Comprendre le monde,
construire l'avenir®

Université Paris-Sud

Ecole Doctorale ED 107
Laboratoire LPTMS

Discipline : Physique Statistique/Theorique

Thèse de doctorat

Soutenue le 08/09/2015 par

Caterina De Bacco

Decentralized network control, optimization and random walks on networks

Directeur de thèse :	Silvio Franz
Co-directeur de thèse :	Satya Majumdar

Composition du jury :

Rapporteurs :	Opper Manfred
	Marsili Matteo
Examinateurs :	Krzakala Florent
	Semerjian Guilhem
	Boguna Marian

Contents

Introduction

In the last years several problems been studied at the interface between statistical physics and computer science. The reason being that often these problems can be reinterpreted in the language of physics of disordered systems, where a big number of variables interacts through local fields dependent on the state of the surrounding neighborhood. The connection between the two domains allows to address various interdisciplinary applications using theories developed in statistical physics, in particular in the context of spin-glass theory. One of the most popular methods developed in this field being the so called *cavity method*. Its popularity is due to the fact that it allows a natural algorithmic formulation that has been proven successful to solve many problems of combinatorial optimisation or constrained-satisfaction.
Among the numerous applications of this type the optimal routing on communication networks is the subject of the first part of the thesis. In this case the objective is to find the optimal way to distribute traffic along the network such that path length and traffic congestion are both minimized, given a set of non-local constraints that the system has to satisfy. Routing communication paths through optimal trajectories would allow the achievement of high quality of transmission because an efficient usage of the resources allows to accommodate a higher number of clients, avoiding at the same time the waste of bandwidth. Moreover, path length minimization implies faster data transmission whereas the congestion minimization make the networks more resilient to failures. To achieve this goal we will exploit the cavity method to formulate efficient algorithms of type message-passing and thus solve the problem through its numerical implementation.
The first routing application that we study is the Node-disjoint path problem, which is characterized by the hard constraint that communications cannot overlap on nodes of the network. For a given number of communications we will determine whether a solutions exists or the system is frustrated, namely the situation where the constraints cannot all be satisfied simultaneously and there will be overlaps at certain nodes.
Next we study the Edge-disjoint path problem, where the previous hard constraint is relaxed: overlaps are allowed at nodes but not along edges. This increases the number of possible path configurations, thus the complexity of the problem. The principle used to solve this problem is to map it into another well known combinatorial optimisation problem, namely the maximum weighted matching on graphs. The objective will be to accommodate the largest number of communications. The results obtained through our algorithm are compared with the ones obtained through other types of algorithms representing the *state-of-the-art* and proposed by engineers and computer scientists on several benchmark instances. Compared to these we have obtained improved performances with respect to the number of accommodated paths and our approach also minimizes the total path length. Remarkably we found good performances also in the case of loopy graphs, where the hypothesis behind the cavity method are usually not satisfied. This achievement has been made possible by implementing a numerical variant to the standard message-passing algorithm which forces the message-passing equations to reach convergence, though towards suboptimal solutions.
We conclude the first part of the thesis by considering the final generalization of the problem: overlap are now allowed but minimized, along with total path length. The goal will be to

determine the Nash equilibrium of the problem, i.e. the stable situation where none of the communications can improve its situation by modifying its path, given the others remain the same. The absence of traffic constraints leads to an exponential complexity in the number of communications, thus preventing a direct numerical implementation. Therefore we will address the problem by considering a mean-field model where each communication makes its decision based on its mean field estimation of the neighborhood configuration.
Next we will address the dynamic version of the cavity method, a model that can be used to study dynamical processes on networks. This has only been recently developed and thus is far from being completely mature to be implemented numerically. The major problem is given by the exponential complexity that characterizes non-Markovian processes: each cavity message is a function of a time trajectory and in general there is an exponential number of possible ones. This is the main reason why its implementation has been limited to either unidirectional dynamics or to processes that are in a stationary state, so that certain types of approximations can be applied to lower the complexity. In this part of the thesis we will describe a method that we have introduced in order to lower the complexity from exponential to polynomial but without making any hypothesis about the reversibility of the dynamic nor on the network directionality. The main idea is to borrow the formalism developed in quantum mechanics to describe a many-particle quantum state in the form of a Matrix Product State representation and represent the dynamic cavity message using a similar decomposition; thus allowing an approximation of the equation with lower complexity. We will test the performance of this model using Glauber dynamics with disorder and the majority rule on random regular graphs.
Another topic that has attracted much interest in statistical physics of dynamic processes is the random walk on networks. The theory has been developed since many years in the case the underneath topology is a d-dimensional lattice. On the contrary the case of random networks has been tackled only in the past decade, leaving many questions still open for answers. In particular, with the fast development of communication networks such as the internet, the random walk is being intensively used in all its variants for instance to route packets of data along the network. Revealing and capturing some aspects related to this stochastic process will be the objective of the second part of the thesis. In particular we will study the average number of distinct sites visited by a random walker on a random topology, with the goal of characterizing how, in the long-time limit, its linear behavior depends on the topology. To address this issue we will reformulate the problem using an auxiliary system made of random variables jointly distributed through a multivariate Gaussian distribution, their variances being directly related to our quantity of interest. Hence we will propose a set of cavity equations to calculate these variances for different graph topologies and thus to iteratively access the information about the average number of distinct visited sites.
Our discussion will proceed by addressing the problem of characterizing rare events associated to random walks on networks. This will be done by using large deviations theory with the idea of transforming rare events under the standard unbiased walk to typical ones under a deformed transition matrix. A study of the spectral properties of the latter will be performed, in particular to analyze the rare events statistics corresponding to the degree of the nodes visited during the random walk. The results of numerical simulations on random graphs have highlighted the presence of two dynamic phase transitions, thus we will conclude this part be analyzing the properties of these.
As a last chapter we will outline a work that is not linked to either the previous topics but that nonetheless has given new insights in the context of out-of-equilibrium physics. In particular we will study a system made of two mesoscopic particles surrounded by a thermal bath. This study allowed us to unveil the properties of a bath-mediated potential acting on the two mesoscopic particles arising for the presence of the bath.

Résumé

Différents problèmes à l'interface entre physique statistique et informatique théorique ont été étudiés ces dernières années. Il s'avère que ces problèmes peuvent se reformuler dans le language de la physique de systèmes désordonnés où un grand nombre de variables interagissent localement. Cette connection entre les deux domaines offre la possibilité de les analyser en s'inspirant de théories avancées de la physique statistique, notamment celles développées dans le contexte des verres de spin comme le méthode de la cavité. En particulier cela donne lieu à des formulations algorithmiques qui se sont révélées très efficaces dans la résolution des problèmes d'optimisation sur réseaux.
Parmi les applications de ce type, le problème de routage optimal des chemins sur un réseau de communication est le sujet de la premiere partie de cette thèse. Il s'agit de distribuer de manière efficace le traffic sur un réseau alors que le système est sujet à des contraintes non locales. L'objectif est d'améliorer la qualité de transmission étant donné que le nombre de clients desservis par un routage optimal sera plus élevé, tout en évitant le gaspillage de bande. Aussi, en minimisant la longueur des chemins on pourrait réduire le délai de transmission et abaisser l'impact d'échec d'une partie du réseau en distribuant le trafic sur plusieures jonctions. Pour atteindre ce but on utilisera le méthode de la cavité qui permettra de formuler des algorithmes efficaces dans la résolution numérique du problème.
D'abord on considérera un problème où les contraintes sont strictes, c'est-à-dire que les chemins ne peuvent pas être en collision sur une même jonction. En fonction du nombre de chemins, la question que l'on abordera sera : existe-t-il une solution ou le système est-il dans une situation de frustration ? C'est-à-dire un scénario où il n'est pas possible de respecter toutes les contraintes et où les collisions sont inévitables.
Ensuite on relaxera les contraintes, en autorisant les collisions sur la même jonction mais pas sur le même lien entre deux jonctions. Cela conduira à augmenter l'espace des configurations possibles des chemins, donc la complexité du problème. Le principe utilisé pour apporter une solution à cette deuxième application sera de la reformuler comme un autre problème d'optimisation combinatoire, notamment le problème de *matching* sur réseau. La question que l'on abordera dans ce deuxième cas sera : combien de chemins peut-on ajouter avant de créer une situation de frustration ?
Les résultats obtenus avec notre algorithme ont été comparés avec ceux provenant d'autres types d'algorithmes *état de l'art* développés par des ingénieurs et des informaticiens à partir de benchmarks disponibles dans la littérature. Nous avons obtenu des performances supérieures à celles de ces derniers en ce qui concerne la quantité de chemins desservis, en minimisant aussi leur longueur.
Enfin on introduira le problème plus general du routage, c'est-à-dire que les collisions seront permises mais l'objectif est de les minimiser, de même que la longueur des chemins. La question que l'on abordera sera de trouver un routage qui correspond à l'*Équilibre de Nash*: le scénario stable où chaque chemin ne peut améliorer sa condition en modifiant sa trajectoire si les autres restent fixes.
L'absence des contraintes de collision impliquant une complexité exponentielle dans le nombre des chemins, il s'agira donc d'adopter une formulation de champs moyen pour pouvoir ap-

procher ce problème.
Dans en deuxième temps on considérera des problèmes de dynamique sur réseaux en utilisant la méthode de la cavité. Celle-ci a été récemment proposée dans sa version dynamique, notamment pour résoudre des problèmes d'inférence dynamique d'épidémies. La complexité du problème est de nouveau exponentielle, mais cette fois par rapport à la variable temps, car le nombre de trajectoires possibles augmente très vite avec la durée du processus. Pour cette raison les problèmes analysés jusqu'à présent avaient une dynamique unidirectionnelle ou étaient considérés dans le cas stationnaire. Il était nécessaire de développer la théorie de manière à traiter de problèmes plus généraux. Nous avons donc formulé une approximation des equations de cavité en utilisant la théorie récemment développée des *Matrix Product States* (MPS), employée en physique statistique quantique, pour obtenir des équations approchées avec complexité polynomiale. L'efficacité des MPS dans la description des chaines de spin laisse bien espérer le succès de ce programme dans notre cas. On donnera des exemples qui montreront la qualité de cette approximation sur une topologie régulière, en particulier on simulera des processus de type dynamique de *Glauber* sur des systèmes désordonnés et du type *règle de la majorité*.

Un autre domaine de recherche, étudié depuis longtemps en physique statistique, est la marche aléatoire. Plusieurs résultats ont été produits dans les cas d'une topologie régulière en d dimensions. Cette théorie n'est cependant pas complètement mature dans les cas où la promenade est effectuée sur un réseau aléatoire. En revanche c'est dans ce cas là que l'on trouve le plus d'applications, notamment dans le contexte des réseaux de communication comme internet, où les données d'information sont souvant transmises par des marches aléatoires. Mieux comprendre ce problème est l'objectif de la deuxième partie de cette thèse.
Dans un premier temps on abordera la question de savoir combien de sites différents la marche visite-t-elle en moyenne dans son chemin. Comprendre comment cette quantité change avec la topologie aléatoire du réseau est l'objectif de cette partie de thèse. Pour atteindre ce but on reformulera le problème dans un système auxiliaire où les variables suivent une distribution Gaussienne multi-dimensionnelle. Cela permettra de lier les variances des ces variables auxiliaires avec notre quantité d'intérêt. La méthode de cavité nous permettra de les évaluer selon la topologie du réseau et donc de résoudre le problème.
Ensuite on étudiera la statistique des événements rares des quantités associées à la marche aléatoire sur réseau. Pour cela on utilisera la théorie des grandes déviations qui nous permettra de transformer les événements rares selon la matrice de transition standard en événements typiques suivant une matrice de transition modifiée. On étudiera donc les propriétés spectrales de cette matrice déformée pour analyser notamment la statistique de degré des sites visités dans une marche aléatoire. Les résultats des simulations numériques nous ont montré l'existence de deux types de transition de phase dynamique. On terminera donc par l'analyse des propriétés associées à ces deux transitions.

Contents outline

The organization of this thesis is the following.
We will begin by describing in Chapter 1 the cavity method and deriving the corresponding cavity equations in the finite temperature case, as well as the zero temperature limit used to describe optimisation problems. In the first part we will address the static case, whereas in the second one the dynamic version of the problem. The outcomes of this analysis will be extensively used throughout the rest of the thesis.
In Chapter 2 we study three different applications of the static cavity method in the zero temperature limit. These will concern variants of routing optimisation on networks starting with the Node-disjoint path problem, followed by Edge-disjoint path problem and eventually the Nash equilibrium version will be outlined.
Chapter 3 is devoted to the description of the Matrix Product State representation of the dynamic cavity equations. After a general introduction of its usage in the context of quantum mechanics, we will discuss its functioning, its validity and how it improves the *state-of-the-art* approximations of the dynamic message-passing. Results of numerical simulations of disordered Glauber dynamics and majority rule on random regular graphs will be shown.
We turn to study the properties of random walks on networks in Chapter 4, which starts with a general introduction of the formalism of this stochastic process in the case of random graphs. The rest of the chapter is divided in two parts. The methodology used to characterize the average number of distinct sites visited during a random walk on random networks is presented in the first one. We will discuss how this problem can be solved by mapping it into a system of jointly distributed random variables following a multivariate Gaussian distribution. The rare event statistics of random walks on networks will be the main topic of the second part of the chapter. This includes the application of large deviation theory that allows to formalize this problem in terms of a deformed transition matrix. The analysis of its spectral properties along with the treatment of the two dynamic phase transitions unveiled by the numerical simulations will conclude this chapter.
The final Chapter 5 treats the independent topic of an out-of-equilibrium system of two mesoscopic particles surrounded by a thermal bath. The model description along with two solvable examples will be given. The outcomes of these examples will support the validity of a relation we claim to exist between the friction kernel of memory and the bath-mediated potential acting on the two particles arising for the presence of the bath.
Three appendices will complete the manuscript with additional information and details. Appendix 1 describes the reinforcement technique used to force convergence of the message-passing equations, a tool used in section 2.2.1. Appendix 2 is devoted to the an extensive description of the operations performed on tensors supporting section 3.2. It is useful in particular for writing the code of the algorithm for numerical implementation. A detailed description of the various random network topologies used in the main text will be given in appendix 3.

Chapter 1

The cavity method

In the last decade several problems of combinatorial optimisation and constrained satisfaction have been successfully modeled by statistical physicists using an iterative method introduced for the first time in the context of spin glasses [1] which goes under the name of *cavity method*. The original method has then been further developed to model the zero-temperature case [2] and to deal with the hard regime of metastable states [3, 4]. This method enables the computation of marginal probabilities of interacting variables lying on graphs, using only local computations. It is exact on trees and represents a valid approximation for locally tree-like structures as is the case of large sparse random networks. This method has been shown [5, 6] to be equivalent to the message-passing algorithm used in computer science called Belief Propagation [7]. Its distributive character makes it suited to be implemented numerically and allows the treatment of single instances of graphs.
In the following we will describe in more details how it works and under what conditions it is accurate. As a first analysis in section 1.1 we will address the static case starting at finite temperatures than showing how it adapts to treat the zero-temperature limit; the latter case being the one of interest when studying optimisation problems such as the ones studied in chapter 2.
In a second moment, in section 1.2, we will modify the general cavity formalism introduced to model the static case to study dynamic processes on networks. This reformulation of the method has been only recently introduced and thus requires further studies in order to develop an efficient algorithmic implementation and to asses its performance. This will be the subject of chapter 3.

1.1 Static case

The static version of the cavity method is used to describe the probability distribution of a system where the variables lie on a graph and are not time dependent. One exploits the graphical model representation of the system to derive information about its statical properties. Examples are marginal probabilities of a subset of variables, averages of observables or optimal configurations in the optimisation version of the problem. To achieve this goal, given a set of initial conditions, one has to iterate the self-consistent cavity equations until convergence; if this is reached, the converged messages can then be collected and suitably used to derive these quantities of interest.
The general model is specified by a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ characterized by $V = |\mathcal{V}|$ nodes (or variable nodes), $E = |\mathcal{E}|$ edges and $F = |\mathcal{F}|$ interaction terms (or function nodes). We denote the neighborhood of a variable node $i \in \mathcal{V}$ as ∂i and of a function node $a \in \mathcal{F}$ as ∂a. Given a variable σ_i lying on a node $i \in \mathcal{V}$ and an interaction term $\psi_a(\sigma_{\partial a})$, which is function of a set of variables $\sigma_{\partial a} = \{\sigma_{a1}, \dots, \sigma_{a|\partial a|}\}$ and lies on $a \in \mathcal{F}$, an edge (i, a) exists in $\mathcal{E}$ if ψ_a

involves the variable σ_i, i.e. if $\sigma_i \in \sigma_{\partial a}$. It is often convenient to represent it using a factor graph diagram as in figure 1.1 where is depicted an example of a graph with three interaction terms and four node variables.

