result, using the first order approximation of the purification scheme of Eq. 4.36. Dashed lines are the first-order result of Eq. 4.93. Dots correspond to the mean-field approximation.

The second equality is straightforward to get from the definitions of  $A$  and  $B$ , and it states that in this regime (low  $p$ ) the decay rate is dominated by the exponents of the typical sequences  $\mathbf{x}$ . The expected value of  $H(\mathbf{x})$  (over sequences  $\mathbf{x}$ ) can be easily carried out since  $x_u$  are independent variables of mean  $1/2$  and it leads to fidelities that decay as  $\exp(-Npf_N)$  with  $f_N$  equal to  $9/2 = 4.5$ ,  $43/16 \approx 2.7$ ,  $21/8 \approx 2.6$ , and  $13/8 \approx 1.6$  for the Bipartite A and B and Subgraphs S1 and S2, respectively.

These coincide with the decay rates obtained by the generating function method in the same limit, where in Eqs. 4.74a to 4.74c we substitute  $\langle K_u^{g_u} \rangle = \langle K_u^{g_u} K_{u_v}^{g_u} \rangle = 1 - 2p$  and  $\langle K_{u_v}^{g_u} \rangle^{c_1+c_2^*} = 1 - p$ . The results (see Figure 4.9) show that the subgraph protocols provide output fidelities comparable, if not better, than those given by the protocols based on channel purification. This is remarkable keeping in mind that purifying channels is much more demanding in terms of resources and efficiency.

As we mentioned at the beginning of the Section, the order of the CPHASES in the Bipartite protocols can give different results, but the decay rates do not change much. For example, if the CPHASES are first applied to every second edge, and then to the remaining edges, the decay rate is  $143/32 \approx 4.5$  for Bipartite A, while it does not change for Bipartite B.

## 4.7 Distribution of graph states in complex networks

In this section we study the behavior of the graph-growth protocols in complex networks. Recall from Section 2.2.1 that complex networks are characterized by statistical properties, and can be modeled as an ensemble of graphs  $\mathcal{G}$ , with a probability  $P(G)$  assigned to every graph  $G$  in the ensemble. A property  $O$  of a complex network is defined as its average over the ensemble,  $\overline{O} = \sum_{G \in \mathcal{G}} P(G) O^G$ . (Note that in the following the overline stands for the ensemble average, except when specifically mentioned otherwise). Recall also that some of these properties are self-averaging, meaning that for large systems (in the limit  $N \rightarrow \infty$ ) a property of a given graph realization  $G$  is the same as the average over different realizations of  $G \in \mathcal{G}$ . As we mentioned in Section 4.5, the free energy is typically self-averaging, and is the one we use here as a figure of merit.

We consider random networks with arbitrary (uncorrelated) degree distribution  $p_k$ , reproduced by the configuration model. Here, we will use the basic generating functions of the degree,  $g_p(z)$ , and the excess degree distributions,  $g_r(z)$ , from Section 2.3.2. The edges in the network are undirected, but the creation of the graph state via the Subgraph protocol is “directed-like”, as each node can have incoming and outgoing neighbors. Thus, we also consider the degree distribution  $p_{i,j}$ , where  $i$  is the in-degree and  $j$  the out-degree. This distribution is very important here, as it defines the implementation of the protocol. For example, in the linear cluster, S1 was defined by  $p_{1,1} = 1$  for all nodes, and S2 by  $p_{0,2} = 1$  and  $p_{2,0} = 1$  at even and odd nodes, respectively. The distribution  $p_{i,j}$  is constrained by  $\sum_{i,j} (j - i) p_{i,j} = 0$ , which means that  $\langle i \rangle = \langle j \rangle = \langle k \rangle / 2$ , but note that, in general, it can be correlated ( $p_{i,j} \neq p_i p_j$ ). The function that generates  $p_{i,j}$  is  $g_p(x, y) = \sum_{i,j} p_{i,j} x^i y^j$ .

In the case of complex networks, we are interested in the average  $\overline{f_N} = \sum_{G \in \mathcal{G}} P(G) f_N^G$  over the ensemble. We expect that, for large  $N$ ,  $f_N$  goes to  $\overline{f}$  due to self-averaging. In order to average  $f_N$ , we have to compute

$$\overline{f_N} = -\frac{1}{\beta N} \sum_{G \in \mathcal{G}} P(G) \log \frac{1}{2^N} \sum_{\mathbf{x}} e^{-\beta H^G(\mathbf{x})}, \quad (4.94)$$

which corresponds to a quenched average, where the disorder corresponding to the network topology, given by  $P(G)$ , is frozen with respect to that of the correlation operator index,  $\mathbf{x}$ . These type of averages appear, for example, in spin glasses (Mézard et al., 1987). Computing this quantity is in general an extremely challenging problem, specially in our case where we have structured disorder and k-body interaction between the spins. Nevertheless, we are interested in the particular regime of high temperatures (low  $p$ ) far away from critical phenomena, long range correlations, and other difficulties that appear at low temperatures. In this regime we can directly use Eq. 4.93:

$$\overline{f_N} = \frac{1}{N} \sum_{G \in \mathcal{G}} P(G) \frac{1}{2^N} \sum_{\mathbf{x}} H^G(\mathbf{x}) = \frac{1}{N} \overline{\frac{1}{2^N} \sum_{\mathbf{x}} H^G(\mathbf{x})}. \quad (4.95)$$

The Hamiltonians for the Bipartite protocols are of the form

$$H^G(\mathbf{x}) = \sum_{u \in V} h_u(\mathbf{x}) + \widetilde{\sum}_{(u,v) \in E} h_{u,v}(\mathbf{x}). \quad (4.96)$$

In the Bipartite A, the local Hamiltonians are

$$h_u(\mathbf{x}) = x_u + \overline{\bigoplus_{v \in \mathcal{N}_u} x_v} \cdot x_u + 2 \bigoplus_{v \in \mathcal{N}_u} x_v + \theta(x_u, \mathbf{x}_{\mathcal{N}_u}) + x_u + \bigoplus_{v \in \mathcal{N}_u} x_v \quad (4.97a)$$

and

$$h_{u,v}(\mathbf{x}) = \theta(x_u, x_v, x_{\tilde{\mathcal{N}}_u}, x_{\tilde{\mathcal{N}}_v}), \quad (4.97b)$$

while in the Bipartite B,

$$h_u(\mathbf{x}) = \overline{x_u} \bigoplus_{v \in \mathcal{N}_u} x_v + 2x_u + x_u \quad (4.98a)$$

and

$$h_{u,v}(\mathbf{x}) = \theta \left( x_u, x_v, \bigoplus_{w \in \mathcal{N}_u} x_w, \bigoplus_{w \in \mathcal{N}_v} x_w, x_{\tilde{\mathcal{N}}_u}, x_{\tilde{\mathcal{N}}_v} \right). \quad (4.98b)$$

(Here, the overline denotes the bit-complement.) As before, the tilde over the sum in Eq. 4.96 and  $\tilde{\mathcal{N}}$  reminds us that the CPHASES are performed in a particular order in the local graph. We can substitute this sum by the expected effect of an edge, which depends on the number of edges already connected to nodes  $u$  and  $v$ . This number is  $n_u + n_v$  with probability

$$\frac{1}{2^{k_u+k_v}} \binom{k_u}{n_u} \binom{k_v}{n_v}. \quad (4.99)$$

Here,  $k_u$  and  $k_v$  are the excess degrees of the vertices in edge  $(u, v)$ , so the average of the network ensemble has to be performed using probabilities  $r_{k_u}$  and  $r_{k_v}$ :

$$\overline{h_{u,v}(\mathbf{x})} = \frac{1}{2^N} \sum_{\mathbf{x}} \sum_{k_u k_v} r_{k_u} r_{k_v} h_{u,v}(\mathbf{x}). \quad (4.100)$$

In the Bipartite A, the average effect of each of these edges is  $1 - \frac{1}{4} [g_r(3/4)]^2$ . In the Bipartite B, it is  $1 - \frac{1}{16} [g_r(1/2) + g_r(3/4)]^2$ . The terms in  $h_u(\mathbf{x})$ , on the other hand, depend directly on the degree, and the average is performed over  $p_k$ :

$$\overline{h_u(\mathbf{x})} = \frac{1}{2^N} \sum_{\mathbf{x}} \sum_k p_k h_u(\mathbf{x}). \quad (4.101)$$

Considering all these terms, and that the summation over  $V$  contains  $N$  elements, while that over  $E$  contains  $M = \langle k \rangle N/2$ , the decay rates are

$$\overline{f_N} = \frac{15}{4} - \frac{5}{4}g_p(0) - \frac{1}{2}g_p(1/2) + \frac{\langle k \rangle}{2} \left( 1 - \frac{1}{4} [g_r(3/4)]^2 \right) \quad (4.102)$$

for Bipartite A and

$$\overline{f_N} = \frac{7}{4} - \frac{1}{4}g_p(0) + \frac{\langle k \rangle}{2} \left( 1 - \frac{1}{16} [g_r(3/4) + g_r(1/4)]^2 \right), \quad (4.103)$$

for Bipartite B.

In the Subgraph protocol, where  $H^G(\mathbf{x}) = \sum_{u \in V} h_u(\mathbf{x})$ , the local Hamiltonian is

$$\begin{aligned} h_u(\mathbf{x}) = & \left[ \overline{x_u} \left[ \frac{\sum_{w \in \mathcal{N}_u^{\text{out}}} x_w}{2} \right] + x_u(j^{(u)} + 1) \right] (1 - \delta_{0,j^{(u)}}) + x_u \delta_{0,i^{(u)}} \delta_{0,j^{(u)}} \\ & + \sum_{v_a \in \mathcal{N}'_u} \left[ \theta(x_u, \mathbf{x}_{\mathcal{N}_u^{\text{out}}}, x_{v_1}, \dots, x_{v_a}) + x_u \oplus x_{v_a} \right]. \end{aligned} \quad (4.104)$$

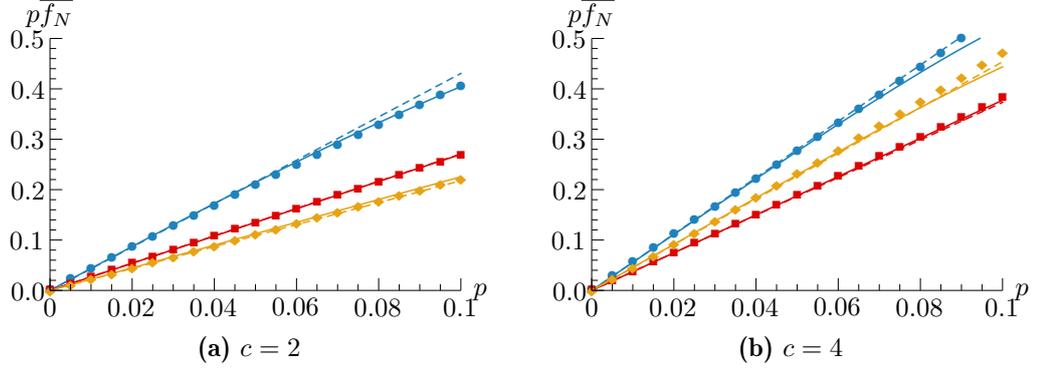
Here  $\mathcal{N}'_u$  is defined as in Eq. 4.49. That is, as the standard neighborhood if  $u$  has at least an outgoing edge ( $j^{(u)} > 0$ ), or as this neighborhood minus the first incoming neighbor if  $j^{(u)} = 0$ . In this case, to average the Hamiltonian we have to take into account the directed degree probability,  $\overline{h(\mathbf{x})} = \sum_{ij} p_{ij} h(\mathbf{x})$ , giving

$$\overline{f_N} = \frac{5}{8} + \frac{17 \langle k \rangle}{16} + \frac{7}{4}g_p(0, 0) - \frac{15}{8}g_p(1, 0) - \frac{1}{2} [g_p(1, 1/2) - g_p(1/2, 1/2)]. \quad (4.105)$$

These results are valid for any network with uncorrelated degree distribution  $p_k$ . To compare the behavior of the three protocols we consider the Erdős-Rényi model (Gilbert, 1959; Erdős and Rényi, 1959, 1960), which is the maximally random graph under the only constrain that the mean degree  $\langle k \rangle = c$  is fixed. Recall that the degrees in this model follow a Poisson distribution, so  $g_p(z) = g_r(z) = e^{c(z-1)}$ , and for  $c > 1$  there is a giant connected component of size comparable to the size of the network. The degree distribution  $p_k$  is fixed by the network model, but in the Subgraph protocol one can choose between different strategies and thereby fix the distribution  $p_{i,j}$  (of course, as long as  $\sum_{i,j} p_{i,j} \delta_{i+j,k} = p_k$ ). For simplicity, here we consider that the direction of each edge is selected at random,

$$p_{i,j} = \frac{p_{i+j}}{2^{i+j}} \binom{i+j}{i}, \quad (4.106)$$

and hence  $g_p(x, y) = e^{c(\frac{x+y}{2}-1)}$ . Figure 4.10 plots  $\overline{p f_N}$  for the three protocols, showing that the best protocol depends on the mean degree of the network. It also shows an average over 10 random realizations of an Erdős-Rényi graph of



**Figure 4.10.** Fidelity decay rate  $\overline{p f_N}$  in an Erdős-Rényi network of mean degree (a)  $c = 2$  and (b) 4. Bipartite A (blue, circles), Bipartite B (red, squares), and Subgraphs (yellow, diamonds). Points are the average over 10 random realizations of networks with  $N = 100$ , randomly sampling 1000 configurations of  $\mathbf{x}$ . Dashed lines are the results at first order in  $p$  (Eqs. 4.102, 4.103, and 4.104).

$N = 100$ , approximating the fidelity by the average of a random sample of 1000 configurations of  $\mathbf{x}$ ,  $F \sim \frac{1}{1000} \sum_{\mathbf{x}_{\text{sample}}} \langle \mathbf{K} \mathbf{x}_{\text{sample}} \rangle$ . This approximation is valid in the low  $p$  regime, where the fidelity is dominated by the typical values of  $\mathbf{x}$ . We observe that, in the Erdős-Rényi, the Subgraph protocol with random edge direction give a better (lower) decay rate  $\overline{f_N}$  for  $c < 2.8$ , while above that the Bipartite B beats it. This stronger dependence of the Subgraph protocol on the mean degree of the network—compare the term in  $\langle k \rangle$  in Eq. 4.104 versus those in Eq. 4.102 and 4.103, which however have a bigger constant term—is in part due to the higher number of noise sources, which depends on the number of edges. Indeed, in the Bipartite protocols only the initial CPHASES depend on the number of edges of the graph, while in the Subgraph protocol every merging of two subgraphs includes a CPHASE and a measurement. These mergings depend on the number of incoming edges, which could be lowered by considering a strategy different from the simpler one considered here where the direction of edges is selected at random. Instead, one could favor, for example, that the leaves of the network have an incoming edge with higher probability than an outgoing edge. Here it is important that we are comparing protocols which have drastically different requirements in terms of resources, so a slight benefit of the bipartite protocols in terms of fidelity does not rule out the use of subgraph protocols.