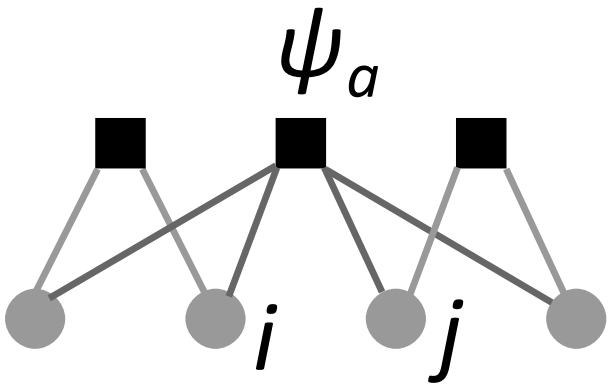


Figure 1.1: Factor graph. Squares represent function nodes $a \in \mathcal{F}$, circles represent variable nodes $i \in \mathcal{V}$. In this example the central interaction ψ_a involves all the four variable nodes, wheres the other two interactions are functions of only two variables.

In general the graph could be either directed or undirected depending on how one defines edges and interaction terms. In this work we focus on the undirected graphs, however one can apply all the following considerations to directed topologies as well.
The joint probability of a vector of variables $\bar{\sigma} = (\sigma_1, \dots, \sigma_V)$ lying on a graph is in general written as:

$$P(\bar{\sigma}) = \frac{1}{Z} \prod_{a \in \mathcal{F}} \psi_a(\sigma_{\partial a}) \tag{1.1}$$

where the factors ψ_a represent local interaction terms introduced above. The simplest example is the pairwise model where variables interact pair-by-pair and $\psi_a(\sigma_{\partial a})$ can be written as $\psi_{ij}(\sigma_i, \sigma_j)$.
The peculiarity of the expression (1.1) lies on its factorized character. This features allows to treat complex global functions using only local information involving the different function nodes. Starting from this consideration, one can then naturally set up an algorithmic routine to exploit numerically this property; a general overview of different algorithms applied on factor graphs can be found in [8].
The graphical structure allows in particular to efficiently derive the marginal probability of a variable node $i \in \mathcal{V}$. The marginal probability of a variable σ_i lying on $i \in \mathcal{V}$ is defined as:

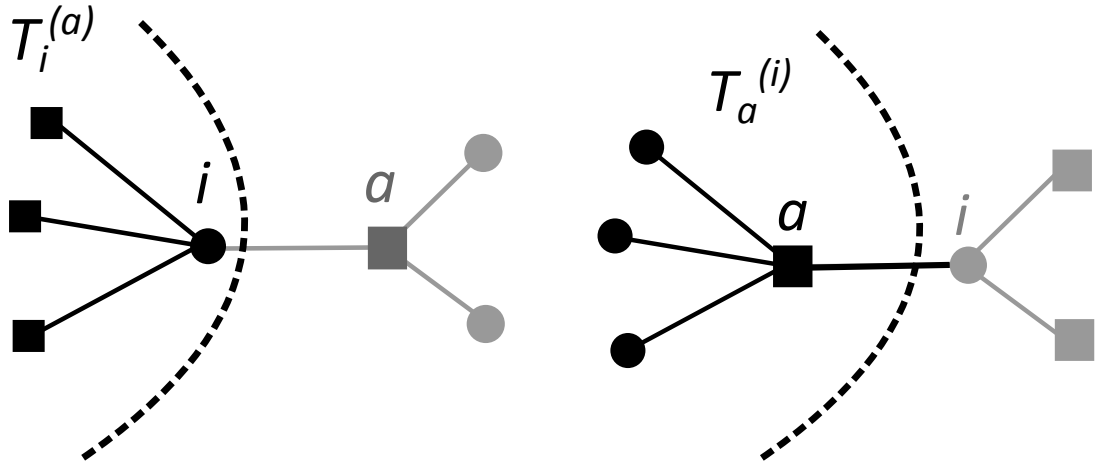
$$P_i(\sigma_i) = \sum_{\bar{\sigma} \setminus i} P(\bar{\sigma}) \tag{1.2}$$

where $\bar{\sigma} \setminus i = \{\sigma_j : j \in \mathcal{V} \setminus \{i\}\}$ is the whole set of variable nodes excluding i. The general idea behind the cavity method is to derive an expression for (1.2) which is exact on trees and is a valid approximation for locally tree-like structures; we now describe how the cavity method does this.
Suppose for a moment that our graph $\mathcal{G}$ is a tree, i.e. there are no loops. If one removes from it a node i, and the corresponding edges (ia) where $a \in \partial i$, then the original tree is modified into a graph $\mathcal{G}^i$ which is made of $|\partial i|$ independent subtrees $T_a^{(i)}$ rooted at $a \in \partial i$. Focusing on one such subtree $T_a^{(i)}$, suppose now that only the edge (ai) is reintroduced. In figure 1.2(b) we represent this situation. We can define as $\hat{m}_{a \to i}(\sigma_i)$ the marginal probability of σ_i in this subtree. Using this definition one can write the marginal probability of σ_i in the original tree

as:

$$P_i(\sigma_i) = \frac{1}{Z_i} \prod_{a \in \partial i} \hat{m}_{a \to i}(\sigma_i) \tag{1.3}$$

where Z_i is a normalization factor.



(a) (b)

Figure 1.2: Cavity graphs. Squares represent function nodes a, circles represent variable nodes i. The two subtrees generated by the cavity removal are depicted.

For the moment we only gave a definition of $\hat{m}_{a \to i}(\sigma_i)$, but now we want to formally characterize it as a function of the interaction terms.
Suppose now to repeat a similar graph modification where this time we only remove the edge (ia) and focus on the subtree $T_i^{(a)}$ rooted at i of the original graph $\mathcal{G}$. In figure 1.2(a) we give a pictorial representation of this sub-graph. In a similar way we define as $m_{i \to a}(\sigma_i)$ the marginal probability of σ_i in this subtree where (ia) is not present. From these two definitions we can notice that: $m_{i \to a}(\sigma_i)$ in $T_i^{(a)}$ depends only on all the other function nodes different from a; $\hat{m}_{a \to i}(\sigma_i)$ in $T_a^{(i)}$ depends on the function node $\psi_a(\bar{\sigma}_{\partial a})$ and on the cavity marginals $m_{j \to a}(\sigma_j)$ of the neighboring variable nodes in $T_a^{(i)}$. Taking these facts into account finally we can write the set of two self-consistent equations:

$$m_{i \to a}^{(t)}(\sigma_i) = \frac{1}{Z_{i \to a}} \prod_{b \in \partial i \setminus a} \hat{m}_{b \to i}^{(t)}(\sigma_i) \tag{1.4}$$

$$\hat{m}_{a \to i}^{(t+1)}(\sigma_i) = \frac{1}{Z_{a \to i}} \sum_{\sigma_{\partial a \setminus i}} \psi_a(\sigma_{\partial a}) \prod_{j \in \partial a \setminus i} m_{j \to a}^{(t)}(\sigma_j) \tag{1.5}$$

where we introduced a superscript (t) denoting the iteration time step. We call these equations *cavity* or *message-passing* equations and the quantities $m_{i \to a}(\sigma_i)$ and $\hat{m}_{a \to i}(\sigma_i)$ *cavity messages*. These equations can be iterated numerically until convergence and the introduction of the superscript (t) aims at helping visualizing this fact. The algorithmic implementation is often called Belief Propagation or *sum-product* algorithm. If convergence is reached then eventually one collects the cavity marginals and use them to calculate the original graph marginals using (1.3). If the graph is a tree then they will return the exact marginals, otherwise they can still be applied on locally tree-like structures for which they will represent a valid approximation. The validity of this approximation relies on the fact that correlations among neighbors of a given node decay exponentially fast, and this is due to the absence of short loops. On the contrary, their presence would forbid the factorization (1.3) because neighbors will not be independent

anymore. For sparse graphs, i.e. graphs where the number of edges E is much smaller than the maximum number of edges $V(V-1)/2$, loop length goes as $\log V$. This means that in the thermodynamic limit $V \to \infty$ the message-passing algorithm gives asymptotically exact results because short loops are absent. In case of loopy graphs usually convergence is not reached; one can nonetheless introduce corrections to take into account correlations between the neighbors of a given node as is done in [9].
We can summarize the arguments made above by describing the general algorithmic routine as follows:

- Initialize messages $m_{i\to a}(\sigma_i)$, $\hat{m}_{a\to i}(\sigma_i)$ at random (for instance as a uniform distribution)
- Pick arbitrarily a variable node $i \in \mathcal{V}$ and update using (1.4)
- Repeat for all other variable nodes
- Pick arbitrarily a function node $a \in \mathcal{F}$ and update using (1.5)
- Repeat for all other function nodes
- Repeat the whole procedure until $m_{i\to a}(\sigma_i)$ and $\hat{m}_{a\to i}(\sigma_i)$ converge

Here for sake of simplicity we used a parallel update for the messages but other types of update rules are possible.
We conclude this section by writing a more compact version of the cavity equations that we are going to use in the rest of the thesis. The principle is to write one of the two types of messages as a function of the other, thus unifying the two equations in a single one. It is particularly convenient to use it in case of variable nodes staying on edges. In this case the variables have at most three neighboring function nodes: two lying on the extremal nodes of the edge and in general one external field acting on the edge variable (this can also be absent). Then one only needs one type of message:

$$m_{a\to i}^{(t+1)}(\sigma_i) = \frac{1}{Z_{\to i}} \sum_{\{\sigma_{\partial a\setminus i}\}} \psi_a(\sigma_{\partial a}) \prod_{k\in\partial a\setminus i} \prod_{b\in\partial k\setminus a} m_{b\to k}^{(t)}(\sigma_k) \tag{1.6}$$

where the normalization factor becomes $Z_{\to i} = Z_{a\to i} \prod_{k\in\partial a\setminus i} Z_{k\to a}$.
A final simplification is obtained in the case of pairwise models, for which the function nodes depend on exactly two variables: $\psi_a(\sigma_{\partial a}) = \psi_{ij}(x_i, x_j)$. In this case the joint probability (1.1) can be written as:

$$P(\bar{\sigma}) = \frac{1}{Z} \prod_{(ij)\in\mathcal{E}} \psi_{ij}(\sigma_i, \sigma_j) \tag{1.7}$$

and from this it follows that the message-passing equations are functions of only one type of message and the order between sum and product could be inverted:

$$m_{i\to j}^{(t+1)}(\sigma_i) = \frac{1}{Z_{i\to j}} \prod_{k\in\partial i\setminus j} \sum_{\{\sigma_k\}} \psi_{ki}(\sigma_k, \sigma_i)\, m_{k\to i}^{(t)}(\sigma_k) \tag{1.8}$$

1.1.1 The optimisation version: the *min-sum* algorithm

Instead of being interested on the values of marginals $P_i(\sigma_i)$ one can ask what is the most likely configuration. If one associates a cost function $E(\bar{\sigma})$ to a given configuration $\bar{\sigma}$ then the most likely configuration, or the optimal one, is the one that minimizes such cost. With this formulation one can set up the optimisation version of the problem.
The two key ingredients needed to specify this type of problem are the cost function and the

constraints, whereas the quantities of interest are usually the optimal configuration and the optimal cost. The optimisation version of the cavity equations (1.4) and (1.5) can be obtained by writing the mapping:

$$\psi_a(\sigma_{\partial a}) = e^{-\beta E_a(\sigma_{\partial a})} \tag{1.9}$$

where β is a parameter that can be tuned as it is usually done in the statistical physics context where it is interpreted as an inverse temperature. Substituting this mapping into (1.4) and (1.5) and taking the logarithm we have that products are transformed into sums. The optimal configuration is obtained by minimizing instead of performing the marginalization (the sum over the neighborhood configuration). In the statistical physics framework this can be translated by considering the zero temperature limit (or $\beta \to \infty$) of the problem. This approach of considering the zero-temperature limit of the cavity equations has been introduced in [10].
Applying all the previous considerations leads to the so called *min-sum* [6] equations:

$$E^{(t)}_{i\to a}(\sigma_i) = \sum_{b\in\partial i\setminus a} \hat{E}^{(t)}_{b\to i}(\sigma_i) + C^{(t)}_{i\to a} \tag{1.10}$$

$$\hat{E}^{(t+1)}_{a\to i}(\sigma_i) = \min_{\{\sigma_{\partial a\setminus i}\}} \left\{ E_a(\sigma_{\partial a}) + \sum_{j\in\partial a\setminus i} E^{(t)}_{j\to a}(\sigma_j) \right\} + \hat{C}^{(t)}_{a\to i} \tag{1.11}$$

Here the constants $C^{(t)}_{i\to a}$ and $\hat{C}^{(t)}_{a\to i}$ are the counterparts of the normalization factors $Z_{i\to a}$ and $Z_{a\to i}$ appearing in (1.4) and (1.5). These constants can be arbitrarily chosen at each iteration step; for instance once can fix them so that $\min_{\sigma_i} \hat{E}^{(t+1)}_{a\to i}(\sigma_i) = 0$ and $\min_{\sigma_i} E^{(t)}_{i\to a}(\sigma_i) = 0$. One then proceeds in the same way as for the *sum-product* by iterating the messages $E_{i\to a}(\sigma_i)$ and $\hat{E}^{(t+1)}_{a\to i}(\sigma_i)$ until convergence. If this is reached, the converged messages $E^*_{i\to a}$ are used to calculate the minimal cost E_{min} and the optimal configuration $\sigma^*_{\partial a}$:

$$E_{min} = \sum_a E_a(\sigma^*_{\partial a}) \tag{1.12}$$

$$\sigma^*_{\partial a} = \arg\min_{\sigma_{\partial a}} \left\{ E_a(\sigma_{\partial a}) + \sum_{i\in\partial a} E^*_{i\to a}(\sigma_i) \right\} \tag{1.13}$$

Also in this case we can make the notation more compact by eliminating one of the two equations and using only one type of message:

$$E^{(t+1)}_{i\to a}(\sigma_i) = \sum_{b\in\partial i\setminus a} \min_{\sigma_{\partial b\setminus i}} \left\{ E_b(\sigma_{\partial b}) + \sum_{j\in\partial b\setminus i} E^{(t)}_{j\to b}(\sigma_j) \right\} + C^{(t)}_{\to i} \tag{1.14}$$

where the new additive constant is $C^{(t)}_{\to i} = C^{(t)}_{i\to a} + \sum_{b\in\partial i\setminus a} \hat{C}^{(t)}_{b\to i}$. Throughout the rest of the thesis we will use this compact notation.

1.2 Dynamic case

The cavity method presented in the previous section can be adapted to describe dynamic processes on networks. In fact, if variables evolve in time, one could describe the time-dependent joint probability distribution using the dynamic version of the message-passing equations (1.4) and (1.5). We will refer to them as dynamic cavity method. The reasoning behind it is similar to the one of the static case but this time there is one important issue that one has to consider as we will see in what follows: the factor graph representation is full of loops, even if the underlying topology is a tree.
As in the static model, we start by writing the joint probability distribution of a set of N variables $\bar{\sigma}^t = (\bar{\sigma}^t_1, \ldots, \bar{\sigma}^t_N)$. This time they are time-dependent, so the entry $\bar{\sigma}^t_i = (\sigma^0_i, \ldots, \sigma^t_i)$

is a t-dimensional vector denoting the trajectory in time of the variable σ_i located on node i. The entry σ_i^t denotes the status of the variable on node i at the instant time step t. Therefore the configuration $\bar{\sigma}^t$ is a Nt-dimensional vector made of all the trajectories of the N variables. We denote with $W(\sigma_i^{t+1}|\sigma_i^t, \{\sigma_j^t\}_{j\in\partial i})$ the transition probability that variable σ_i takes value σ_i^{t+1} at time $t+1$ given the configuration $(\sigma_i^t, \{\sigma_j^t\}_{j\in\partial i})$ at the previous time step t. We then define $P_0(\{\sigma_i^0\})$ (with $i = 1, \ldots, N$) the probability distribution of the configuration at the initial time step $t = 0$. These two ingredients are all what is needed to specify the joint probability distribution of a configuration of N trajectories:

$$P(\bar{\sigma}^t) = P_0(\{\sigma_i^0\}) \prod_{i=1}^{N} \prod_{s=0}^{t-1} W(\sigma_i^{s+1}|\sigma_i^s, \{\sigma_j^s\}_{j\in\partial i}) \tag{1.15}$$

This expression resembles the static case factor graph representation (1.1) if one considers the products $\prod_{s=0}^{t-1} W(\sigma_i^{s+1}|\sigma_i^s, \{\sigma_j^s\}_{j\in\partial i})$ as the function nodes, i.e. the interaction terms. We denote the graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $V = |\mathcal{V}|$ nodes and $E = |\mathcal{E}|$ edges. This time though the situation is more complex because the function nodes involve variables at different times steps, with the consequence that the corresponding factor graph representation is full of short loops. The situation is depicted in figure 1.3.