# CHAPTER 5

---

## Conclusions

---

In this thesis, we have considered distribution of bipartite and multipartite entanglement in large systems with a complex structure. Quantum complex networks offer a powerful framework for quantum communication and distributed tasks. Regardless of their intricate structure, complex networks can be studied by their statistical properties, which allows to analytically compute some interesting properties and to deal with them without knowing their exact structure.

In the bipartite scenario, long-distance entanglement can be established exploiting the emergence of a giant connected component in a percolation-type process. The percolation threshold, which amounts to the minimum level of entanglement needed to entangle two distant nodes with finite probability, and the size of the giant connected component, which determines this probability, have a strong dependence on the structure of the network. We have studied a local quantum preprocessing of the network that can change the structure of the network and thereby lower the percolation threshold, enabling quantum communication with less entanglement resources. The quantum preprocessing we have proposed is local in two senses. First, quantum operations are done always on qubits that belong to the same node. Second, the decision whether or not to perform such operation depends on the local structure of the network (the degree of the target node and the status of its neighbors) and on information about general statistical properties of the network—it is not required to possess the information about the precise wiring of all nodes, which can be overwhelmingly large, on many occasions unknown, and even changing in time. We provided analytical results for the relevant network properties with and without quantum preprocessing for arbitrary networks with uncorrelated degree distribution, showing that the preprocessing can substantially improve communication over such networks. We have also studied numerically the

Watts-Strogatz small world model and a real world network, and have found a similar behavior.

We also considered a more realistic scenario with the presence of noise in the connections. We have shown that in complex networks a direct implementation of the entanglement percolation strategy, without quantum preprocessing, allows for faithful quantum communication (above a fixed fidelity threshold) between a large number of nodes. The noise severely limits the number of steps or connections through which information is transmitted. However, in complex networks, one can reach a sizable amount of nodes with a moderately low number of steps. If the fidelity threshold allows for a path length slightly higher than the average path length, all nodes in the giant component become faithfully connected. The path length distribution is peaked at low values (scaling as  $\log N$  in complex networks versus  $N^{1/d}$  in  $d$ -dimensional lattices), and has finite width (constant in  $N$  versus  $N^{1/d}$ ). This shows that in complex networks a finite fraction of faithfully connected nodes appears for much smaller limiting path lengths and reaches the giant component size abruptly. Hence, here the advantage of complex networks is twofold: the average path length which marks the transition scales logarithmically with the network size, and the additional steps needed to reach the non-limited scenario is finite. This means that a small increase in the fidelity of the connections, for example via purification, can lead to an abrupt recovery of the non-limited scenario. A possible way to obtain this effect could be to use secondary paths to purify the entanglement in the shortest path connecting the two nodes that one wants to entangle (see [Lapeyre Jr. et al., 2012](#)).

Our results also contribute to the field of classical complex networks. We have given analytical results for the gain in the percolation thresholds and the size of the giant component for uncorrelated complex networks that undergo a set of local inversions (ie a transformation that produces the complement of the induced subgraph of the target node). These transformations are also related to the local complementation that leaves graph states with an equivalent entanglement. The problem at hand of studying how critical properties of a network can be drastically modified by a given set of network transformations might be of general interest to other disciplines in the field. In particular, the enhancement or delay of the percolation threshold is related to explosive percolation, a new area of study in complex networks that studies a first-order phase transition. Finally, we have addressed the problem of limited-path percolation in uncorrelated and small world complex networks, which can find various applications in a classical setting where one needs to exclude, eg due to some finite resources, paths exceeding a certain length.

In the multipartite scenario, we have focused on the creation and distribution of graph states, which are an important class with many practical applications, including measurement-based quantum computation. We consider a realistic model with errors in the channels, operations and measurement. To overcome this noise, we proposed a distributed protocol that again does not depend on information

about the global structure of the network, but only on the neighborhood of nodes. This protocol is thus specially motivated to be implementable in complex networks. Moreover, it uses multipartite purification of small subgraphs, which makes it efficient with respect to the network size. We compared it with two other protocols that rely on bipartite purification between a central node, which has a global overview of the network, and the rest of the nodes.

As a figure of merit, we used the fidelity of the generated graph state and its decay rate as the size of the network increases. Using generating function methods we have been able to compute exactly the fidelity for linear clusters of arbitrary size, allowing for a direct comparison with the approximate methods we develop. We have rephrased the problem of computing the effects of noise in the operations in terms of the thermal properties of a classical spin system, with the same interaction patterns as the underlying graph. Indeed, the fidelity itself can be seen as the analog of the partition function of such system, while its decay rate plays the role of the free energy. The well-known methods from statistical physics, such as the mean-field approximation, can be used to study its behavior. We have also studied the three protocols in networks with a complex structure, that give rise to effective classical spin models with quenched disorder. Our results show that the protocol using subgraph purification and merging is comparable, and in some cases even better, than those which rely on bipartite states. This is quite remarkable, as the latter rely on quantum repeaters and require much more resources.

In the complex networks case, the multipartite protocol could be possibly enhanced by devising an optimized strategy via the directed degree distribution  $p_{i,j}$ . As an example, in networks with many nodes of degree 1, one should go for a directed degree distribution in which those nodes are as much as possible the receivers of GHZ states, instead of the senders, so less connections would need to be made. In other words, one should tend to  $p_{1,0}$  as high as possible (compared to  $p_{0,1}$ ). Also, in nodes of too high degree, the protocol might fail due to the noise threshold in the multipartite purification, which depends on the size of the GHZs (and thus, on the degree of the network nodes). In this case, one could always separate the node in two (or more) and treat them as independent nodes, each creating and distributing a subgraph among a subset of the neighborhood. This might prove useful in networks with a scale-free degree distribution, which have a long tail and a high presence of hubs.

It remains an open question to relate the decay in fidelity with the actual use one can make of graph states. Clearly, there is a regime where the fidelity is exponentially small, but a finite decay rate still signatures valuable resources. For example,  $N$  independent, almost perfect Bell pairs result in a graph state with exponentially low fidelity; however this is clearly a very valuable resource. Hence, the decay rate of the fidelity (or alternatively, a local noise equivalent that one had to apply to each qubit of a perfect, large state to get the same fidelity) provides a more informative measure. It is important to emphasize that for the

three protocols, as well as for all the networks under study, we have not only computed this decay rate, but also completely characterized the noise graph states that emerge from the protocol. For a particular application, having full knowledge of the generated graph gives probably a closer picture of the generated resource.

As a final consideration, the structure of the final graph state have also an effect on the noise it accumulates and on the possible purification schemes that one can apply to it. Using a given underlying network, one could in principle generate distributed graph states with many different topologies. Here we have focused on a distributed graph that mimicked the structure of the underlying communication network, but of course one could generate other topologies. One could also distribute different states, and create a large graph state with a structure different from that of the network recalling that, once the graph is created, local operations (Van den Nest et al., 2004; Joo and L Feder, 2011) might be used to transform the graph state and reach the desired structure. Note that different graph states might be locally equivalent under these types of transformations, but the differences in their structure can translate into a different sensibility to noise in their generation.

An open question is if states like this—with a complex structure—are useful for some tasks, and in particular whether they are a universal resource for measurement-based quantum computation or not. It is known, for example, that a universal resource for measurement-based quantum computation must have unbounded entanglement width (Van den Nest et al., 2006), a measure connected to the rank width of the graph (Oum and Seymour, 2006). In the Gilbert model, for example, Lee et al. (2012) showed that in the regime  $p = c/N$  the rank width is less or equal than 2 below the threshold ( $c < 1$ ), and it scales at least as  $Nf(c)$  above it. Thus, below the threshold the graph state is not a resource for measurement-based quantum computation, but above it has unbounded rank width and it might be useful. This is, however, only a necessary condition. For instance, it is known that if the graph is fully connected it is locally equivalent to a giant GHZ (with bounded rank width), and hence not useful as a resource. Investigating if a graph above the threshold, but still in the sparse regime, is a resource is indeed an interesting open question. Of course, one can also try to generate other states that are useful for a specific task.

With this work, we would like to draw attention from quantum communication and information processing on regular lattices to the more general and feature-full complex networks. As we have seen, methods from the complex networks field allow to compute interesting quantities requiring only statistical properties. Although this might seem a limitation, it can in fact represent an advantage, as it makes mathematically tractable some problems that are hard or impossible to solve on lattices. Also, it provides a minimal description in scenarios where complete knowledge of the system is not available or is hard to provide. On the other hand, we have shown that new phenomena appear if networks and the operations one can perform on them are governed by the laws of quantum mechanics. An example of

that is also the new behavior found in the emergence of subgraphs in quantum random networks (Perseguers et al., 2010b). Finally, a quantum information approach could also be used to address classical problems, like detection of the community structure of a network, which might be revealed using quantum processes, eg by the evolution of a quantum walk (Tsomokos, 2011), and search algorithms (Paparo and Martín-Delgado, 2012; Garnerone et al., 2012; Sánchez-Burillo et al., 2012).



---

## Bibliography

---

- Achlioptas, D., D'Souza, R. M., and Spencer, J. (2009), Explosive percolation in random networks, *Science* **323**, 1453–5.
- Acín, A., Cirac, J. I., and Lewenstein, M. (2007), Entanglement percolation in quantum networks, *Nature Physics* **3**, 256–259.
- Albert, R. and Barabási, A.-L. (2002), Statistical mechanics of complex networks, *Reviews of Modern Physics* **74**, 47–97.
- Albert, R., Jeong, H., and Barabási, A.-L. (1999), Internet: Diameter of the World-Wide Web, *Nature* **401**, 130–131.
- (2000), Error and attack tolerance of complex networks, *Nature* **406**, 378–382.
- Amaral, L. A. N., Scala, A., Barthélémy, M., and Stanley, H. E. (2000), Classes of small-world networks., *Proceedings of the National Academy of Sciences of the United States of America* **97**, 11149–52.
- Amit, D. J., *Modeling Brain Function* (Cambridge University Press, 1989), 1st edn.
- Appel, K. and Haken, W. (1977), The Solution of the Four-Color-Map Problem, *Scientific American* **237**, 108–121.
- Araújo, N. and Herrmann, H. J. (2010), Explosive Percolation via Control of the Largest Cluster, *Physical Review Letters* **105**, 035701.
- Aschauer, H., Dür, W., and Briegel, H. J. (2005), Multiparticle entanglement purification for two-colorable graph states, *Physical Review A (Atomic, Molecular, and Optical Physics)* **71**, 012319.

- Barabási, A.-L. (1999), Emergence of Scaling in Random Networks, *Science* **286**, 509–512.
- Barabási, A.-L., Jeong, H., Néda, Z., Ravasz, E., Schubert, A., and Vicsek, T. (2002), Evolution of the social network of scientific collaborations, *Physica A: Statistical Mechanics and its Applications* **311**, 590–614.
- Bender, E. A. and Canfield, E. R. (1978), The asymptotic number of labeled graphs with given degree sequences, *Journal of Combinatorial Theory, Series A* **24**, 296–307.
- Bennett, C. H., Bernstein, H. J., Popescu, S., and Schumacher, B. (1996a), Concentrating partial entanglement by local operations, *Physical Review A (Atomic, Molecular, and Optical Physics)* **53**, 2046–2052.
- Bennett, C. H., Brassard, G., Crépeau, C., Jozsa, R., Peres, A., and Wootters, W. K. (1993), Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels, *Physical Review Letters* **70**, 1895–1899.
- Bennett, C. H., Brassard, G., Popescu, S., Schumacher, B., Smolin, J. A., and Wootters, W. K. (1996b), Purification of Noisy Entanglement and Faithful Teleportation via Noisy Channels, *Physical Review Letters* **76**, 722.
- Bennett, C. H., DiVincenzo, D. P., Smolin, J. A., and Wootters, W. K. (1996c), Mixed-state entanglement and quantum error correction, *Physical Review A (Atomic, Molecular, and Optical Physics)* **54**, 3824.
- Bialas, P. and Oles, A. K. (2008), Correlations in connected random graphs, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **77**, 036124–10.
- Bianconi, G. and Marsili, M. (2005), Loops of any size and Hamilton cycles in random scale-free networks, *Journal of Statistical Mechanics: Theory and Experiment* **2005**, P06005.
- Boguñá, M., Pastor-Satorras, R., and Vespignani, A. (2003), Absence of Epidemic Threshold in Scale-Free Networks with Degree Correlations, *Physical Review Letters* **90**, 028701.
- Bohman, T. and Frieze, A. (2001), Avoiding a giant component, *Random Structures and Algorithms* **19**, 75–85.
- Bollobás, B. (1980), A probabilistic proof of an asymptotic formula for the number of labelled regular graphs, *European Journal of Combinatorics* **1**, 311–316.
- Bornholdt, S. and Schuster, H. G. (eds.), *Handbook of graphs and networks: from the Genome to the Internet* (Wiley-VCH, 2003), 1st edn.