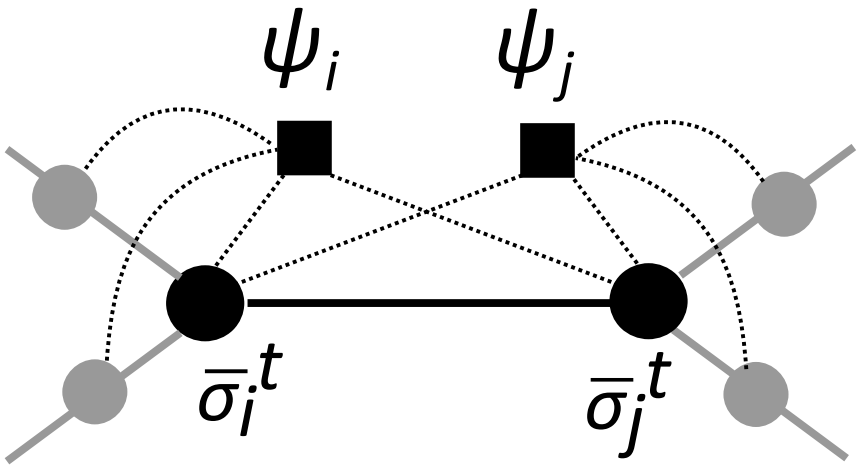


Figure 1.3: Loopy factor graph representation. Squares represent function nodes $\psi_i = \prod_{s=0}^{t-1} W(\sigma_i^{s+1}|\sigma_i^s, \{\sigma_j^s\}_{j\in\partial i})$. Circles represent variable nodes $\bar{\sigma}_i^t$. Notice the presence of short loops, even though the underlying graph is a tree.

This issue is overcome by taking as variable nodes pairs of variables $(\bar{\sigma}_i^t; \bar{\sigma}_j^t)$ an putting them on the edges of the factor graph. The function nodes, the transition probabilities, will be put on nodes $i \in \mathcal{V}$ instead. This is represented in figure 1.4.

Also in this case we are interested to calculate quantities such as marginals of a single or multiple variables so that these could then be used to calculate the average values of observables. Given that variable nodes have always exactly two function nodes as neighbors, the cavity equations corresponding to (1.4) and (1.5) can be written using only one type of message. We could repeat considerations similar to the static case but this time instead of calculating directly the marginal of a single trajectory $P_i(\bar{\sigma}_i^t)$, the form of the variable node allows us to first factorize the 2-variable joint probability $P_{ij}(\bar{\sigma}_i^t, \bar{\sigma}_j^t)$ as a product of messages:

$$P_{ij}(\bar{\sigma}_i^t, \bar{\sigma}_j^t) = \mu_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1})\mu_{ji}(\bar{\sigma}_j^t|\bar{\sigma}_i^{t-1}) \tag{1.16}$$

where the message $\mu_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1})$ represents the marginal probability of the single trajectory $\bar{\sigma}_i^t$ in a cavity graph where the neighbor trajectory $\bar{\sigma}_j^{t-1}$ has been fixed up to time $t-1$ and the function node in j has been removed. This equation is exact for trees and it is a valid

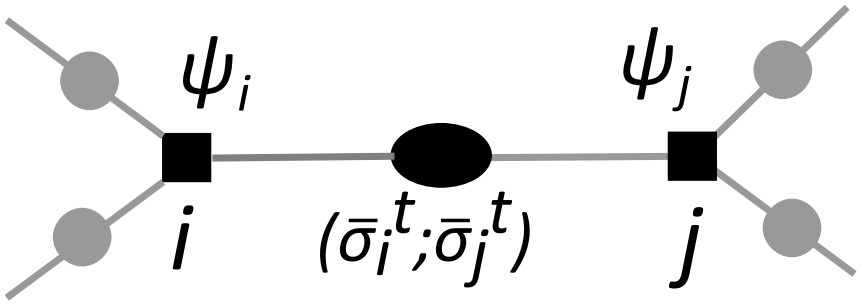


Figure 1.4: Dynamic factor graph representation. Squares represent function nodes $\psi_i = \prod_{s=0}^{t-1} W(\sigma_i^{s+1}|\sigma_i^s, \{\sigma_j^s\}_{j\in\partial i})$. Circles represent variable nodes $(\bar{\sigma}_i^t; \bar{\sigma}_j^t)$. This representation using pairs of variables on edges allows to avoid short loops.

approximation for locally tree-like structures.
The messages so defined can be iteratively derived by keeping the cavity removal along the neighbors, with a procedure similar to the static case. At the end we could write the following cavity iteration:

$$\mu_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1}) = \frac{1}{Z_{ij}} P_i(\sigma_i^0) \sum_{\{\bar{\sigma}_k^{t-1}\}} \prod_{s=0}^{t-1} W(\sigma_i^{s+1}|\sigma_i^s, \{\sigma_j^s\}_{j\in\partial i}) \prod_{k\in\partial i\setminus j} \mu_{ki}(\bar{\sigma}_k^{t-1}|\bar{\sigma}_i^{t-2}) \tag{1.17}$$

Here for the sake of simplicity we have assumed a factorized initial distribution $P_0(\{\sigma_i^0\}) = \prod_{j=1}^{V} P_i(\sigma_i^0)$ and will adopt this choice for the rest of the thesis; the Z_{ij} is a normalization term.

One important remark is that the message $\mu_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1})$ should not be confused with the conditional probability $P_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1})$ in the original graph. In fact the former is a quantity defined in a modified graph (the cavity graph) whereas the latter is defined by the formula $P_{ij}(\bar{\sigma}_i^t, \bar{\sigma}_j^{t-1}) = P_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1})P_j(\bar{\sigma}_j^{t-1})$.

From (1.16) one could obtain the 1-variable marginals by summing out the other variable:

$$P_i(\bar{\sigma}_i^t) = \sum_{\bar{\sigma}_j^t} P_{ij}(\bar{\sigma}_i^t, \bar{\sigma}_j^t) \tag{1.18}$$

Notice that substituting (1.16) into (1.18) and using the fact that:

$$\sum_{\sigma_j^t} \mu_{ji}(\bar{\sigma}_j^t|\bar{\sigma}_i^{t-1}) = \mu_{ji}(\bar{\sigma}_j^{t-1}|\bar{\sigma}_i^{t-2}) \tag{1.19}$$

one could equivalently write:

$$P_i(\bar{\sigma}_i^t) = \frac{1}{Z} P_i(\sigma_i^0) \sum_{\{\bar{\sigma}_k^{t-1}\}} \prod_{s=0}^{t-1} W(\sigma_i^{s+1}|\sigma_i^s, \{\sigma_j^s\}_{j\in\partial i}) \prod_{k\in\partial i} \mu_{ki}(\bar{\sigma}_k^{t-1}; \bar{\sigma}_i^{t-2}) \tag{1.20}$$

thus obtaining an expression similar to (1.17) but where we use the whole neighborhood $k \in \partial i$ in the right-most product as it was done for the static case.
It is possible, with the same reasoning, to calculate form this joint probability the one-time marginal $P_i(\sigma_i^t)$ representing the probability that variable located at node i has value σ_i^t at the time step t:

$$P_i(\sigma_i^t) = \sum_{\bar{\sigma}_i^{t-1}} P_i(\bar{\sigma}_i^t) = \sum_{\bar{\sigma}_i^{t-1}, \bar{\sigma}_j^t} P_{ij}(\bar{\sigma}_i^t, \bar{\sigma}_j^t) \tag{1.21}$$

All other types of marginals could be derived by using similar arguments.
The single trajectory $\bar{\sigma}_i^t$ can take d^t possible values, where d is the number of values that σ_i^t can take, for example $d = 2$ in the case of Ising spin. This implies that solving the cavity equation (1.17) takes an exponential number of operations: one has to perform a summation

over $d^{(t-1)(k_i-1)}$ possible configurations, where k_i is the degree of node i. Indeed, even though the dynamics is Markovian (the transition probability involves only the previous time step), the final marginal probability of a single trajectory depends on all the entire histories of the neighbors. This is unfeasible to be implemented hence some sort of approximation need to be introduced if one wants to solve the problem. Exploring this possibility is the goal of chapter 3. There we will describe a method that we introduced based on the Matrix Product State formulation of quantum mechanics. This will let us reduce the complexity of the cavity equation from exponential to polynomial and it does not rely on strict assumptions regarding the type of network (directed or undirected) or of the dynamics (irreversible or reversible).
Contrarily to the static version, the dynamic cavity method has started to be studied only recently [11]; various approximations have been proposed to tackle the problem at least in some cases [12, 13, 14].
For instance, if the graph is directed, i.e. the interaction between nodes i and j works only on one direction, then $\bar{\sigma}_i^t$ does not depend on $\bar{\sigma}_j^{t-1}$ and we can write the message $\mu_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1}) = \mu_{ij}(\bar{\sigma}_i^t)$. This would imply considering only the incoming neighbors $k \in \partial i_{IN}$ inside the product on the right-hand side of (1.17), allowing to simplify the equations. If applied to partially symmetric networks this can be considered as an heuristic approximation, which becomes exact for fully asymmetric topologies.
Another type of approximation is a factorization in time of the messages that goes under the name of *one-time approximation*. It has been used in [11, 12] for both fully and partially asymmetric networks to study the stationary state of a diluted Ising model on graphs. In those works the main assumption is to neglect correlations in time of the stationary ($t \to \infty$) solution, an hypothesis that allows a time factorization of the cavity equations and therefore the numerical implementation. Even though without bond disorder they showed similar results between theory and simulations, the situation is worsening by introducing disorder. Moreover in [12] for low temperature dynamics, depending on the update rule, they either find a different fixed-point (sequential update) compared with a Monte Carlo Markov Chain simulation, or the solution is oscillating, thus not reaching a fixed-point (parallel update). Nonetheless the intermediate time steps before reaching the stationary states show very different dynamics between Monte Carlo and the one-time cavity approximation. The convergence region of non-equilibrium stationary states under the one-time cavity approximation has been studied in [13] obtaining better results compared with other approximations.
Another way to make the problem solvable is to study processes that are described by a unidirectional dynamics, i.e. irreversible processes. That means that one only needs to keep track of an instant time step instead of the whole trajectory. The time step that has to be recorded is the one where a transition to another state happened, the main consequence of this fact being that the complexity of the algorithm becomes polynomial in t, thus it is feasible to implement the cavity equations. Another important consequence is that the validity of cavity method in these processes allows the adaptation to solve inference problems, i.e. problems where an intermediate state at time t is given and from this one wants to infer the state at a certain previous time step $t_0 < t$, this will be explained in more details below; or optimisation problems, i.e. problems where one assigns a cost function to each of the states and wants to find the optimal one.
In chapter 3 we will describe a new method to approximate the dynamic cavity equation (1.17) that is solvable in polynomial time. This works for parallel updates and does not rely on any assumption about the symmetry (or asymmetry) of the network topology nor on the irreversibility of the process. This allows to obtain results comparable to Markov Chain Monte Carlo simulations, when the latter could be used. However the potential of this method relies in the possibility of studying problems where Monte Carlo fails or is inefficient, in particular it well adapts to study infinite systems and inference and optimisation models of dynamic pro-

cesses on networks. By this we mean problems where a final configuration $\bar{\sigma}^T = (\sigma_1^T, \ldots, \sigma_V^T)$ is given and one wants to infer what was the most likely initial configuration $\bar{\sigma}^0 = (\sigma_1^0, \ldots, \sigma_V^0)$ causing that final one, given the chosen transition rule. Formally this is defined as the problem of finding:

$$\bar{\sigma}_*^0 = \arg\max_{\{\bar{\sigma}^0\}} P(\bar{\sigma}^0|\bar{\sigma}^T) \tag{1.22}$$

This problem is often tackled by using a Bayesian approach. Namely one uses Bayes' rule [15]:

$$P(\bar{\sigma}^0|\bar{\sigma}^T) \propto P(\bar{\sigma}^T|\bar{\sigma}^0)P(\bar{\sigma}^0) \tag{1.23}$$

Assuming that the initial probability distribution is uniform, then the inference problem (1.22) can be reduced to the one of calculating:

$$\bar{\sigma}_*^0 = \arg\max_{\{\bar{\sigma}^0\}} P(\bar{\sigma}^T|\bar{\sigma}^0) \tag{1.24}$$

Therefore a strategy could be to run message-passing equations similar to (1.17) for all the possible initial configurations and at the end of $t = T$ time steps collect the results to calculate $P(\bar{\sigma}^T|\bar{\sigma}^0)$. A common approach is to assign an energy to each of the initial configurations such that the most likely one will be assigned the least energy:

$$E(\bar{\sigma}^0) = -\log P(\bar{\sigma}^T|\bar{\sigma}^0) \tag{1.25}$$

Thus one can rank accordingly the configurations and select the best one given this definition of energy. Message-passing have been recently proven to be efficient when studying inference problems of epidemics on networks[16, 17]. In these works models with unidirectional dynamics are considered, so that the computational complexity is highly reduced and thus one can iterate the equations numerically.

1.2.1 Glauber dynamics

One of the most studied among the dynamic processes on networks is the Glauber dynamic [18]. This is a binary-variable model where the interactions are between neighboring nodes and their strength is given by a stochastic variable J_{ij}. The field acting at time t on a given node $i \in \mathcal{V}$ is defined by:

$$h_i^t = \phi_i + \sum_{j \in \partial i} J_{ij}\sigma_j^t \tag{1.26}$$

where ϕ_i is a local field that does not depend on the state of the neighborhood. To fix the ideas, in the rest we will consider only the case $\phi_i = 0$.
The transition probability of node i getting value $\sigma_i^{t+1} \in \{+1, -1\}$ at time step $t+1$, given the state of its neighborhood at the previous time t, is then given by:

$$W(\sigma_i^{t+1}|\{\sigma_j^t\}_{j\in\partial i}) = \frac{e^{-\beta\sigma_i^{t+1}h_i^t}}{2\cosh(\beta h_i^t)} \tag{1.27}$$

The parameter β is treated as an inverse temperature and it controls the strength of the disorder. For $\beta \to \infty$ (or $T \to 0$) the dynamic is deterministic and the system tends towards the ground state with lowest energy. On the contrary, for $\beta \to 0$ (or $T \to \infty$) the dynamic is completely stochastic. The last thing one has to do in order to completely determine the dynamics is define an update routine and the time step. Here we will focus on discrete time steps and *parallel dynamics*, meaning that at each time step t all the variables are simultaneously updated. A famous model that uses this type of dynamics is the Little model of neural networks [19]. An alternative update rule is the *sequential dymanics*, the case where at each time step only one variable is updated. An iteration sweep is completed when all the V variables have been updated.
Considering the parallel dynamics, the joint distribution of the entire system at time step $t+1$

is given by:

$$P(\sigma_1^{t+1},\ldots,\sigma_V^{t+i}) = \prod_{i\in\mathcal{V}} W(\sigma_i^{t+1}|\{\sigma_j^t\}_{j\in\partial i})P(\sigma_1^t,\ldots,\sigma_V^t) \tag{1.28}$$

where the transition probability is the Glauber one as in (1.27).
This probability distribution obeys detailed-balance:

$$W(\bar{\sigma}'|\bar{\sigma})P(\bar{\sigma}) = W(\bar{\sigma}|\bar{\sigma}')P(\bar{\sigma}') \tag{1.29}$$

in the case that interactions are symmetric:

$$J_{ij} = J_{ji} \tag{1.30}$$

This is the case we will consider here. This means that the dynamics will eventually reach equilibrium, with a Gibbs probability distribution of the form:

$$P_{eq}(\bar{\sigma}) \propto e^{-\beta H_\beta(\bar{\sigma})} \tag{1.31}$$

where the function $H_\beta(\bar{\sigma})$ is the Peretto's *pseudo-Hamiltonian* [20] given by:

$$H_\beta(\bar{\sigma}) = -\frac{1}{\beta}\sum_{i\in\mathcal{V}} \log 2\cosh(\beta h_i) \tag{1.32}$$

An alternative valid Hamiltonian, in the sense that is a function bounded from below and it decreases almost monotonically with every step of the dynamics, is a quantity that can be written as a sum of two-variable functions [21, 22]:

$$H^t(\bar{\sigma}^t) = -\sum_{(ij)\in\mathcal{E}} J_{ij}\sigma_i^t\sigma_j^{t-1} \tag{1.33}$$