- Bose, S., Vedral, V., and Knight, P. L. (1999), Purification via entanglement swapping and conserved entanglement, *Physical Review A (Atomic, Molecular, and Optical Physics)* **60**, 194.
- Briegel, H. J., Browne, D. E., Dür, W., Raussendorf, R., and Van Den Nest, M. (2009), Measurement-based quantum computation, *Nature Physics* pp. 19–26.
- Briegel, H. J., Dür, W., Cirac, J. I., and Zoller, P. (1998), Quantum Repeaters: The Role of Imperfect Local Operations in Quantum Communication, *Physical Review Letters* **81**, 5932.
- Briegel, H. J. and Raussendorf, R. (2001), Persistent Entanglement in Arrays of Interacting Particles, *Physical Review Letters* **86**, 910–913.
- Broadfoot, S., Dorner, U., and Jaksch, D. (2010a), Long-distance entanglement generation in two-dimensional networks, *Physical Review A (Atomic, Molecular, and Optical Physics)* **82**, 042326.
- (2010b), Singlet generation in mixed-state quantum networks, *Physical Review A (Atomic, Molecular, and Optical Physics)* **81**, 042316.
- Broadfoot, S., Jaksch, D., and Dorner, U. (2009), Entanglement percolation with bipartite mixed states, *Europhysics Letters* **88**, 50002.
- Browne, D. E., Elliott, M. B., Flammia, S. T., Merkel, S. T., Miyake, A., and Short, A. J. (2008), Phase transition of computational power in the resource states for one-way quantum computation, *New Journal of Physics* **10**, 023010.
- Cabello, A., Danielsen, L. E., Lopez-Tarrida, A. J., and Portillo, J. R. (2011), Quantum social networks, *Journal of Physics A: Mathematical and Theoretical* **45**, 285101.
- Callaway, D. S., Newman, M. E. J., Strogatz, S. H., and Watts, D. J. (2000), Network Robustness and Fragility: Percolation on Random Graphs, *Physical Review Letters* **85**, 5468.
- Calsamiglia, J., Hartmann, L., Dür, W., and Briegel, H. J. (2005), Spin Gases: Quantum Entanglement Driven by Classical Kinematics, *Physical Review Letters* **95**, 180502.
- (2007), Entanglement and decoherence in spin gases, *International Journal of Quantum Information* **5**, 509.
- Campbell, E. T. (2007), Distributed quantum-information processing with minimal local resources, *Physical Review A (Atomic, Molecular, and Optical Physics)* **76**.

- Campbell, E. T., Fitzsimons, J., Benjamin, S. C., and Kok, P. (2007a), Adaptive strategies for graph-state growth in the presence of monitored errors, *Physical Review A (Atomic, Molecular, and Optical Physics)* **75**, 042303.
- (2007b), Efficient growth of complex graph states via imperfect path erasure, *New Journal of Physics* **9**, 196–196.
- Carle, T., Kraus, B., Dür, W., and de Vicente, J. I. (2012), Purification to Locally Maximally Entangleable States, arXiv:1208.2553 [quant-ph].
- Catanzaro, M. (2008), *Dynamical Processes in Complex Networks*, Ph.D. thesis, Universitat Politècnica de Catalunya.
- Cho, Y. S., Kim, J. S., Park, J., Kahng, B., and Kim, D. (2009), Percolation Transitions in Scale-Free Networks under the Achlioptas Process, *Physical Review Letters* **103**, 135702.
- Cirac, J. I., Zoller, P., Kimble, H. J., and Mabuchi, H. (1997), Quantum State Transfer and Entanglement Distribution among Distant Nodes in a Quantum Network, *Physical Review Letters* **78**, 3221.
- Cohen, R., Erez, K., Ben-Avraham, D., and Havlin, S. (2000), Resilience of the Internet to Random Breakdowns, *Physical Review Letters* **85**, 4626.
- (2001), Breakdown of the Internet under Intentional Attack, *Physical Review Letters* **86**, 3682–3685.
- da Costa, R., Dorogovtsev, S. N., Goltsev, A. V., and Mendes, J. F. F. (2010), Explosive Percolation Transition is Actually Continuous, *Physical Review Letters* **105**, 255701.
- Cuquet, M. and Calsamiglia, J. (2009), Entanglement Percolation in Quantum Complex Networks, *Physical Review Letters* **103**, 240503–4.
- (2011), Limited-path-length entanglement percolation in quantum complex networks, *Physical Review A (Atomic, Molecular, and Optical Physics)* **83**, 032319–14.
- (2012), Growth of graph states in quantum networks, *Physical Review A (Atomic, Molecular, and Optical Physics)* **86**, 042304.
- De Sola Pool, I. and Kochen, M. (1978), Contacts and Influence, *Social Networks* **1**, 5–51.
- de Solla Price, D. J. (1965), Networks of Scientific Papers, *Science* **149**, 510–515.

- Deutsch, D., Ekert, A. K., Jozsa, R., Macchiavello, C., Popescu, S., and Sanpera, A. (1996), Quantum Privacy Amplification and the Security of Quantum Cryptography over Noisy Channels, *Physical Review Letters* **77**, 2818–2821.
- Dodds, P. S., Muhamad, R., and Watts, D. J. (2003), An experimental study of search in global social networks., *Science* **301**, 827–829.
- Dorogovtsev, S. N., *Lectures on Complex Networks* (Oxford University Press, Oxford, 2010).
- Dorogovtsev, S. N., Goltsev, A. V., and Mendes, J. F. F. (2002), Pseudofractal scale-free web, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **65**, 066122.
- (2008), Critical phenomena in complex networks, *Reviews of Modern Physics* **80**, 1275–61.
- Dorogovtsev, S. N., Mendes, J. F. F., and Samukhin, A. N. (2003), Metric structure of random networks, *Nuclear Physics B* **653**, 307–338.
- Duan, L.-M., Lukin, M. D., Cirac, J. I., and Zoller, P. (2001), Long-distance quantum communication with atomic ensembles and linear optics., *Nature* **414**, 413–8.
- Duan, L.-M. and Monroe, C. (2010), Colloquium: Quantum networks with trapped ions, *Reviews of Modern Physics* **82**, 1209–1224.
- Dür, W., Aschauer, H., and Briegel, H. J. (2003), Multiparticle Entanglement Purification for Graph States, *Physical Review Letters* **91**, 107903.
- Dür, W. and Briegel, H. J. (2007), Entanglement purification and quantum error correction, *Reports on Progress in Physics* **70**, 1381–1424.
- Dür, W., Briegel, H. J., Cirac, J. I., and Zoller, P. (1999), Quantum repeaters based on entanglement purification, *Physical Review A (Atomic, Molecular, and Optical Physics)* **59**, 169.
- Dür, W., Calsamiglia, J., and Briegel, H. J. (2005a), Multipartite secure state distribution, *Physical Review A (Atomic, Molecular, and Optical Physics)* **71**, 042336.
- Dür, W., Hartmann, L., Hein, M., Lewenstein, M., and Briegel, H. J. (2005b), Entanglement in Spin Chains and Lattices with Long-Range Ising-Type Interactions, *Physical Review Letters* **94**, 097203.
- Dür, W., Hein, M., Cirac, J. I., and Briegel, H. J. (2005c), Standard forms of noisy quantum operations via depolarization, *Physical Review A (Atomic, Molecular, and Optical Physics)* **72**, 052326.