This last expression is more convenient to be used when considering averages: one only needs two-variable joint distributions $P(\sigma_i^t,\sigma_j^{t-1})$ instead of expression (1.32) which needs the entire neighborhood joint distribution. To see that this is a valid Hamiltonian in the sense specified above, we can write the difference [21]:

$$\Delta H = H^{t+1}(\bar{\sigma}^{t+1}) - H^t(\bar{\sigma}^t) = -\sum_i \left[\sigma_i^{t+1} - \sigma_i^{t-1}\right]\sum_{j\in\partial i} J_{ij}\sigma_j^t \tag{1.34}$$

From this expression one sees that $H^t(\bar{\sigma}^t)$ does not change, i.e. $\Delta H = 0$, when either consecutive steps are identical or when two states alternate, i.e. $\sigma_i^{t+1} - \sigma_i^{t-1} = 0 \quad \forall i$. The first case is when the system has reached a fixed point, whereas the second is when a system is stuck in a *2-cycle*. This last situation is peculiar of the synchronous dynamics and in often seen for parallel updates.
The evolution of this dynamics has been studied using cavity method in [11, 23, 13] with single-spin time trajectories as variables. The computational complexity therefore grows exponentially for unidirectional graphs, allowing numerical simulations only for few time steps.
Suppose that one is able to simulate a system under this dynamics. Examples of quantities of interest to be measured as the system evolves are the one-time single-spin magnetization $m_i(t)$ defined as:

$$m_i(t) = \sum_{\{\sigma_i^t\}} \sigma_i^t P_i(\sigma_i^t) = \langle\sigma_i^t\rangle \tag{1.35}$$

where $P_i(\sigma_i^t)$ is the one-time marginal (1.21). From this expression one could define two other observables, the total magnetization:

$$m(t) = \frac{1}{V}\sum_i m_i(t) \tag{1.36}$$

and the (quadratic) change in magnetization $\Delta m(t)$ between two time-steps:

$$\Delta m(t) = \frac{1}{V}\sum_{i=1}^{V} \left[m_i(t) - m_i(t-1)\right]^2 \tag{1.37}$$

This last expressions gives indications if the dynamics has reached equilibrium, thus can be used for instance as a measure of algorithmic convergence when one expects that an equilibrium regime has to be reached. A similar quantity often use in Spin Glass theory [24] is the Edwards-Anderson order parameter:

$$q_{EA} = \frac{1}{V}\sum_i \langle \sigma_i^t \rangle^2 \tag{1.38}$$

This parameter is used to measure the spins-glass phase, in fact a measured $q_{EA} \neq 0$ denotes a freezing of the spins in random orientations. Notice that in this case the magnetization could give misleading interpretations because it could be that the total $m(t) = 0$, as is the case of the paramagnetic phase where $m_i(t) = 0$. Thus one cannot distinguish whether the system is frozen in a random orientation (spin-glass phase) or is in a paramagnetic phase just by looking at $m(t)$.
These quantities, along with the energy (1.33) will be the observables that we will measure to test the validity of the dynamic cavity method approximation described in chapter 3.

1.2.2 The majority rule

Another example of a well studied dynamic process on networks is the majority rule.
In this model a variable at time t takes the value taken by the majority of its neighbors at the previous time step:

$$\sigma_i^{t+1} = sign\left(\sum_{j\in\partial i} \sigma_j^t\right) \tag{1.39}$$

if $\left(\sum_{j\in\partial i} \sigma_j^t\right) \neq 0$, otherwise

$$\sigma_i^{t+1} = \begin{cases} \sigma_i^t & \text{with probability } 1/2 \\ -\sigma_i^t & \text{with probability } 1/2 \end{cases} \tag{1.40}$$

The transition rule can thus be defined as:

$$W(\sigma_i^{t+1}|\{\sigma_j^t\}_{j\in\partial i}) = \begin{cases} \mathbb{I}\left\{\sigma_i^{t+1} = sign\left(\sum_{j\in\partial i}\sigma_j^t\right)\right\} & \text{if } \sum_j \sigma_j^t \neq 0 \\ 1/2 & \text{if } \sum_j \sigma_j^t = 0 \end{cases} \tag{1.41}$$

where $\mathbb{I}(a) = 1$ if the boolean variable a is true, otherwise $\mathbb{I}(a) = 0$. The majority rule (1.41) can be mapped into the $\beta \to \infty$ limit of a Glauber dynamics as in (1.27) in the ferromagnetic scenario, i.e. when $J_{ij} = J = +1\ \forall (ij) \in \mathcal{E}$.
A part from the case where $\sum_j \sigma_j^t = 0$ and the state σ_i^t is determined stochastically, for other scenarios the process is deterministic. Nonetheless this definition can be generalized to a stochastic model where a noise is introduced so that there is a finite probability that the state makes en error, i.e. it updates with the opposite sign of the majority. This can be formalized introducing a real parameter $Q \in [0, 1]$ such that:

$$W(\sigma_i^{t+1}|\{\sigma_j^t\}_{j\in\partial i}) = \begin{cases} Q - (2Q-1)\mathbb{I}\left\{\sigma_i^{t+1} = sign\left(\sum_{j\in\partial i}\sigma_j^t\right)\right\} & \text{if } \sum_j \sigma_j^t \neq 0 \\ 1/2 & \text{if } \sum_j \sigma_j^t = 0 \end{cases} \tag{1.42}$$

which means that the probability for variable σ_i^{t+1} to take the sign of the majority is not exactly 1 but instead $1 - Q$.
Focusing on the case $Q = 0$, one interesting question for this problem is whether the dynamics evolves to a *consensus* regime, namely a situation where all the variables take the same value. The system has an up-down symmetry therefore, without loss of generality, here we define consensus as the situation where all variables take $\sigma_i^t = +1$ (to focus the ideas on one of the two states). In general the way the systems converges to consensus, if this is reached at all, depends on the initial bias θ_0 of the number of $+1$-variables. Formally, this quantity could be

defined as a real number in $[0,1]$ such that $\forall i \in \mathcal{V}$ the initial marginal distribution is:

$$\begin{aligned} P_0(\sigma_i^0 = +1) &= \frac{1+\theta_0}{2} \\ P_0(\sigma_i^0 = -1) &= \frac{1-\theta_0}{2} \end{aligned}$$

This definition implies that the larger θ_0 the higher the initial number of $+1$ variables, therefore the more likely consensus will be reached. This suggests the definition of a quantity of interest for this problem as the *consensus threshold* θ_c, namely the minimum value of θ_0 such that the final state has all variables $\sigma^t = +1$ with probability 1. Formally:

$$\theta_c = \inf\{\theta_0 : P_{\theta_0}(\sigma_1^t = +1, \ldots, \sigma_V^t = +1) = 1\} \tag{1.43}$$

where $P_{\theta_0}(\sigma_1^t, \ldots, \sigma_V^t)$ is the probability to have the configuration $(\sigma_1^t, \ldots, \sigma_V^t)$ at final time t given the initial bias θ_0 at time $t = 0$.
Here we did not specify the final time step t, this is often set to $t = \infty$, a situation where one wants that eventually the systems reaches consensus, no matter how long it will take. However one could also fix an arbitrary finite t so that consensus has to be reached arbitrarily fast.
As for the case $Q > 0$, one usually fixes $\theta_0 = 0$ or very small, i.e. unbiased or almost unbiased initial condition, and looks for the minimum Q such that consensus is not reached. In fact the source of noise causes the variables to make errors when updating, thus leading the dynamics out of consensus for increasing noise strength Q.

Exact calculations of the dynamics is not possible in general, therefore this model has often been studied numerically using Monte Carlo simulations [25, 26]. Analytical approaches usually rely on using mean-field approximations where one neglects dynamic correlations, so that the state of a node depends on the average fraction of *active* nodes, i.e. nodes with $\sigma = +1$; thus treating the state of a node and the one of its neighbors as independent. Although simple to derive and solve, mean-field theories are not very accurate to study sparse graphs, i.e. when the average degree $\langle k \rangle \ll V$. Indeed it has been shown that it is not appropriate to describe the zero temperature Glauber dynamics, i.e. equivalent to the majority rule, on random networks [27]. The theory can be improved by considering dynamic correlations at a pairwise-level, such as for pair approximation. The cost is an increase in complexity that becomes more and more important when increasing the accuracy. This is the case of approximate master equations models [28], where the state of the neighborhood is taken into account generating large systems of differential equations which are hard to solve. This complexity can be reduced by considering irreversible dynamics. For equilibrium models, i.e. where detailed-balance is satisfied, these different approximation methods result in dynamics evolving to the same final state for $t \to \infty$, despite for finite t they may differ. However, for non-equilibrium models they differ also for $t \to \infty$. This is the case of the majority rule with $Q > 0$.
Recently the majority rule (1.41), i.e. $Q = 0$, has been studied using the dynamic cavity equations (1.17) [23]. This is a pioneering work in the sense that is one of the first to propose the cavity formalism to describe dynamic processes on networks. In that work they focus on regular graphs and give bounds for the consensus threshold θ_c as a function of the connectivity k. In order to give estimates for infinite systems, they consider growing finite networks of size V and calculate the threshold $\theta_{c,V}(k)$ as V grows. They performed numerical simulations for varying k to see that this converges to a limit $\theta_{c,rgraph}(k)$ such that above this threshold the dynamics converges with high probability to all $+1$. On the contrary, below this value the dynamics gets stuck on either a 2-cycle or on a stationary point. This threshold decays to zero with increasing connectivity k.

1.2.3 Monte Carlo Markov Chain simulations

In this section we outline a technique often used to simulate numerically dynamical processes, it is usually called Monte Carlo Markov Chain (MCMC). Its popularity is due to its simplicity and the fact that it adapts well to simulate any type of process, regardless network topology, dynamic type or type of variables considered. Despite these nice properties it presents also several drawback, but we will describe them below. This technique will be useful for comparison with our proposed dynamic cavity method, in fact in section 3.2.3 we will present numerical simulations of both Glauber dynamics and Majority rule obtained through Monte Carlo and our proposed method based on matrix product states.
The literature about this topic is very rich but often this causes confusion about what specific variant of the Monte Carlo method is used. Therefore here we present only the variant of the algorithm that we are going to use in our numerical simulations. The algorithm under parallel and synchronous dynamics (the one considered in this work) works as following.
We initialize the system $(\sigma_1^0, \dots, \sigma_V^0)$ in a state sampled from the initial distribution $P_0(\sigma_1^0, \dots, \sigma_V^0)$. Assuming that this is factorized as $P_0(\{\sigma_i^0\}) = \prod_{j=1}^{V} P_i(\sigma_i^0)$, the initial value of each variable σ_i^0 is assigned to σ with probability $P_i(\sigma)$.
Then $\forall i \in \mathcal{V}$ we update the variable σ_i^1 at the next time step $t = 1$ using a transition rule given by the model considered, for instance (1.27) for the Glauber dynamics or (1.41) for the majority rule. For a binary variable as in our case, this can be achieved by extracting a real random number r uniformly in $[0, 1]$ and then calculating $W(\sigma_i^{t+1} = +1|\{\sigma_j^t\}_{j\in\partial i})$ (here $t = 0$). If this number is greater than the random r than the variable σ_i^1 is assigned value $+1$, otherwise the assigned value will be -1. Similar reasoning applies if one calculates instead $W(\sigma_i^{t+1} = -1|\{\sigma_j^t\}_{j\in\partial i})$. The same procedure is repeated for all V variables and this is considered as one sweep. Note that the order with which the variables are picked does not affect the final result. Under parallel dynamic at each Monte Carlo sweep the new variables σ_i^{t+1} are updated using the old ones σ_i^t, but these remain always the same inside the sweep routine. On the contrary, for sequential updates one would pick one variable, update it, and then possibly use this new value to update one of its neighbors. Therefore in general these two types of update gives different configurations.
One performs $t = T$ sweeps, where T is the maximum time of the dynamics and is given as input. Eventually one gets a final configuration $(\sigma_1^T, \dots, \sigma_V^T)$ that is only one of the possible ones obtained starting from the initial distribution and following the chosen update rule. Therefore one has to repeat the entire routine for N_{real} number of instances. The integer number N_{real} is given as input and the greater its value the greater the statistical sampling from the desired distribution (1.15).
Although being simple to implement and flexible in the sense that it easily adapts to simulate any type of network or dynamics, this algorithm presents several drawbacks.
Firstly, if one is interested to sample a particular event, for instance the all $+1$ configuration, it takes a big number of realizations N_{real} in order to get good statistics, unless this happens with high probability. In particular, when considering rare events, most of the Monte Carlo realizations will be rejected because will not end up in the desired rare configuration.
Secondly, it works well for finite system sizes but it does not apply to infinite systems. When the goal is to study the thermodynamic limit $V \to \infty$, the Monte Carlo algorithm has to be run for increasing system sizes and then the infinite limit can only be extrapolated from these results.
Finally, this algorithm is not suited to set up inference problems as the one defined in (1.22). The strategy of letting the system evolve through the chosen dynamics, starting from each possible initial configuration so to calculate $P(\bar{\sigma}^T|\bar{\sigma}^0)$, does not work well in this case. In fact one would have to run a big number of realizations N_{real} for each of the possible initial config-

urations in order to get a good statistics of the set of trajectories, among the one starting at a given $\bar{\sigma}^0$, that converge to the desired final state.

1.2.4 Population dynamics

Monte Carlo method is not appropriate to describe infinite systems. Therefore a common strategy used to study the thermodynamic limit $V \to \infty$ through numerical simulations is to start by using a relatively small finite size V and then let the system grow towards bigger values of V. The statistics of the macroscopic observables of interest are collected as a function of V and the results on the infinite system is extrapolated by the one collected for finite and growing V.
However, if one has access to at least some approximate probability distribution of the one of interest $P(\bar{\sigma}^t)$, then it could use this to study directly the thermodynamic limit using the *population dynamics* algorithm [29]. To fix the ideas think for instance at the cavity messages (1.17) $\mu_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1})$ of the dynamical case (but the rest will be valid also if one considers the static messages $m_{i\to a}(\sigma_i)$ and $\hat{m}_{a\to i}(\sigma_i)$ (1.4), (1.5)), as the approximate marginals of the exact distribution. The idea behind population dynamics is to approximate these functions by sampling N i.i.d. copies of μ_{ij}, this set of copies $\mathcal{P} = \{\mu\}$ is called population [6]. Then for large N the empirical distribution of this sample converges to the actual distribution of μ_{ij}. The algorithmic steps for the dynamical cavity method are the following:

- Initialize $\{\mu\}$ using the initial distribution $P_0(\bar{\sigma}^0)$
- Pick a random $\mu^* \in \mathcal{P}$ and an integer k with distribution $P(k)$
- Pick k other $\mu \in \mathcal{P}$ and k numbers J with distribution $P(J)$
- Update μ^* using the update rule (1.17) where the μ_{ki} are replaced by the set of k μ and J extracted before
- Repeat for all $\mu \in \mathcal{P}$ and for t steps

At the end one can calculate macroscopic averages using the population. For example the total magnetization (1.36)

$$m(t) = \frac{1}{V}\sum_i m_i(t) \tag{1.44}$$

can be calculated as:

$$m(t) = \frac{1}{R}\sum_{n=1}^{R} m_n(t) \tag{1.45}$$

where R is a sufficiently big number, typically of the order of V, and the $m_n(t)$ are calculated using the population in the following way.

- Pick randomly two $\mu \in \mathcal{P}$ and calculate $P(\sigma^t)$ using (1.16) and (1.21)
- Calculate $m_n(t) = \sum_{\{\sigma^t\}} \sigma^t P(\sigma^t)$

The resulting $m(t)$ tends to the expected macroscopic average over the chosen graph type, characterized by $P(k)$ and over the disorder, characterized by $P(J)$, for large values of N and R.
Similar considerations can be done for the static equations, but in this case one has to iterate the algorithm for several iteration steps. These are not related to any dynamics time (we are in the static case). This time one has to wait until convergence of the population and this can be

checked, for instance, by calculating (1.45) at the end of each iteration step, and controllin the difference with the one obtained at the previous step. Once the averages are stable enough, i.e. up to statistical fluctuations of the order of $1/\sqrt{N}$, than this is a signal that the population has converged.
Here we outlined the dynamical case because this is the scenario where we will use this algorithm in 3.2. In particular we will address the majority rule, or zero temperature Glauber dynamics, on regular graphs of infinite size. On the contrary we did not use population dynamics in our application of the static problem. In routing problems in fact the presence of a parameter M (the number of communications), enters explicitly in the expressions of the messages $\mu_{ij}(\bar{I}_{ij})$ because it represents the domain of the variables $\bar{I}_{ij}$, which is of size $2M+1$. But when we fix M at the same time we are fixing a system size V, as we will see in section (2.1). The result is the impossibility of decoupling the messages domain from the system size, thus preventing us to properly employ the thermodynamic limit through population dynamics.