- D'Souza, R. M. and Mitzenmacher, M. (2010), Local Cluster Aggregation Models of Explosive Percolation, *Physical Review Letters* **104**, 195702.
- Erdős, P. and Rényi, A. (1959), On random graphs, I, *Publicationes Mathematicae (Debrecen)* **6**, 290–297.
- (1960), On the evolution of random graphs, *Publication of the Mathematical Institute of the Hungarian Academy of Science* **5**, 17–61.
- Euler, L. (1736), Solutio problematis ad geometriam situs pertinentis, *Commentarii Academiae Scientiarum Imperialis Petropolitanae* **8**, 128–140.
- Fedrizzi, A., Ursin, R., Herbst, T., Nespoli, M., Prevedel, R., Scheidl, T., Tiefenbacher, F., Jennewein, T., and Zeilinger, A. (2009), High-fidelity transmission of entanglement over a high-loss free-space channel, *Nature Physics* **5**, 389–392.
- Garnerone, S., Zanardi, P., and Lidar, D. (2012), Adiabatic Quantum Algorithm for Search Engine Ranking, *Physical Review Letters* **108**, 230506.
- Gilbert, E. N. (1959), Random Graphs, *The Annals of Mathematical Statistics* **30**, 1141–1144.
- Gisin, N., Ribordy, G., Tittel, W., and Zbinden, H. (2002), Quantum cryptography, *Reviews of Modern Physics* **74**, 145–195.
- Glancy, S., Knill, E., and Vasconcelos, H. (2006), Entanglement purification of any stabilizer state, *Physical Review A (Atomic, Molecular, and Optical Physics)* **74**, 032319–11.
- Goh, K.-I., Kahng, B., and Kim, D. (2001), Universal Behavior of Load Distribution in Scale-Free Networks, *Physical Review Letters* **87**, 278701.
- Goltsev, A. V., Dorogovtsev, S. N., and Mendes, J. F. F. (2008), Percolation on correlated networks, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **78**, 051105–13.
- Goyal, K., McCauley, A., and Raussendorf, R. (2006), Purification of large bicolorable graph states, *Physical Review A (Atomic, Molecular, and Optical Physics)* **74**, 032318.
- Greenberger, D. M., Horne, M. A., and Zeilinger, A., Going Beyond Bell's Theorem, in M. Kafatos (ed.), *Bells Theorem Quantum Theory and Conceptions of the Universe*, 3, pp. 69–72 (Kluwer, 1989).
- Grimmett, G., *Percolation* (Springer, 1989), 1st edn.
- Gühne, O., Tóth, G., Hyllus, P., and Briegel, H. J. (2005), Bell Inequalities for Graph States, *Physical Review Letters* **95**, 120405.

- Hartmann, L., Calsamiglia, J., Dür, W., and Briegel, H. J. (2005), Spin gases as microscopic models for non-Markovian decoherence, *Physical Review A (Atomic, Molecular, and Optical Physics)* **72**, 052107.
- (2007a), Weighted graph states and applications to spin chains, lattices and gases, *Journal of Physics B: Atomic, Molecular and Optical Physics* **40**, S1.
- Hartmann, L., Kraus, B., Briegel, H. J., and Dür, W. (2007b), Role of memory errors in quantum repeaters, *Physical Review A (Atomic, Molecular, and Optical Physics)* **75**, 032310–17.
- Hein, M., Dür, W., Eisert, J., Raussendorf, R., Van Den Nest, M., and Briegel, H. J., Entanglement in graph states and its applications, in G. Casati, D. L. Shepelyansky, P. Zoller, and G. Benenti (eds.), *Proceedings of the International School of Physics "Enrico Fermi" on Quantum Computers, Algorithms and Chaos*, pp. 115–218 (2006).
- Hein, M., Eisert, J., and Briegel, H. J. (2004), Multiparty entanglement in graph states, *Physical Review A (Atomic, Molecular, and Optical Physics)* **69**, 062311.
- Holland, P. W. and Leinhardt, S. (1981), An Exponential Family of Probability Distributions for Directed Graphs, *Journal of the American Statistical Association* **76**, 33.
- Horodecki, M., Horodecki, P., and Horodecki, R. (1997), Inseparable Two Spin-1/2 Density Matrices Can Be Distilled to a Singlet Form, *Physical Review Letters* **78**, 574–577.
- (1999), General teleportation channel, singlet fraction, and quasidistillation, *Physical Review A (Atomic, Molecular, and Optical Physics)* **60**, 1888–1898.
- (2000), Limits for Entanglement Measures, *Physical Review Letters* **84**, 2014–2017.
- Horodecki, R., Horodecki, M., and Horodecki, K. (2009), Quantum entanglement, *Reviews of Modern Physics* **81**, 865–942.
- Jané, E. (2002), Purification of two-qubit mixed states, *Quantum Information and Computation* **2**, 348.
- Jeong, H., Mason, S. P., Barabási, A.-L., and Oltvai, Z. N. (2001), Lethality and centrality in protein networks, *Nature* **411**, 41–2.
- Johnson, D. B. (1977), Efficient Algorithms for Shortest Paths in Sparse Networks, *Journal of the ACM* **24**, 1–13.

- Joo, J. and L Feder, D. (2011), Edge local complementation for logical cluster states, *New Journal of Physics* **13**, 063025.
- Jungnitsch, B., Moroder, T., and Gühne, O. (2011), Taming Multiparticle Entanglement, *Physical Review Letters* **106**, 190502.
- Kay, A. and Pachos, J. (2007), Multipartite purification protocols: Upper and optimal bounds, *Physical Review A (Atomic, Molecular, and Optical Physics)* **75**, 062307–14.
- Kieling, K., Gross, D., and Eisert, J. (2007a), Cluster state preparation using gates operating at arbitrary success probabilities, *New Journal of Physics* **9**, 200–200.
- Kieling, K., Rudolph, T., and Eisert, J. (2007b), Percolation, Renormalization, and Quantum Computing with Nondeterministic Gates, *Physical Review Letters* **99**.
- Kimble, H. J. (2008), The quantum internet, *Nature* **453**, 1023–1030.
- Klovdahl, A. S., Potterat, J. J., Woodhouse, D. E., Muth, J. B., Muth, S. Q., and Darrow, W. W. (1994), Social networks and infectious disease: the Colorado Springs Study., *Social science medicine* **38**, 79–88.
- Kollár, B., Kiss, T., Novotný, J., and Jex, I. (2012), Asymptotic Dynamics of Coined Quantum Walks on Percolation Graphs, *Physical Review Letters* **108**, 230505.
- Kraus, B. and Cirac, J. I. (2004), Discrete Entanglement Distribution with Squeezed Light, *Physical Review Letters* **92**, 013602.
- Kruszynska, C. and Kraus, B. (2009), Local entanglability and multipartite entanglement, *Physical Review A (Atomic, Molecular, and Optical Physics)* **79**, 052304.
- Kruszynska, C., Miyake, A., Briegel, H. J., and Dür, W. (2006), Entanglement purification protocols for all graph states, *Physical Review A (Atomic, Molecular, and Optical Physics)* **74**, 052316.
- Lapeyre Jr., G. J., Perseguers, S., Lewenstein, M., and Acín, A. (2012), Distribution of entanglement in networks of bi-partite full-rank mixed states, *Quantum Information and Computation* **12**, 0502–0534.
- Lapeyre Jr., G. J., Wehr, J., and Lewenstein, M. (2009), Enhancement of entanglement percolation in quantum networks via lattice transformations, *Physical Review A (Atomic, Molecular, and Optical Physics)* **79**, 042324.
- Lee, C., Lee, J., and Oum, S.-i. (2012), Rank-width of random graphs, *Journal of Graph Theory* **70**, 339–347.