Chapter 2

Routing optimisation on random networks

The power of the cavity method is the fact that global information is processed through local updates among neighborhoods of nodes. This is particularly useful in the case of certain types of optimisation problems where the cost function and the constraints are global and thus global optimisation is required. Routing and path optimisation of networks are a famous example among these. The high number of variables involved and their non-localized nature require a global optimisation that is typically unfeasible to be performed through an extensive search through the whole configuration space. Indeed they are often classified among the NP-complete class [30] of hard combinatorial problems.
A typical routing optimisation problem on networks is a defined as follows. One has a set of users that want to either move along the network or communicate with each other. They are subject to a set of constraints and a cost function that depends on the specific problem. Typically the cost function takes into account variables that involve the quality of the network performance such as traffic, delay and total path length. Typical constraints take into account link or node capacity limits and path continuity. The aim of the optimisation problem is to route path trajectories (or communications) so to minimize the cost function and respect the constraints.
These problems find many interdisciplinary applications such as in communication systems, transmission networks and virtual circuit design. The interaction between paths is non-localized hence local protocols fail to find the optimal configuration and global optimisation is required. The types of algorithms proposed in the literature to tackle these problems use strategies such as greedy heuristics, Montecarlo and linear programming techniques. In this chapter we present a distributed algorithm that exploits message-passing techniques to efficiently tackle several variants of routing optimisation problems by using local information.
The chapter outline is the following. We first set up the general framework that is shared among all the proposed variants, this being based on the cavity method formalism. Then in the following sections we present three specific problems starting from the one that has the hardest constraints (the Node-Disjoint Path problem), then relaxing a bit the hard constraint (the Edge-Disjoint Path problem), to finish with the complete constraint relaxation by introducing the Nash Equilibrium variant of the problem.

2.0.5 The model

Here we set up the analytical formalism that will be used in all the next sections. In all these problems the combination of cost function and constraints are chosen so to minimize both path length and traffic, the latter represented by paths overlap. The model is conveniently formalized

by taking variables I_{ij} sitting on edges because the cost function $f(I_{ij})$ typically involves the flow passing throuhg an edge and the constraints, that act as factor nodes, are usually defined on nodes of the graph. Therefore variables nodes are surrounded at most by three factor nodes: one for each terminal node of the edge and one (if not absent) representing the cost of that edge. Then the cavity formulation only needs one type of message as in (1.14), with variable nodes denoted as (ij) and factor nodes as i. Then the left-hand side of (1.14) can be written as $E^{(t+1)}_{(ij)\to j}(\bar{I}_{ij})$, whereas the first sum in the right-hand side is made of only two contributions so that at the end we obtain:

$$E^{(t+1)}_{(ij)\to j}(\bar{I}_{ij}) = \min_{\bar{I}_{\partial i\setminus(ij)}|constraint} \left\{ \sum_{k\in\partial i\setminus j} E^{(t)}_{(ki)\to i}(\bar{I}_{ki}) \right\} + f(\bar{I}_{ij}) + C^{(t)}_{\to j} \tag{2.1}$$

Notice that the edge cost $f(\bar{I}_{ij})$ stays outside the minimum because, for a given variable $\bar{I}_{ij}$, its value is fixed. Furthermore, notice how the term $E_b(\sigma_{\partial b})$ does not appear explicitly anymore. This is indeed implicitly present behind the *constraint*: it plays the role of the interaction that couples the neighbors together. This scenario is represented in figure 2.1. We notice the redundancy of using the edge notation in $(ij) \to j$, so from now on we drop it in favor of ij. In this way the message $E^{(t+1)}_{(ij)\to j}(\bar{I}_{ij})$ becomes $E^{(t+1)}_{ij}(\bar{I}_{ij})$.

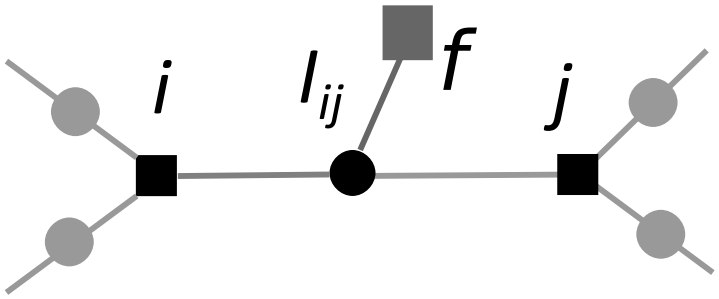


Figure 2.1: Factor graph for routing. Black squares represent function nodes, these typically represent the constraints coupling neighboring variables. Circles represent variable nodes that in this case are located on edges of the underlying topology. Grey squares represent external fields acting on the variable nodes, typically these are the cost of the edge.

The underlying network is represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $i \in \mathcal{V}$ and edges $(ij) \in \mathcal{E}$. We define a set of M communications $\mathcal{C}$ as paths along edges of the graph. These originates at source nodes S (sender) ad terminates at receiver nodes R. Therefore a communication $\mathcal{C}^\mu \in \mathcal{C}$ can be characterized by a pair of nodes (S^μ, R^μ), with $\mu = 1, \dots, M$, i.e. the sender and receiver of that specific communication. At each node $i \in \mathcal{V}$ we define a variable Λ^μ_i that defines the state of the node. This is assigned at the beginning and will be an input parameter of the problem:

$$\Lambda^\mu_i = \begin{cases} +1 & \text{if } i \text{ is a sender for communication } \mu \\ -1 & \text{if } i \text{ is a receiver for communication } \mu \\ 0 & \text{if } i \text{ is neither a sender nor a receiver for communication } \mu \end{cases} \tag{2.2}$$

We call a site *transit node* if $\Lambda^\mu_i = 0$. In principle all communications can pass through a given node i, therefore we need an M-dimensional vector to fully characterize it: $\bar{\Lambda}_i = (\Lambda^1_i, \dots, \Lambda^M_i)$. We define the norm of a vector as $||\bar{\Lambda}_i|| := \sum_{\mu=1}^M |\Lambda^\mu_i|$, where $|\Lambda^\mu_i|$ is the absolute value of Λ^μ_i. In the problems that we consider here each node can be either a sender or a receiver for only one communication at a time (or a transit node), hence with this convention we have $||\bar{\Lambda}_i|| \in \{0, 1\}$. Variables are defined on edges $(ij) \in \mathcal{E}$ and they represent the communication

flow passing along the given edge:

$$I_{ij}^{\mu} = \begin{cases} +1 & \text{if } \mu \text{ passes through } (ij) \text{ from } i \text{ to } j \\ -1 & \text{if } \mu \text{ passes through } (ij) \text{ from } j \text{ to } i \\ 0 & \text{if } \mu \text{ does not pass through } (ij) \end{cases} \tag{2.3}$$

Notice that in this formalism $I_{ij}^{\mu} = -I_{ji}^{\mu}$. We term these variables currents and define for each edge (ij) a vector $\bar{I}_{ij} := (I_{ij}^1, \dots, I_{ij}^M)$ that collects information on all currents involved in that edge. The state of the whole network is thus characterized by the set $\{\bar{I}_{ij} : (ij) \in \mathcal{E}\}$. In general $||\bar{I}_{ij}|| \in [-M, \dots, M]$ but depending on the type of constraint this interval can be reduced.

A constrained that is always present in the different variants of the problems is path contiguity: a path is a set of consecutive links, i.e. no interruptions are allowed. This constraint causes the interaction between currents to be highly non-localized and this represent one of the main challenges of the problem. For instance this makes it difficult to use Montecarlo algorithms because one has to define a suitable Montecarlo step where only a finite number of variables are modified at each step. Hence if one wants to slightly modify a path and respect the contiguity constraint it is not trivial how to do it by only varying few edges.

Around a given node $i \in \mathcal{V}$ we can represent this constraint as the Kirchhoff law:

$$\sum_{j \in \partial\, i} I_{ij}^{\mu} - \Lambda_i^{\mu} = 0 \qquad \forall \mu = 1, \dots, M \tag{2.4}$$

In this work we consider costs $f(||\bar{I}_{ij}||)$ that are functions of the total current passing through an edge so that the overall network cost is a sum over all the edges:

$$c(\{\bar{I}_{ij}\}) = \sum_{(ij)\in\mathcal{E}} f(||\bar{I}_{ij}||) \tag{2.5}$$

This cost can be of different shapes but a typical choice is power-law $f(I) = I^{\alpha}$ with α a parameter that can be tuned in order to favor or penalizes traffic. This approach has been proposed in [31] where both the cases are studied considering a transportation network. In the models we consider here we adopt a linear cost, i.e. $\alpha = 1$. In this way the total cost will be exactly equal to the total path length $L_{tot} = \sum_{\mu=1}^{M} L^{\mu}$, where we define the length L^{μ} of the path μ as the number of edges contributing to the trajectory. Now we have all the ingredients needed to set up the *min-sum* cavity equation (2.1) (rewritten using the compact notation) of the routing problem:

$$E_{ij}^{(t+1)}(\bar{I}_{ij}) = \min_{\bar{I}_{ki}|constraint} \left\{ \sum_{k\in\partial i\setminus j} E_{ki}^{(t)}(\bar{I}_{ki}) \right\} + f(||\bar{I}_{ij}||) \tag{2.6}$$

where we have dropped the additive constant $C_{\to i}^{(t)}$ because from now on we assume that messages are normalized such that $\min_{\bar{I}} E_{ij}^{(t+1)}(\bar{I}) = 0$. This is one of the possible choices and it is equivalent to fixing the constant at each time step. Equation (2.6) is saying that: in order to calculate $E_{ij}(\bar{I}_{ij})$ for a given current $\bar{I}_{ij}$, one has to find the best configuration of the neighbors $\{\bar{I}_{ki}\}$ such that the set of constraint is respected. Here *constraint* always contains Kirchhoff law but, as we will see in the next sections, it will contain other constraints specified by the model considered.

One then arbitrarily initializes the messages and iterates this equation until convergence. Once this is reached, the converged messages $E_{ij}^{*}(\bar{I}_{ij})$ are collected to calculate the optimal configu-

ration $\{\bar{I}^*_{ij}\}$ using the incoming and outgoing messages for a given edge:

$$\bar{I}^*_{ij} = \arg\min_{\bar{I}} \left\{ E^*_{ij}(\bar{I}) + E^*_{ji}(-\bar{I}) - f(||\bar{I}||) \right\} \tag{2.7}$$

where the last term is subtracted to avoid double counting of the cost of the single edge (ij). We sum over all $(ij) \in \mathcal{E}$ to find the different paths and total length:

$$L_{tot} := \sum_{(ij)\in\mathcal{E}} ||\bar{I}^*_{ij}|| \tag{2.8}$$

Finally, the minimal cost is then:

$$c^* := \min_{\bar{I}} \frac{1}{|\mathcal{E}|} \sum_{(ij)\in\mathcal{E}} \left\{ E^*_{ij}(\bar{I}) + E^*_{ji}(-\bar{I}) - f(||\bar{I}||) \right\} \tag{2.9}$$

Notice that in order to find the optimal configuration $\{\bar{I}^*_{ij}\}$ it is necessary to perform the calculation (2.7) $\forall (ij) \in \mathcal{E}$. These calculations are carried out link by link as if the energies per link were statistically independent. It is not intuitively clear that doing this will result in the optimal paths which do not overlap and are also fully connected from the source to the receiver. This is a consequence of having used messages which implicitly contain global information on the constraints and path lengths, so that the energies per link are indeed globally interdependent albeit in a non-obvious manner.

With this general set up the computational complexity of the algorithm is exponential in the number of communications M. Indeed there are two main issues: the first is that messages $E_{ij}(\bar{I}_{ij})$ can a priori take 3^M values corresponding to all possible combinations of currents passing through a single edge (ij). The complexity is then due to the size of the currents' configuration space. Secondly, even if the configuration space is reduced (for instance by introducing some constraints), for a given output current $\bar{I}_{ij}$, there is a large number of possible neighborhood's configuration $\{\bar{I}_{ki}\}_{k\in\partial i\setminus j}$ around the node i that is consistent with the set of constraints. In the minimum inside (2.6) one has indeed to consider all the possible configuration of currents entering and exiting node i. Unfortunately this number grows exponentially with the degree of node i and with M.

In the next sections we will tackle these two problems separately. The first problem will be dealt by shrinking the configuration space. This can be achieved by introducing an hard constraint that forbids many configurations, and this method will be explored when considering the the Node-Disjoint Path problem in section 2.1. We tackled the second issue by performing a mapping from a constrained routing problem into a combinatorial weighted matching [32] problem. This has the effect or reducing the complexity to the one of a standard weighted matching algorithm. This method will be presented when considering the Edge-Disjoint Path problem in section 2.2. Finally we will relax further the constraints (and the configuration space increases) outlining the Nash Equilibrium counterpart of the problem in section 2.3.

For the first two problems we performed extensive numerical simulations to test the algorithms and compare it with alternative algorithms found in literature. For the Edge-Disjoint Path we found benchmark results that allowed performance comparison. Unfortunately this was not possible for the Node-Disjoint Path because of the absence of benchmarks; nonetheless, in this case we compared results with a greedy algorithm. We also study statistical and scaling properties of quantities of interest as a function of network size and number of paths starting from the collected solutions.

As graph topologies we considered several types of complex networks. Here we quickly summarize their properties but we leave more details in the appendix 3. Standard regular random graphs (Reg): each node has fixed degree k; Erdős Rényi random graphs (ER) [33]: edges are drawn at random between each pair of nodes with probability $p = \langle k \rangle / V$; a decorated random

graph (RER): starting from a regular random graph of degree k_1 (which is the minimum degree of this graphs), we then randomly add new edges as in the ER model until the final average connectivity is $\langle k \rangle = k > k_1$. Notice that the degree distribution in this case can be obtained by writing $k = k_1 + d$ where d is Poisson distributed with $\langle d \rangle = k - k_1$.

One final remark concerns the infinite system case. In all what follows we tested the algorithm on single instances of graphs and did not use population dynamics because we found it not appropriate to study routing problems. The reason is the presence of the parameter M that affects the domain of the fluxes $\bar{I}_{ij}$ and thus enters explicitly in the expressions of the messages. In fact when we fix M at the same time we are fixing a system size V, because we extract random pairs (S, R) with density M/V, thus coupling the message domain with the system size. There is also another problem, that such a macroscopic oriented approach would introduce averages over all possible configurations (S, R), including both frustrated and unfrustrated configurations with much higher energies. Namely configurations for which the decision problem, such as is the case in the node-disjoint path problem, could not find a solution. Thus the macroscopic averages are highly biased by the fewer frustrated configurations and more complex algorithm should be designed to discard such cases. For these two reasons we did not consider in the population dynamic counterpart of the algorithm but focused only on averages over single instances.

2.1 The Node-Disjoint Path problem

The Node-Disjoint Path problem (NDP) is a variant of a routing problem where path trajectories are subject to the hard constraint that no two paths can share a node, thus overlap at nodes is forbidden. In figure 2.2 we can see an example of non optimized network (left) and the corresponding NDP solution (right). We see that communications are forced to reroute and take longer paths in order to avoid overlap at nodes. The total path length of the new solution is minimized in accordance with the node-disjointness constraint.

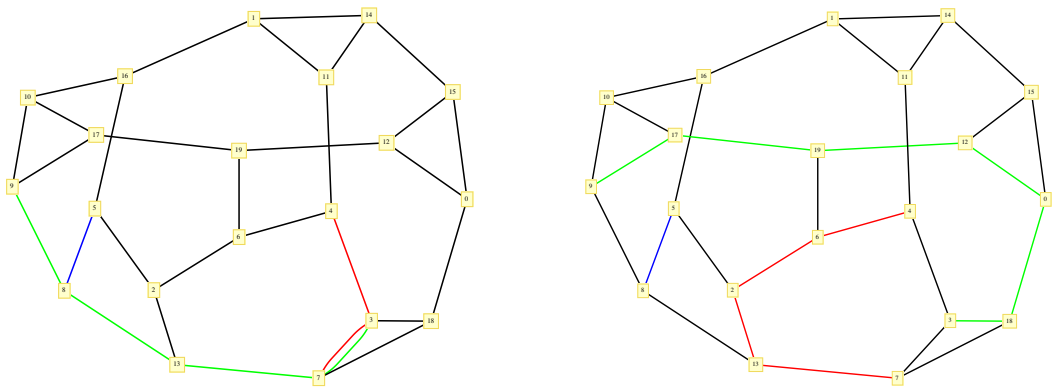


Figure 2.2: NDP network example. (Left) Example of a network with three communications. The blue and red communications are overlapping at nodes with the green one. (Right) The NDP optimized solution: the green and red ones are forced to reroute through longer paths. The total path length in nonetheless the minimal achievable when enforcing the node-disjointness constraint.