- Lloyd, A. L. (2001), How Viruses Spread Among Computers and People, *Science* **292**, 1316–1317.
- Lo, H.-K. and Popescu, S. (2001), Concentrating entanglement by local actions: Beyond mean values, *Physical Review A (Atomic, Molecular, and Optical Physics)* **63**.
- López, E., Parshani, R., Cohen, R., Carmi, S., and Havlin, S. (2007), Limited path percolation in complex networks, *Physical Review Letters* **99**, 188701.
- Lu, C.-Y., Zhou, X.-Q., Gühne, O., Gao, W.-B., Zhang, J., Yuan, Z.-S., Goebel, A., Yang, T., and Pan, J.-W. (2007), Experimental entanglement of six photons in graph states, *Nature Physics* **3**, 91–95.
- Ma, X.-S., Herbst, T., Scheidl, T., Wang, D., Kropatschek, S., Naylor, W., Wittmann, B., Mech, A., Kofler, J., Anisimova, E., Makarov, V., Jennewein, T., Ursin, R., and Zeilinger, A. (2012), Quantum teleportation over 143 kilometres using active feed-forward, *Nature* **489**, 269–273.
- Matsumoto, M. and Nishimura, T. (1998), Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator, *ACM Transactions on Modeling and Computer Simulation* **8**, 3–30.
- Matsuzaki, Y., Benjamin, S. C., and Fitzsimons, J. (2010), Probabilistic Growth of Large Entangled States with Low Error Accumulation, *Physical Review Letters* **104**, 050501.
- Mattle, K., Weinfurter, H., Kwiat, P. G., and Zeilinger, A. (1996), Dense Coding in Experimental Quantum Communication, *Physical Review Letters* **76**, 4656–4659.
- Mertens, S., Computational Complexity, in D. Bruß and G. Leuchs (eds.), *Lectures on Quantum Information*, chap. 2, pp. 17–36 (Wiley-VCH, 2007).
- Mézard, M., Parisi, G., and Virasoro, M. A., *Spin glass theory and beyond* (World Scientific Publishing, 1987), 1st edn.
- Milgram, S. (1967), The small world problem, *Psychology Today* **1**, 60–67.
- Modlowska, J. and Grudka, A. (2008), Nonmaximally Entangled States Can Be Better for Multiple Linear Optical Teleportation, *Physical Review Letters* **100**, 110503–4.
- Molloy, M. and Reed, B. (1995), A critical point for random graphs with a given degree sequence, *Random Structures and Algorithms* **6**, 161–180.

- Monz, T., Schindler, P., Barreiro, J., Chwalla, M., Nigg, D., Coish, W., Harlander, M., Hänsel, W., Hennrich, M., and Blatt, R. (2011), 14-Qubit Entanglement: Creation and Coherence, *Physical Review Letters* **106**, 130506.
- Moore, C. and Newman, M. E. J. (2000), Exact solution of site and bond percolation on small-world networks, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **62**, 7059–7064.
- Muelken, O. and Blumen, A. (2011), Continuous-Time Quantum Walks: Models for Coherent Transport on Complex Networks, *Physics Reports* **502**, 37–87.
- Newman, M. E. J. (2000), Models of the Small World, *Journal of Statistical Physics* **101**, 819 – 841.
- (2001a), Clustering and preferential attachment in growing networks, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **64**, 025102(R).
- (2001b), Scientific collaboration networks. I. Network construction and fundamental results, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **64**, 016131.
- (2001c), Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **64**, 016132.
- (2001d), The structure of scientific collaboration networks., *Proceedings of the National Academy of Sciences of the United States of America* **98**, 404–409.
- (2002a), Spread of epidemic disease on networks, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **66**, 016128.
- (2002b), The structure and function of networks, *Computer Physics Communications* **147**, 40–45.
- (2003a), Properties of highly clustered networks, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **68**, 026121.
- , Random graphs as models of networks, in S. Bornholdt and H. G. Schuster (eds.), *Handbook of graphs and networks: from the Genome to the Internet*, chap. 2, pp. 35–68 (Wiley-VCH, 2003b), 1st edn.
- (2009), Random Graphs with Clustering, *Physical Review Letters* **103**, 058701–4.
- , *Networks: An Introduction* (Oxford University Press, 2010).
- Newman, M. E. J. and Barkema, G. T., *Monte Carlo Methods in Statistical Physics* (Oxford University Press, 1999).

- Newman, M. E. J., Strogatz, S. H., and Watts, D. J. (2001), Random graphs with arbitrary degree distributions and their applications, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **64**, 026118.
- Newman, M. E. J. and Ziff, R. (2000), Efficient Monte Carlo Algorithm and High-Precision Results for Percolation, *Physical Review Letters* **85**, 4104–4107.
- Nielsen, M. A. (1999), Conditions for a Class of Entanglement Transformations, *Physical Review Letters* **83**, 436–439.
- Nielsen, M. A. and Chuang, I. L., *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).
- Nielsen, M. A. and Vidal, G. (2001), Majorization and the interconversion of bipartite states, *Quantum Information and Computation* **1**, 76–93.
- Ostilli, M. (2012), Cayley Trees and Bethe Lattices: A concise analysis for mathematicians and physicists, *Physica A: Statistical Mechanics and its Applications* **391**, 3417–3423.
- Oum, S.-i. and Seymour, P. (2006), Approximating clique-width and branch-width, *Journal of Combinatorial Theory, Series B* **96**, 514–528.
- Palmer, C. R. and Steffan, J. G., Generating network topologies that obey power laws, in *Global Telecommunications Conference, 2000. GLOBECOM'00. IEEE*, pp. 434–438 (2000).
- Panagiotou, K., Spöhel, R., Steger, A., and Thomas, H. (2011), Explosive Percolation in Erdős-Rényi-Like Random Graph Processes, *Electronic Notes in Discrete Mathematics* **38**, 699–704.
- Paparo, G. D. and Martín-Delgado, M. A. (2012), Google in a quantum network, *Scientific Reports* **2**, 444.
- Pastor-Satorras, R., Vázquez, A., and Vespignani, A. (2001), Dynamical and Correlation Properties of the Internet, *Physical Review Letters* **87**, 258701.
- Perseguers, S. (2010), Fidelity threshold for long-range entanglement in quantum networks, *Physical Review A (Atomic, Molecular, and Optical Physics)* **81**, 012310.
- Perseguers, S., Cavalcanti, D., Lapeyre Jr., G. J., Lewenstein, M., and Acín, A. (2010a), Multipartite entanglement percolation, *Physical Review A (Atomic, Molecular, and Optical Physics)* **81**, 032327.
- Perseguers, S., Cirac, J. I., Acín, A., Lewenstein, M., and Wehr, J. (2008), Entanglement distribution in pure-state quantum networks, *Physical Review A (Atomic, Molecular, and Optical Physics)* **77**, 022308–14.

- Perseguers, S., Lewenstein, M., Acín, A., and Cirac, J. I. (2010b), Quantum random networks, *Nature Physics* **6**, 539–543.
- Plenio, M. B. and Virmani, S. (2007), An introduction to entanglement measures, *Quantum Information and Computation* **7**, 1–51.
- Popescu, S. and Rohrlich, D. (1997), Thermodynamics and the measure of entanglement, *Physical Review A (Atomic, Molecular, and Optical Physics)* **56**, R3319–R3321.
- Popp, M., Verstraete, F., Martín-Delgado, M. A., and Cirac, J. I. (2005), Localizable entanglement, *Physical Review A (Atomic, Molecular, and Optical Physics)* **71**, 042306.
- Radicchi, F. and Fortunato, S. (2009), Explosive Percolation in Scale-Free Networks, *Physical Review Letters* **103**, 168701.
- Raussendorf, R., Bravyi, S., and Harrington, J. (2005), Long-range quantum entanglement in noisy cluster states, *Physical Review A (Atomic, Molecular, and Optical Physics)* **71**, 062313.
- Raussendorf, R. and Briegel, H. J. (2001), A One-Way Quantum Computer, *Physical Review Letters* **86**, 5188–5191.
- Raussendorf, R., Browne, D. E., and Briegel, H. J. (2003), Measurement-based quantum computation on cluster states, *Physical Review A (Atomic, Molecular, and Optical Physics)* **68**, 022312.
- Ravasz, E. and Barabási, A.-L. (2003), Hierarchical organization in complex networks, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **67**, 026112.
- Riordan, O. and Warnke, L. (2011), Explosive percolation is continuous, *Science* **333**, 322–4.
- Ritter, S., Nölleke, C., Hahn, C., Reiserer, A., Neuzner, A., Uphoff, M., Mücke, M., Figueroa, E., Bochmann, J., and Rempe, G. (2012), An elementary quantum network of single atoms in optical cavities, *Nature* **484**, 195–200.
- Rohde, P. P. and Barrett, S. D. (2007), Strategies for the preparation of large cluster states using non-deterministic gates, *New Journal of Physics* **9**, 198–198.
- Sánchez-Burillo, E., Duch, J., Gómez-Gardeñes, J., and Zueco, D. (2012), Quantum Navigation and Ranking in Complex Networks, *Scientific Reports* **2**, 605.