The principal consequence of this constraint is that the configuration space of the variables

(currents) is shrunk. In fact, saying that paths cannot overlap at nodes, means that along an edge $(ij) \in \mathcal{E}$ can pass at most one communication at a time, i.e. $||\bar{I}_{ij}|| \in \{0, 1\}$. Therefore the number of possible values taken by $\bar{I}_{ij}$ is shrunk from 3^M to $2M + 1$, one incoming and one outgoing for each communication and zero. This problem was studied as a pure theoretical problem by mathematicians in the series of graph minors [34], under the name of subgraph homeomorphism problem. But it also finds many applications in the field of communication networks. In particular in communication systems where the quality of the performance strictly depends on a limited network capacity, it is paramount to manage traffic in order to prevent congestion and allow a smooth information flow. This can be seen for instance in virtual circuit routing where switches located at nodes may become bottlenecks. In general the NDP suits well to manage networks where the quality of service (QoS) is one of the main requirements. For instance in wireless communication signal interference (path overlap) causes low transmission quality whereas in optical networks transmissions using the same wavelength cannot share the same edge or vertex, hence all communications of the same wavelength must be non-overlapping (disjoint). Finally, another important feature of the NDP is its distributive character. This means that load is balanced among users and this makes the network more resilient to failures and thus more robust.
Several algorithms have been proposed to solve this problem but often depend on the specific network topology [35] and mostly focus on the optimisation version of the problem, i.e. maximizing the number of paths routed [36]. The satisfiability version of the problem, i.e. whether all paths can be routed successfully without overlap, is not considered; theoretical studies often give bounds to the achievable approximation instead of providing a practical algorithm for individual instances and fail to calculate path lengths and possible overlaps at the same time as part of the optimisation process observables. In this work we focus on studying the satisfiability version and use the cavity method to solve the problem. The results of this study can be found in [37].

2.1.1 The NDP model

Adopting the general formalism introduced in section 2.0.5, here we specialize to the NDP case. The peculiarity of this model is the hard constraint of node-disjointness. This corresponds to have $||\bar{I}_{ij}|| \in \{0, 1\}$ and therefore we take:

$$f(||\bar{I}||) = \begin{cases} \infty & \text{if} \quad ||\bar{I}|| \geq 2 \\ 1 & \text{if} \quad ||\bar{I}|| = 1 \\ 0 & \text{if} \quad ||\bar{I}|| = 0 \end{cases} \tag{2.10}$$

so that the cost function (2.10) represents indeed the total path length.
One then has to solve the *min-sum* equation (2.6) iteratively and to do this we define a protocol for taking into account only the allowed configurations at each edge given the current value $\bar{I}$ passing through it and $\bar{\Lambda}_i$ at vertex i.

If $|\bar{\Lambda}_i| = 0$ then:

$$E_{il}(\bar{I}_{il} = \bar{0}) = \min \left\{ \sum_{j \in \partial i \setminus l} E_{ji}(\bar{I}_{ji} = \bar{0}), \right. \tag{2.11}$$
$$\left. \min_{j_1, j_2 \in \partial i \setminus l; \mu \in M} \left[E_{j_1 i}(I^{\mu}_{j_1 i} = +1) + E_{j_2 i}(I^{\mu}_{j_2 i} = -1) + \sum_{k \in \partial i \setminus l, j_1, j_2} E_{ki}(\bar{I}_{ji} = \bar{0}) \right] \right\}$$

$$E_{il}(I^{\mu}_{il} = \pm 1) = \min_{j \in \partial i \setminus l} \left\{ E_{ji}(I^{\mu}_{ji} = \pm 1) + \sum_{k \in \partial i \setminus l, j} E_{ki}(\bar{I}_{ki} = \bar{0}) \right\} + 1 \tag{2.12}$$

If $\Lambda_i^\mu = \pm 1$ then:

$$E_{il}(\bar{I}_{il} = \bar{0}) = \min_{j \in \partial i \setminus l} \left\{ E_{ji}(I_{ji}^\mu = \mp 1) + \sum_{k \in \partial i \setminus l,j} E_{ki}(\bar{I}_{ki} = \bar{0}) \right\} \quad (2.13)$$

$$E_{ji}(I_{ji}^\nu = \pm 1) = +\infty \quad (\nu \neq \mu) \quad (2.14)$$

$$E_{ji}(I_{ji}^\mu = \mp 1) = +\infty \quad (2.15)$$

$$E_{ji}(I_{ji}^\mu = \pm 1) = \sum_{j \in \partial i \setminus l} E_{ji}(\bar{0}) + 1 \quad (2.16)$$

The constant $+1$ that appears equations (2.12) and (2.16) are the costs assigned for a unit of current passing through the considered edge (i.e. $f(1) = 1$). This cost is the one required for the shortest paths but can be generalized to other arbitrary types of costs.

We can interpret these equations in the following way. Equation (2.11) represents the case where i is a transit node and no current passes through edge (ij), then the allowed configurations are that either no currents pass through the remaining neighboring edges (first term inside curly brackets) or one current enters and then exits i through a pair of neighboring edges, all others edges being unused (second term inside brackets). In figure 2.3(left and center) you can see a diagram representing the different allowed configurations for a transit node. Equation (2.12) represents the case where i is a transit node and the communication μ passes through edge (ij); in this case the only allowed configuration is that where the same communication μ enters/exits from one of the other neighboring edges, all others being unused. In figure 2.3(right) this configuration is shown. Similar considerations are used to formulate the equations (2.13-2.16) for senders and receivers.

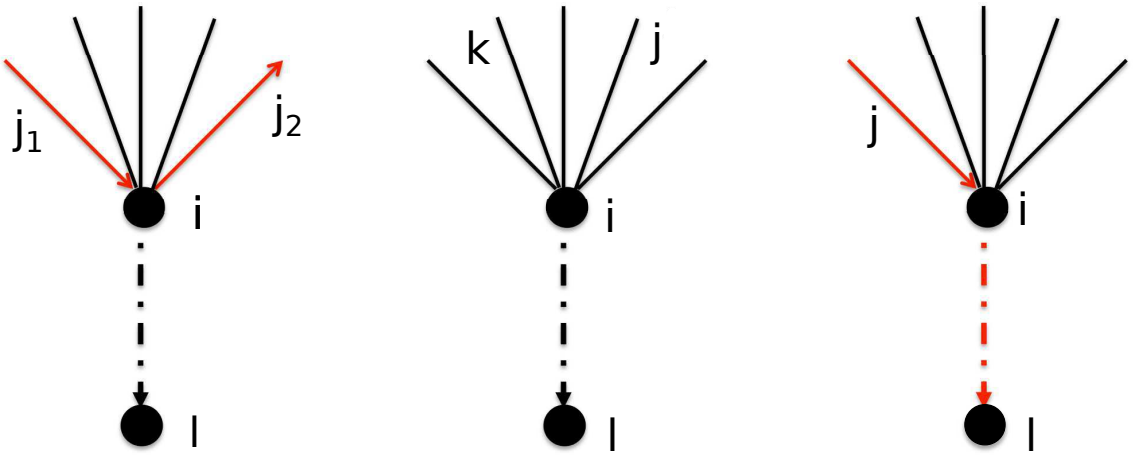


Figure 2.3: Transit node cavity diagram. Left and center represent the two terms inside the min brackets in equation (2.11). Right represents equation (2.12).

Once convergence is achieved one can collect them and obtain the optimal solution using (2.7) and the total path length using (2.8). Notice that, because of the choice of the cost function (2.10), in this case total path length and total cost (2.9) exactly coincide.

2.1.2 Numerical results

For the NDP we study sparse regular, ER and RER random graphs as they are the most interesting for the problem at hand, but the methodology can be easily extended to accommodate other sparsely connected architectures. Clearly, due to the hard constraint of node disjoint

paths, typically no solutions would be found in graphs having a non negligible number of nodes with degree $k = 1, 2$. Moreover, graphs with a small number of high degree nodes (hubs) or with high modularity measure, such as scale-free or planar graphs, are not interesting for the node-disjoint routing problem since when a paths passes through one of these special nodes it leads directly to graph fragmentation, hence frustration. The situation would be very different for constraints on edges instead, but this will be explained in the next section.
From the simulations results we analyzed the statistics of the solutions' path lengths to derive information of the scaling with system size. Then we compared the results of the message-passing with those given by a greedy algorithm. Here we present the main results, leaving the details and further statistical analysis to the paper [37].
The greedy protocol considers only local information around the sources and then builds up a solution step by step recursively, hence reducing considerably the complexity but at the same time completely ignoring other communication positions in the network. A typical greedy algorithm works in the following way: start by choosing an arbitrary pair (S, R), find the shortest path linking the two nodes (we used breast-first search) and then remove nodes belonging to this path from the available network nodes. Choose a second pair and repeat the procedures until either all the M paths from sources to destinations have been established or no solution can be found due to frustration. Clearly, the performance of this algorithm is strictly dependent on the order in which we choose the pairs. For instance, in the extreme case the first pair selected is the one with the longest shortest path among all the M communications; this implies that we have effectively a more restricted graph and choice of paths, leading for a long second path and even more restricted choice of paths later on.

Firstly we collected the total length of the instance for which a solution was found (unfrustrated case). We found the scaling of the total path length to go as a cubic function of $\frac{M \log V}{V \log^{\gamma}(k-1)}$ with exponent γ that depends on the type of graph. A qualitative explanation of the scaling is as follows. The average path length in random graphs goes as $\langle l \rangle \sim \log V / \log\langle k \rangle$ ([38]) and in our case we have M paths to consider. We can refine the dependence on k using instead $\langle l \rangle \sim \log V / \log(k-1)$. Now, suppose all communications take their shortest path, the quantity $x = M \log V / \log^{\gamma}(k-1)$ would be a good estimate of graph occupancy for the NDP, where the exponent γ has been introduced as a free parameter to account for the approximation in the expression for $\langle l \rangle$ as a function of k for different types of graphs. Furthermore, if we divide by the number of available nodes V we can define the occupancy ratio as $\frac{M \log V}{V \log^{\gamma}(k-1)} = x$. Therefore in this simple case we would expect L_{tot}/V increasing linearly in x. If overlaps are prohibited, for a sufficiently high value of M the communications are increasingly forced to take longer routes, leading to a faster than linear increase in the scaling variable x. From numerical simulations we found for the NDP a cubic increase $\frac{L_{tot}}{V} = ax + cx^3$ in the scaling variable $x = \frac{M \log V}{V \log^{\gamma}(k-1)}$. For small M this function agrees well with the linear shortest path behavior but for values of $x > 0.2$ the steeper increase of L_{tot} becomes predominant. These two phases can be distinguished in figures 2.4 where the different topologies are considered and the average connectivity is fixed ($\langle k \rangle = 3$) so that we can fix $\gamma = 0$.

Moreover, we find that the message-passing (MP) always outperforms greedy breadth-first search algorithms not only in finding better solution in terms of path length but also in reaching a higher frustration threshold. This can be seen in figure 2.5 where we plot the failure ratio defined as the number of unsuccessful instances (for which a solution is not found) over the total number of realizations as a function of the scaling variable x. We notice that the greedy algorithm reaches the frustration point (as a function of x) earlier than the corresponding global MP algorithm, regardless the system sizes or graph type. This supports the reasoning that, if a solution exists, a global management of the entire set of communications is required in order to find an optimal solution. Whereas if each communication acts selfishly, seeking the corresponding shortest path, unsolvable overlaps between communications emerge at lower x

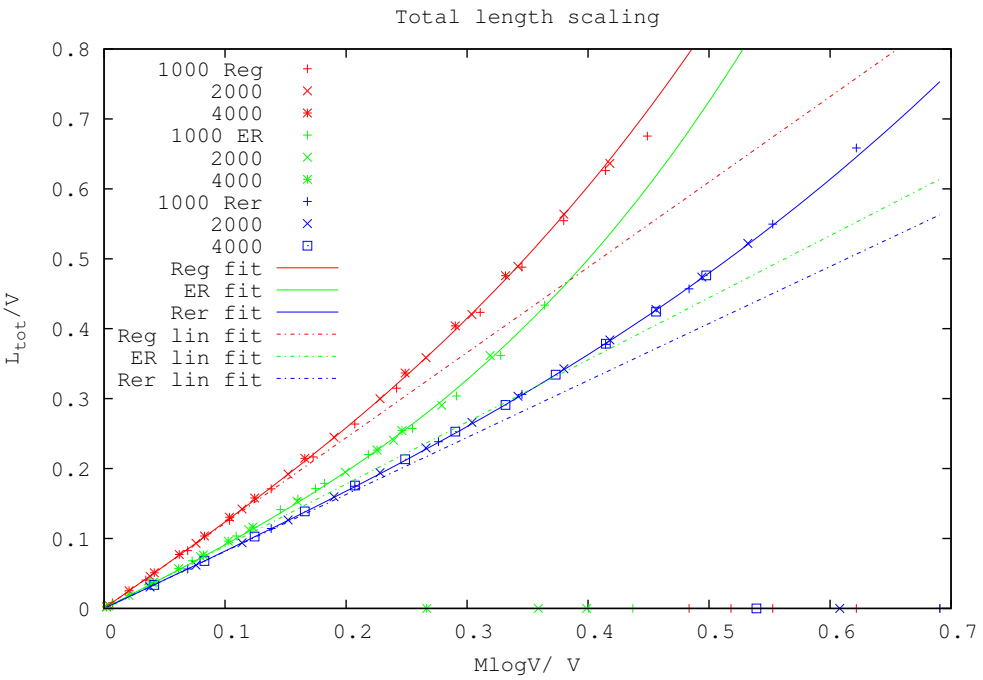


Figure 2.4: Scaling in total length. We plotted the scaling of the total length per node as a function of $M \log V/V$ for Reg, ER and RER graphs of different system sizes; $\langle k \rangle = 3$ for Reg and ER and $\langle k \rangle = 4$ for RER. We can see different slopes in the cubic fits of the various curves, for instance ER graphs achieve shorter lengths but with higher cubic slope, meaning that their length increases faster with traffic due to the smaller number of path choices.

values.

The performance in term of path length is shown in figures 2.6, 2.7 and 2.8, where we plot the total length found by the two algorithms when a solution was found. We can identify the sparse interval where the algorithms give similar results because communications are far apart and take the shortest paths. As the number of communications grow we observe an intermediate regime where global optimisation performs better than greedy; and finally the dense regime where the greedy algorithm fails to find a solution whereas the global optimisation algorithm succeeds up to a critical M value. This shows that a global strategy is needed to optimally route paths which do not overlap at nodes but also have minimal path lengths. Cubic fits are also plotted (solid black line for global optimisation, dashed line for the greedy algorithm) whereas the dotted line represents the shortest path (not considering overlap) trivial solution, i.e. the sum of the M shortest path lengths, which is linear in x. Vertical lines show the frustration points where no solution is found and the total path length is set to zero. Both algorithms show an increased failure rate as the system size increases, presumably due to the unscaled limit on the number of iterations allowed and possibly inherent finite-size effects.

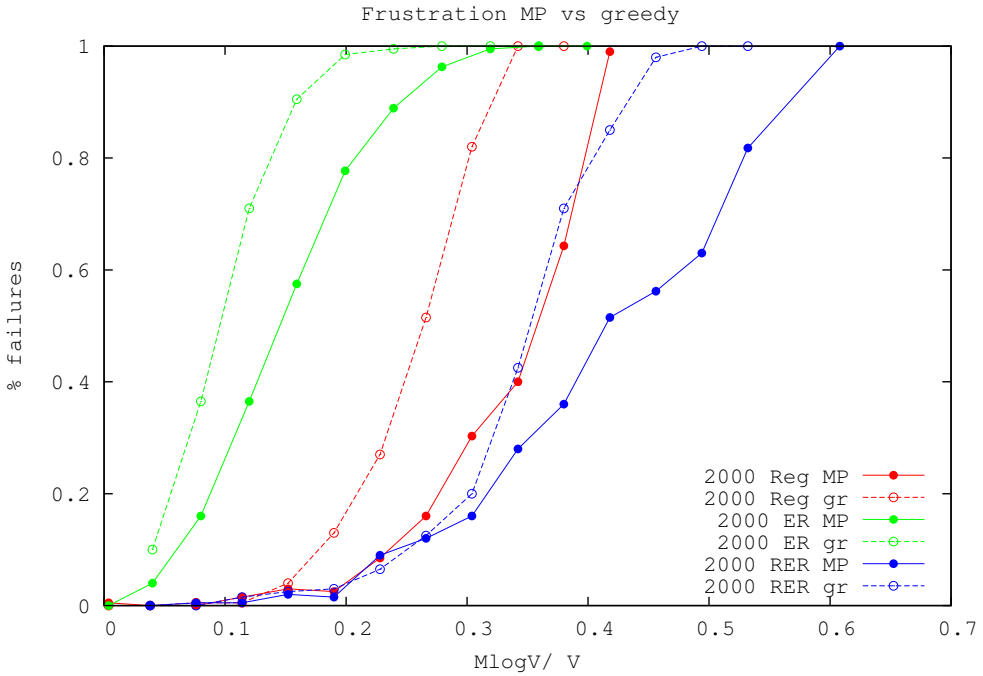


Figure 2.5: Failure rate for greedy and MP algorithms. We plotted for $V = 2000$ and for Reg and ER of degree $\langle k \rangle = 3$ and RER of $\langle k \rangle = 4$, the failure rate as a function of $\frac{M \log V}{V}$. We can notice that the greedy algorithm fails to find solution earlier than MP for all types of graph considered. ER reaches frustration sooner due to less path choices, whereas RER has higher frustration threshold because of the higher connectivity. The MP data shown are results of averages calculated over a smaller number of instances than for the greedy algorithm, hence the lines connecting them are less smooth.

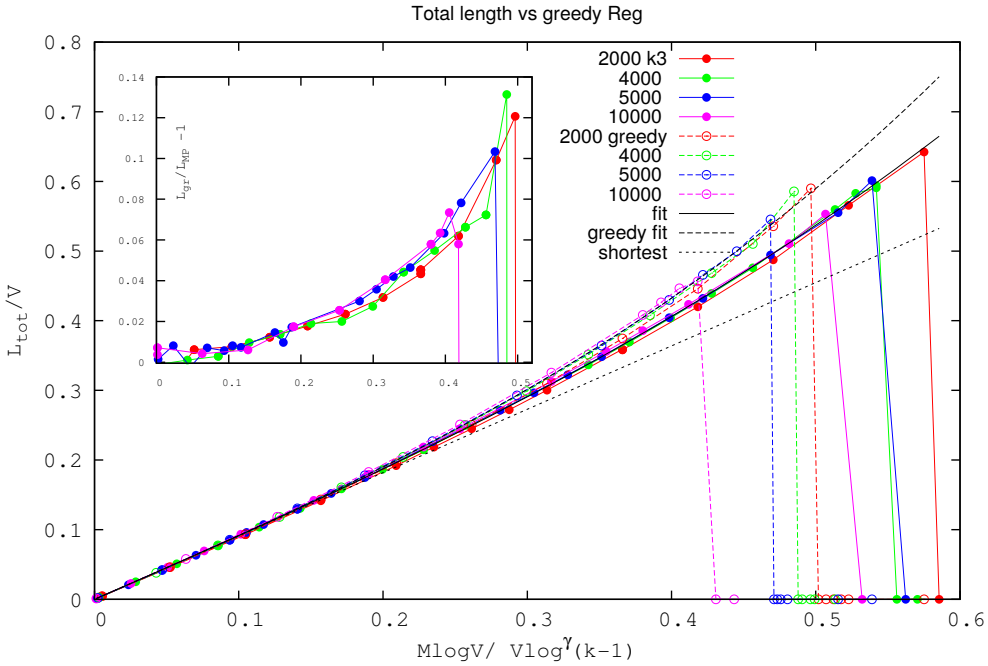


Figure 2.6: Expected total normalized path length - greedy vs global optimisation algorithms - Reg graphs. System size $V = 2000, 4000, 5000, 10000$,$\gamma = 0.87$ and degree $k = 3$. Inset: Ratio $L_{greedy}/L_{MP} - 1$ is plotted as a function of x. Notice the worse performance of the greedy algorithm.

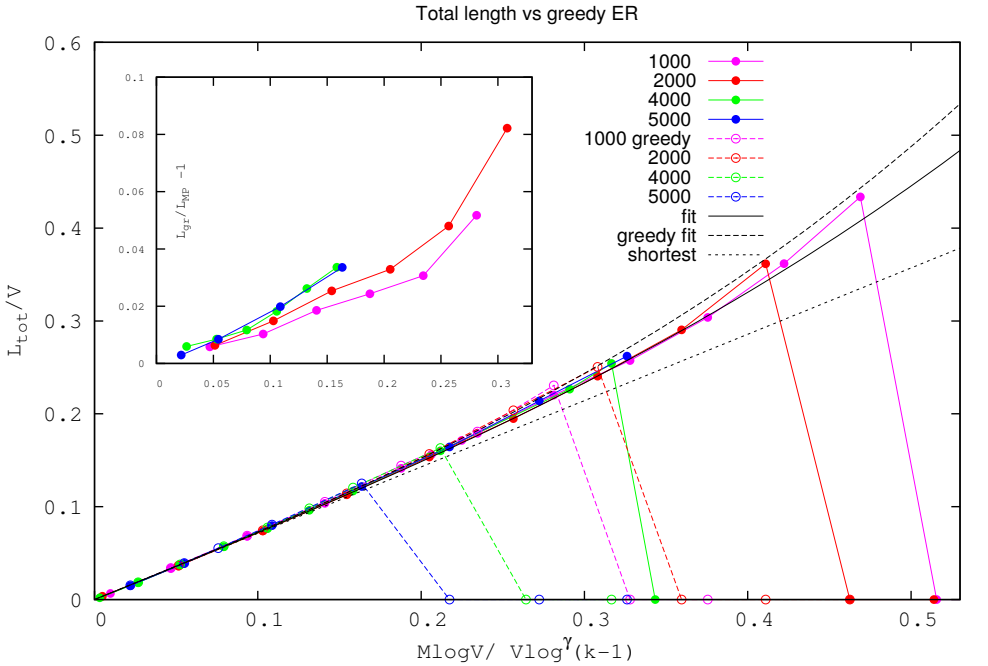


Figure 2.7: Expected total normalized path length - greedy vs global optimisation algorithms ER. System sizes $V = 1000, 2000, 4000, 5000$, $\gamma = 0.69$ and degree $\langle k \rangle = 3$. Inset: the ratio $L_{greedy}/L_{MP} - 1$ is plotted as a function of x. Notice the worse performance of the greedy.

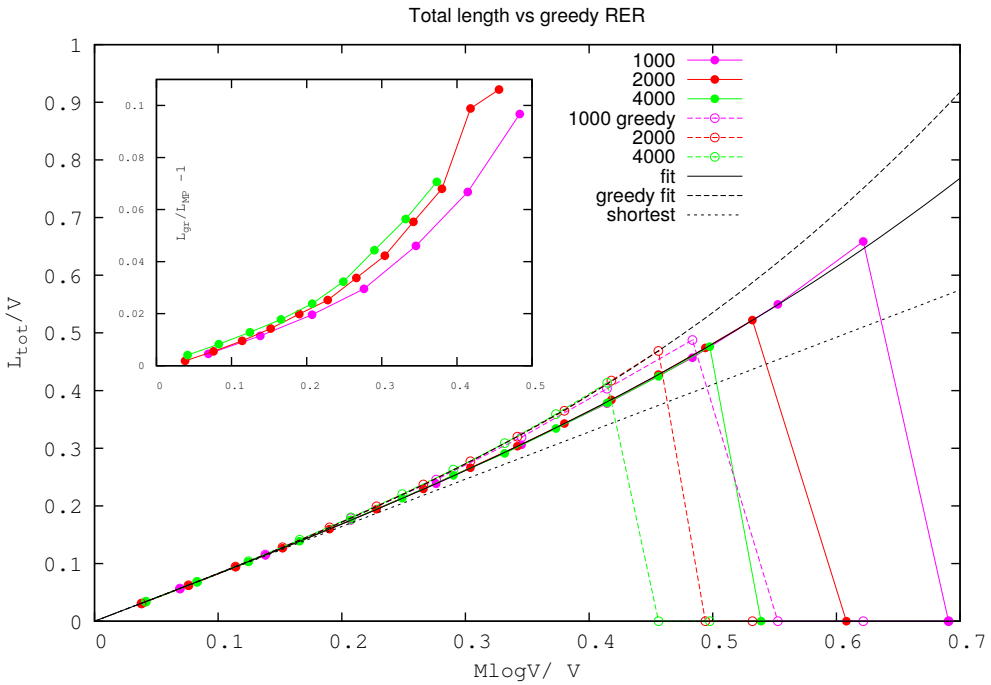


Figure 2.8: Expected total normalized path length - greedy vs global optimisation algorithms RER. System sizes $V = 1000, 2000, 4000$ and degree $\langle k \rangle = 4$. Inset: the ratio $L_{greedy}/L_{MP} - 1$ is plotted as a function of x. Notice the worse performance of the greedy.

2.2 The Edge-Disjoint Path problem

The Edge-Disjoint Path (EDP) problem is a variant of the routing problem where paths cannot overlap at edges. A pictorial representation of the problem is given in figure 2.9.

In this way we relax a bit the hard node-disjointness constraint and allow a bigger number of allowed configurations. Indeed node-disjointness implies edge-disjointness but the reverse is not true. This increase in configuration space forces one to come up with a clever way to run the message-passing equation when searching for the minimal cost neighborhood configuration as in equation (2.6). Here we present an efficient way to perform this minimization by introducing a mapping between the EDP on a given graph and the combinatorial maximum weighted matching [32] on an auxiliary graph.

In this work we focus on the optimisation version of the problem, namely we ask what is the maximum number of communications $M_{acc} \leq M$ that can be accommodated at the same time, under the constraint that paths cannot overlap at edges. Moreover, the additional requirement of minimization of the total path length will be considered.

Apart from a purely theoretical interest [39], it finds various application in communication systems where a high quality of performance and full bandwidth exploitation is required. Indeed routing via edge disjoint paths allows for an efficient bandwidth allocation among users because overlap avoidance means full bandwidth exploitation by each single user. Example of applications are in real-time servers, large-scale video-servers and in admission control virtual circuit routing where one needs to reserve in advance a given path for each communication request so that once the communication is established no interruption will occur. In particular the EDP captures several aspects of the Routing and Wavelength Assignment (RWA) problem in all-optical networks [40]. The aim of this problem is to find the optimal way to route optical communications and the same time the optimal way to assign a wavelength to each of them so that the least number of them will be used. Many algorithms have been proposed by computer

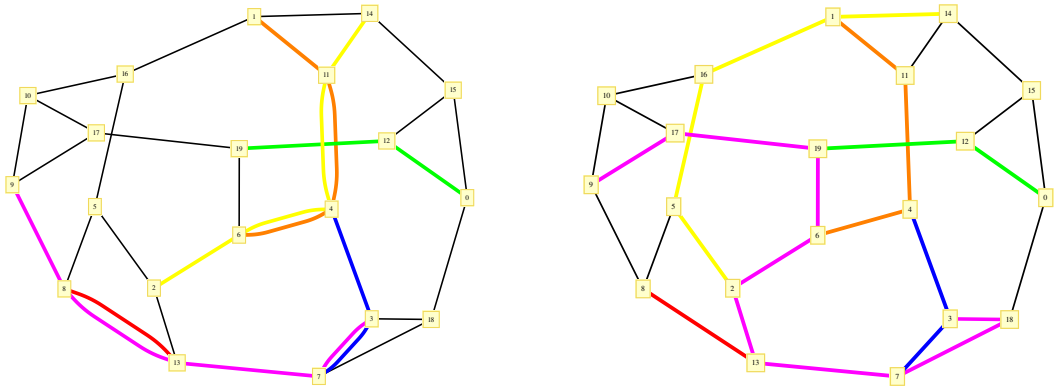


Figure 2.9: An instance of the EDP problem over a 3-regular random graph of $V = 20$ and $M = 6$: examples of solutions of the unconstrained (left) and optimal (right) EDP problem are displayed. In the latter, the purple communication is redirected along a longer path to avoid edge-overlap. The yellow one has two shortest paths of equal length (degeneracy) in the unconstrained case, but once EDP is enforced this degeneracy is broken and only one of the two is optimal (right).

scientists and engineers to tackle the EDP problem but here we focus on three of them. These will be compared with our message-passing algorithm on several benchmark instances. The results of this work can be found in [41].

2.2.1 The EDP model

As for the NDP, also in this case we adopt the general formalism introduced in section 2.0.5 and specialize to the EDP case. Even in this case the disjointness constraint, this time on edges, corresponds to have $||\bar{I}_{ij}|| \in \{0, 1\}$. Again we take the cost function:

$$f(||\bar{I}||) = \begin{cases} \infty & \text{if} \quad ||\bar{I}|| \geq 2 \\ 1 & \text{if} \quad ||\bar{I}|| = 1 \\ 0 & \text{if} \quad ||\bar{I}|| = 0 \end{cases} \tag{2.17}$$

so that the cost function (2.17) represents the total path length. One then has to iterate equation (2.6) until convergence and then collect the results. In order to maximize the number of accommodated paths it is necessary to slightly modify the original instance of the graph $G(\mathcal{V}, \mathcal{E})$ by introducing an extra edge between each pair (S^μ, R^μ) with sufficiently large cost. This will be such that the algorithm could still always find a solution possibly using these expensive extra edges. By construction, the cost of each of these M extra edges should be larger than the maximum possible weight a single path can take. Then the solution of the EDP problem is obtained by discarding the paths passing through the extra edges.
For the EDP problem it is not possible to define a protocol similar to the one used in the NDP. In fact this is prevented by the size of the configuration space: for a given $\bar{I}_{ij}$ the number of possible neighboring currents entering and exiting node i grows exponentially with the degree of that node. Nevertheless, the calculation can be performed efficiently by reducing it to a maximum weight matching problem [32] on an auxiliary weighted complete graph G'_i. Here we explain in more detail this mapping.
The first step is to build the auxiliary graph G'_i. This is a complete graph made of the neighbors

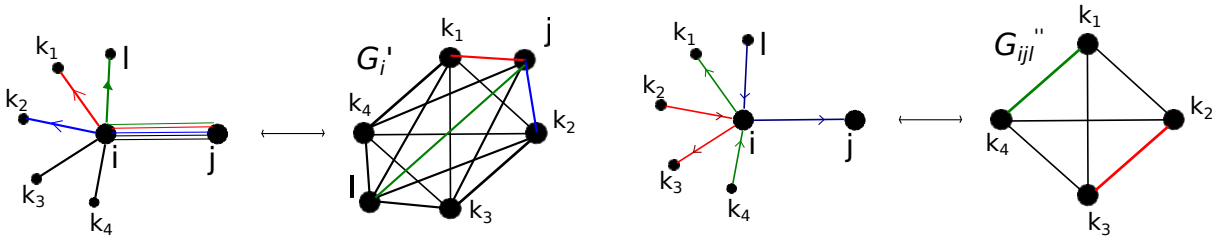


Figure 2.10: Mapping into a weighted matching problem. Left: intermediate step where G'_i is built. On the leftmost part we show an example of several communications passing along (ij) and exiting along the remaining neighbors $k \in \partial i \setminus j$. Right: the final step where G''_{ijl} is built; the best configuration around node i when the blue current passes through (ij) is given by the minimum weighted matching on the complete auxiliary graph G''_{ijl}. Edges red and green represent the best matching, i.e. the configuration where two other communications enter/exit neighbors of $i \setminus j$.

$k \in \partial i$. This graph is weighted with weights on edges defined as entries of a matrix Q:

$$Q_{kl} = -\min_{1 \leq |\nu| \leq M} \{E_{ki}(\nu) + E_{li}(-\nu)\} + E_{ki}(0) + E_{li}(0) \tag{2.18}$$

where we adopted the shortcut $E_{kl}(\nu) = E_{kl}(\bar{I}_{kl})$ with $I^{\mu}_{kl} \equiv \delta_{\mu,\nu}$ for $\nu > 0$, $I^{\mu}_{kl} \equiv -\delta_{\mu,\nu}$ for $\nu < 0$ and $I^{\mu}_{kl} \equiv 0$ for $\nu = 0$. This notation maps the M-dimensional vectors $\bar{I}_{ij}$ to the $2M+1$ possible current configurations ν allowed by the edge-disjointness constraint along a given edge. The computation of matrix Q, that requires $O(Mk^2)$ operations, should be performed only once at the beginning of the update routine for node $i \in G$.
The second step is to focus on a given neighbor $j \in \partial i$ and a given communication μ with the goal to update the message $E_{ij}(\mu)$. In this case we assume that μ is passing through the edge (ij) and therefore, for current conservation, it has to arrive from a second neighbor $l \in \partial i \setminus j$. This will be the optimal one in terms of energy minimization. Then the least costly configuration involving the remaining part of the neighborhood will be:

$$q^{\min}_{jl} = -M_{jl} + \sum_{k \in \partial i \setminus \{j,l\}} E_{ki}(0) \tag{2.19}$$

where M_{jl} is the maximum weight of a matching on a second auxiliary graph that we define as G''_{ijl}. This graph is a complete graph as well, this time obtained from G'_i by removing nodes j and l (and all their incident edges). A matching is a subset of edges of G''_{ijl} that do not share any vertex [32]. The maximum weighted matching is a matching with maximal weight, i.e. maximal sum of weights of the edges belonging to the matching set. Finding M_{jl} is indeed equivalent to assigning to some of the remaining pairs of neighboring nodes currents $\nu \in [-M, \ldots, M]$ that enters through one of them and exits through the other, such that the overall cost of the configuration is minimum. The key point is that the matching condition, i.e. the fact that edges in the solution set cannot have a vertex in common, in our problem translates in the condition of forbidding edge overlaps. The major consequence of this fact is to reduce the computation of the minimum inside equation (2.6) to computing the solution of a standard combinatorial optimisation problem, i.e. the maximum weighted matching. This mapping is represented in figure 2.10.