- Sangouard, N., Simon, C., de Riedmatten, H., and Gisin, N. (2011), Quantum repeaters based on atomic ensembles and linear optics, *Reviews of Modern Physics* **83**, 33–80.
- Scala, A., Amaral, L. A. N., and Barthélemy, M. (2001), Small-world networks and the conformation space of a short lattice polymer chain, *Europhysics Letters* **55**, 594–600.
- Schlingemann, D. and Werner, R. F. (2001), Quantum error-correcting codes associated with graphs, *Physical Review A (Atomic, Molecular, and Optical Physics)* **65**, 12308.
- Schneier, B., *Applied Cryptography*, vol. 1 (John Wiley & Sons, 1996), 2nd edn.
- Sen(De), A., Sen, U., Brukner, Č., Bužek, V., and Żukowski, M. (2005), Entanglement swapping of noisy states: A kind of superadditivity in nonclassicality, *Physical Review A (Atomic, Molecular, and Optical Physics)* **72**, 042310.
- Serrano, M. A. and Boguñá, M. (2005), Tuning clustering in random networks with arbitrary degree distributions, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **72**, 036133–8.
- (2006a), Clustering in complex networks. I. General formalism, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **74**, 056114–9.
- (2006b), Clustering in complex networks. II. Percolation properties, *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)* **74**, 056115–8.
- Serrano, M. A., Maguitman, A., Boguñá, M., Fortunato, S., and Vespignani, A. (2007), Decoding the structure of the WWW: A comparative analysis of Web crawls, *ACM Transactions on the Web (TWEB)* **1**, 10.
- Shannon, C. E. (1948), A Mathematical Theory of Communication, *Bell System Technical Journal* **27**, 379–423.
- Shao, J., Buldyrev, S. V., Cohen, R., Kitsak, M., Havlin, S., and Stanley, H. E. (2008), Fractal boundaries of complex networks, *Europhysics Letters* **84**, 48004.
- Siek, J. G., Lee, L.-Q., and Lumsdaine, A., *The Boost Graph Library, C++ In-Depth Series*, vol. 243 (Addison-Wesley, 2002).
- Solomonoff, R. and Rapoport, A. (1951), Connectivity of random nets, *The Bulletin Of Mathematical Biophysics* **13**, 107–117.
- Sporns, O., Tononi, G., and Edelman, G. M. (2000), Theoretical neuroanatomy: relating anatomical and functional connectivity in graphs and cortical connection matrices., *Cerebral Cortex* **10**, 127–141.

- Stauffer, D. and Aharony, A., *Introduction To Percolation Theory* (CRC Press, 1994), 2nd edn.
- Sykes, M. F. and Essam, J. W. (1964), Exact Critical Percolation Probabilities for Site and Bond Problems in Two Dimensions, *Journal of Mathematical Physics* **5**, 1117.
- Travers, J. and Milgram, S. (1969), An Experimental Study of the Small World Problem, *Sociometry* **32**, 425–443.
- Tsomokos, D. (2011), Quantum walks on complex networks with connection instabilities and community structure, *Physical Review A (Atomic, Molecular, and Optical Physics)* **83**, 052315.
- Ursin, R., Tiefenbacher, F., Schmitt-Manderbach, T., Weier, H., Scheidl, T., Lindenthal, M., Blauensteiner, B., Jennewein, T., Perdigues, J., Trojek, P., Ömer, B., Furst, M., Meyenburg, M., Rarity, J., Sodnik, Z., Barbieri, C., Weinfurter, H., and Zeilinger, A. (2007), Entanglement-based quantum communication over 144 km, *Nature Physics* **3**, 481–486.
- Van den Nest, M., Dehaene, J., and De Moor, B. (2004), Graphical description of the action of local Clifford transformations on graph states, *Physical Review A (Atomic, Molecular, and Optical Physics)* **69**, 022316–7.
- Van den Nest, M., Miyake, A., Dür, W., and Briegel, H. J. (2006), Universal Resources for Measurement-Based Quantum Computation, *Physical Review Letters* **97**, 150504.
- Vedral, V., Plenio, M. B., Rippin, M. A., and Knight, P. L. (1997), Quantifying Entanglement, *Physical Review Letters* **78**, 2275–2279.
- Verstraete, F., Popp, M., and Cirac, J. I. (2004), Entanglement versus Correlations in Spin Systems, *Physical Review Letters* **92**, 027901.
- Verstraete, F. and Verschelde, H. (2003), Optimal Teleportation with a Mixed State of Two Qubits, *Physical Review Letters* **90**, 097901.
- de Vicente, J. I., Carle, T., Streitberger, C., and Kraus, B. (2012), Complete Set of Operational Measures for the Characterization of Three-Qubit Entanglement, *Physical Review Letters* **108**, 060501.
- Vidal, G. (1999), Entanglement of Pure States for a Single Copy, *Physical Review Letters* **83**, 1046.
- Watts, D. J. and Strogatz, S. H. (1998), Collective dynamics of ‘small-world’ networks, *Nature* **393**, 440–442.

- White, J. G., Southgate, E., Thomson, J. N., and Brenner, S. (1986), The Structure of the Nervous System of the Nematode *Caenorhabditis elegans*, *Philosophical Transactions of the Royal Society B: Biological Sciences* **314**, 1–340.
- Wilf, H., *Generatingfunctionology* (AK Peters, Ltd., London, 2006), 2nd edn.
- Wu, L. and Zhu, S. (2011), Entanglement percolation on a quantum internet with scale-free and clustering characters, *Physical Review A (Atomic, Molecular, and Optical Physics)* **84**, 052304.
- Yook, S.-H., Jeong, H., and Barabási, A.-L. (2002), Modeling the Internet’s large-scale topology., *Proceedings of the National Academy of Sciences of the United States of America* **99**, 13382–6.
- Zeilinger, A. (1999), A Foundational Principle for Quantum Mechanics, *Foundations of Physics* **29**, 631–643.
- Zwenger, M., Dür, W., and Briegel, H. J. (2012), Measurement-based quantum repeaters, *Physical Review A (Atomic, Molecular, and Optical Physics)* **85**, 062326.
- Żukowski, M., Zeilinger, A., Horne, M. A., and Ekert, A. K. (1993), “Event-ready-detectors” Bell experiment via entanglement swapping, *Physical Review Letters* **71**, 4287.