We can reduce further the computational complexity by a factor of M by noting the following. The possibility that the considered current μ passes also through another pair of neighboring edges (different from the considered (ij) and (il)) is allowed. Nonetheless it will be ruled out by the minimization calculation because this option will have higher cost.
In the previous argument we assumed that the neighbor l was the optimal one where μ has to pass. Indeed the final step of the message-passing update is to minimize over the possible

choices of $l \in \partial i \setminus j$ given the matrix $q^{\min}$. This is defined as the matrix with entries as in (2.19). Eventually on has:

$$E_{ij}^{(t+1)}(\mu) = \min_{l \in \partial i \setminus j} \left\{ E_{li}^{(t)}(\mu) + q_{jl}^{\min} \right\} + c_{ij}(\mu) \tag{2.20}$$

where $c_{ij}(\mu)$ is the cost of edge (ij), that in our case is 0 if $\mu = 0$ and 1 otherwise.
The overall computational complexity required to evaluate equation (2.20) can be split as follows. Firstly, one needs to evaluate the weight matrix Q, this costs $O(Mk^2)$ operations. The matching routine over the complete graphs G''_{ijl} has complexity $O(k^3 \log k)$ [42]. There are $O(k^2)$ possible combinations of j and l in ∂i for which we need to calculate the matching. Putting all together we obtain that the overall complexity of this algorithm will be:

$$O(k^5 \log k + Mk^2)$$

which is polynomial in the variables k and M. Once we have performed this whole procedure, we get all the information we need to calculate the $2M + 1$ update messages $E_{ij}^{t+1}(\mu)$, for each $j \in \partial\, i$, adding a term $O(kM)$ to the final complexity (which is nonetheless negligible compared to the previous two).
Finally we outline how to tackle the special cases where $\Lambda_i^\nu \neq 0$ (node i is either a sender or a receiver for a communication μ) and the case $\nu = 0$. The case $\Lambda_i^\mu \in \{\pm 1\}$ for a given $\mu \in [1, \ldots, M]$ requires the same computation as before but provided that an auxiliary node, indexed by the communication label μ, is added to the original graph G and connected to node i. Its exiting messages will be fixed once at the beginning in the following way and never updated: $E_{\mu i}(\nu) = -\infty$ if $0 < \nu = \mu$ (sender) or $0 < -\nu = \mu$ (receiver), and $E_{\mu i}(\nu) = +\infty$ otherwise. The case of $\mu = 0$, in which no current passes through edge (ij), regardless of what happens on the other edges, is addressed by calculating a matching on a special auxiliary graph. This is the $(k-1)$-node complete graph composed made of all nodes $l \in \partial\, i \backslash j$.

2.2.2 Numerical results

We performed extensive numerical simulations that can be divided into two sets. The first refers to simulations on regular, ER, SF and RER random graphs of various sizes and we compared results with a greedy algorithm similar to the one used for the NDP. This time though we restart n times the greedy (we used $n = 50$) and at the end kept the best results among the n instances. We call this variant multi-start greedy (MSG). In the second set of simulations we run message-passing on benchmark instances over various graphs found in literature [43, 44] and compared the performance with three types of algorithms: an Ant Colony optimisation (ACO), a Montecarlo-like Local Search (LS) and the MSG. Here we give the main results, leaving the details to the paper [41].
Before describing the main results we may outline an efficient strategy to force messages' convergence that we adopted in all the simulations we performed. This is called reinforcement technique [45, 46] and its based on the idea of introducing external local fields on edges so to bias messages to align to them; thus aiding convergence but towards a sub-optimal solution. The magnitude of these external fields is increased at each time step and is set by a parameter that one has to tune. This controls the trade-off between faster convergence and reaching a better solution. The details of this technique are given in appendix 1. The usage of this technique has been particularly useful to solve instances on loopy graphs, where standard message-passing usually fails to converge.

Firstly we found that the message-passing always outperform the greedy both in terms of total path length minimization and in the number of accommodated paths M_{acc}. An example of these results is plotted in figure 2.11 for the case of SF graphs; similar results are seen for

the other topologies. As for the NDP, this shows that an efficient usage of local information is crucial to achieve good performances.

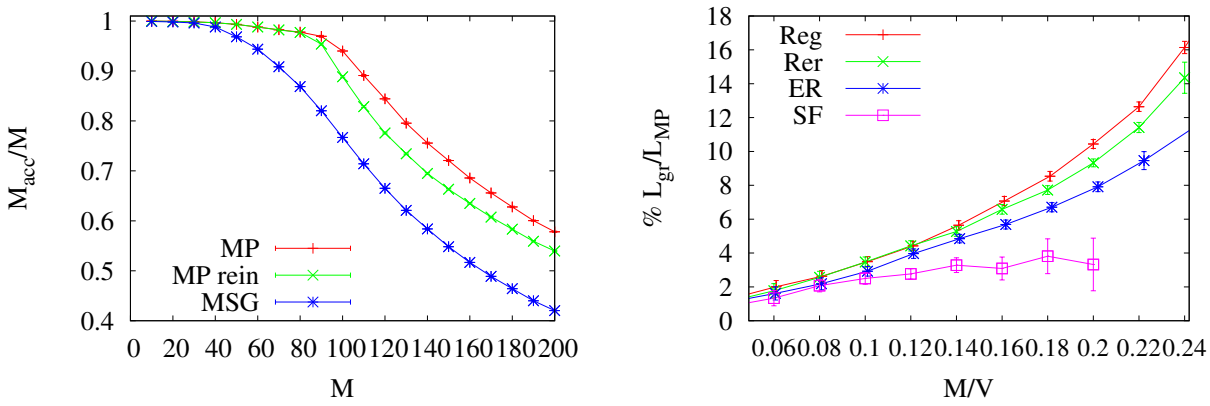


Figure 2.11: MP vs greedy performance. Left: M_{acc}/M for SF of size $V = 10^3$ and average degree $\langle k \rangle = 3$. Error bars are smaller than the size of the symbols. Right: relative total length $y = 100(L_g/L_{MP} - 1)$ between greedy and message-passing. Here L_g and L_{MP} denote the total path lengths calculated with MSG and MP respectively. We use Reg, RER, ER and SF graphs of fixed size $V = 10^3$ and average degree $\langle k \rangle = 5$. The behavior of SF is different from the other due to the topology rich of small-degree nodes. These act as bottlenecks preventing the use of alternative edges and thus causing frustration.

We found a scaling variable $x = \frac{M \log V}{V}$ similar to the NDP case and it can be justified using the same reasoning. In figure 2.12 we plot the scaling behavior with system size of the fraction $1 - M_{acc}/M$ of unaccommodated communications and the average total path length L/V of accommodated paths. We can distinguish two different behaviors depending on x: for small values almost all the paths are accommodated, whereas at some value x^* the curves for different values of V depart from zero. This behavior can be interpreted as a SAT/UNSAT transition, in analogy with the terminology of constraint-satisfaction problems [3]. Notice how the curves collapse in the SAT regime whereas they split in the UNSAT case. Nonetheless the curves with bigger sizes seems to superimpose, suggesting to interpret x as the correct scaling in the thermodynamic limit $V \to \infty$. Therefore this observed mismatch could be due to finite-size effects.

The second set of simulations involves a comparison with two other types of algorithms on benchmark instances. The first one is an Ant Colony optimisation metaheuristic [43]. This method builds an EDP solution incrementally from partial solutions provided by a set of M ants. Each ant generates a path for a given communication making probabilistic decisions during the construction steps. These are made by processing local information modeled as *pheromone* information provided by other ants. The advantage of this method is to divide the EDP in subproblems and to use local information. The drawback is that it relies on several parameters that need to be carefully tuned in order to have a sensitive solution. Moreover the computational time increases considerably with the system size. The second algorithm is a Montecarlo-based Local Search [44], that uses as main Montecarlo step a path rewiring based on rooted spanning trees. Unfortunately the running time grows rapidly with the system size, making it computationally expensive when used on large graphs. In Table 2.1 we report the performance comparison in terms of M_{acc} between the two versions of MP (with and without reinforcement) and the other 3 types of algorithms. We find the the message-passing always performs either equally or better than the other two. Remarkably, the best performances are given on graphs that are highly non locally tree-like as meshes and planar graphs. We would

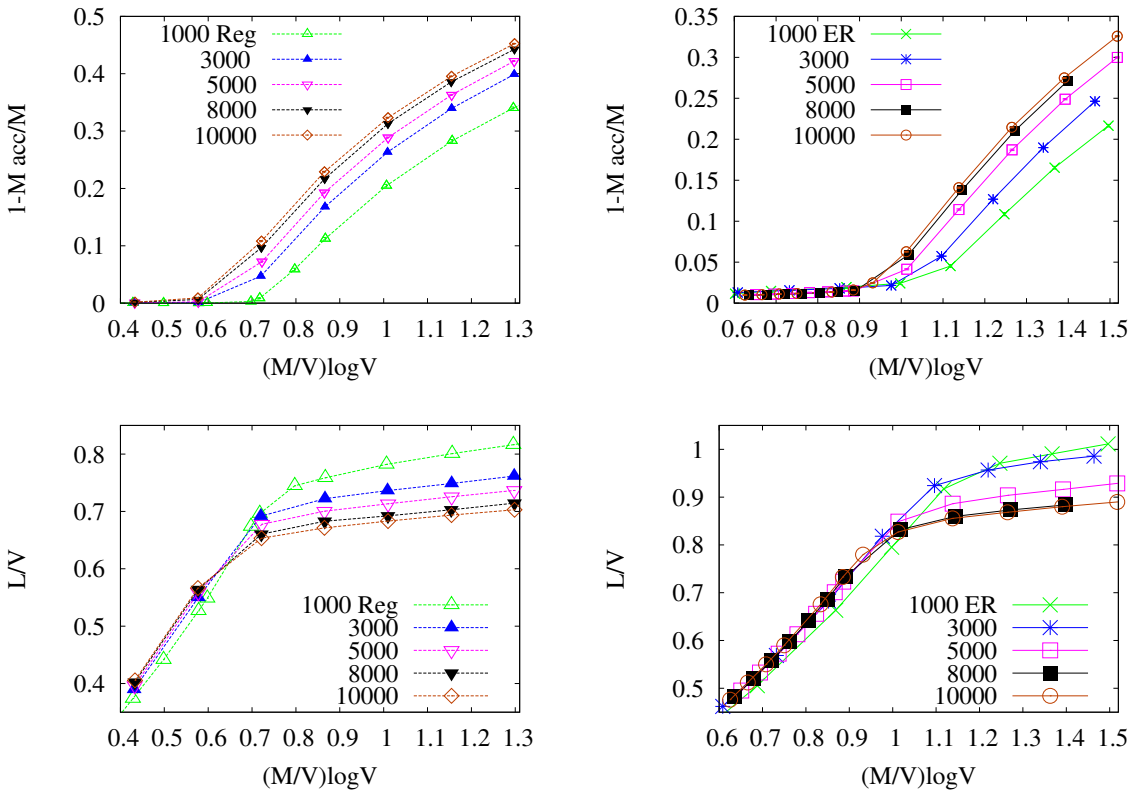


Figure 2.12: Finite-size effects. We plot $1 - M_{acc}/M$ (top) and the total length per node L/V (bottom) for Reg (left) snd ER (right) graphs as a function of the scaling variable $\frac{M \log V}{V}$. We can notice the finite-size effects decreasing with system size leading to the curves corresponding to the biggest graphs $V = 8000$, 10000 to almost superimpose. Note that in the SAT phase the total length grows linearly in $\log V$ for all system sizes as expected but in the UNSAT phase the graphs split. Error bars are smaller than point size.

expect the assumptions behind the cavity method to fail in this cases. Indeed, even though the standard MP fails to converge, the version with reinforcement always finds a solution that is always better than the other algorithms. The larger performance gap is seen on larger set of commodities and bigger graphs. Performance improvement reaches 27% with respect to LS, the best one between the other algorithms tested. The same considerations can be made in the case of planar graphs. We claim that this gap would increase with system size, but unfortunately the size of benchmark graphs remains limited to $V \leq 500$. As a final remark: given these algorithms did not consider path length minimization, we could not compare performance in terms of this quantity.

2.3 The Nash equilibrium problem with congestion

The NDP and EDP problems were both characterized by the hard constraint of disjointness with the consequence of shrinking the configuration space by preventing overlaps. It is then a natural question to ask what could it happen if we relax this constraint further. Namely, what happens if we do allow overlap but penalize (or reward) it. This question has already been addressed using message-passing techniques in [47, 31]. In this case they define a non-linear cost function $f(||\bar{I}_{ij}||)$ such that:

$$f(||\bar{I}_{ij}||) = ||\bar{I}_{ij}||^{\alpha} \tag{2.21}$$

and the goal is to find the optimal configuration that minimizes the global cost function (2.5) that in this case is given by $\sum_{(ij)\in\mathcal{E}} ||\bar{I}_{ij}||^{\alpha}$. Here α is a parameter that one tunes in order to either penalize ($\alpha \geq 1$) or convey ($\alpha < 1$) traffic congestion. An example of the optimal configurations for the two cases is given in figure 2.13. Notice that with this generalization the total cost of a communication coincides with the communication path length only when $\alpha = 1$. In general the cost of communication μ is defined as:

$$d^{\mu}(\bar{\gamma}) = \sum_{(ij)\in\gamma^{\mu}} f(||\bar{I}_{ij}||) \tag{2.22}$$

where $\bar{\gamma}$ is the M-dimensional vector with entires γ^{μ}, the latter representing the path taken by communication μ, i.e. $\gamma^{\mu} = \{(ij) \in \mathcal{E} : I^{\mu}_{ij} = 1\}$.

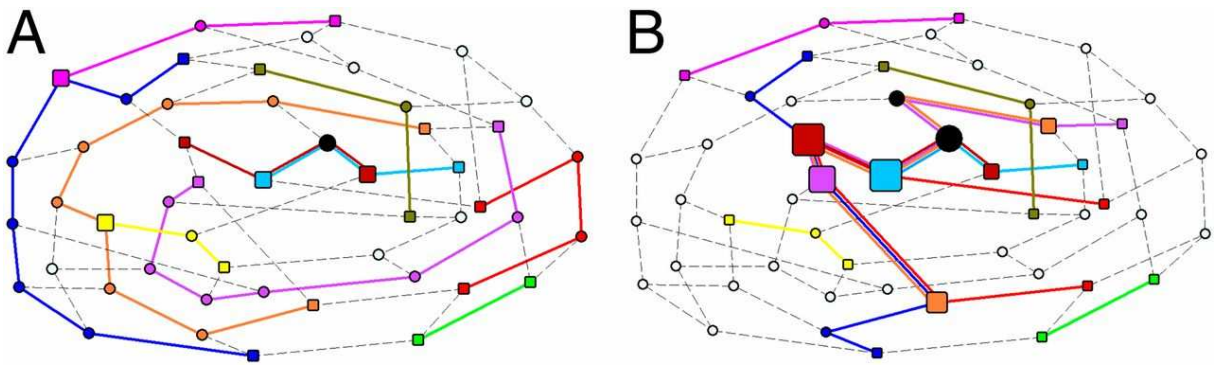


Figure 2.13: Example of congestion problem taken from [31]. The path of each communication is illustrated by nodes and edges of a specific color, whereas black nodes are shared by more than one path. The size of a node is proportional to the amount of traffic through it, and square nodes represent the source or destination of each communication. Left: Optimized solution with $\alpha = 2$, congestion is penalized. Right: optimal solution with $\alpha = 1/2$, congestion is consolidated.

Instance				MP			MP rein = 0.002			MSG (greedy)			ACO			LS			MP gain vs.		
Name	$\|V\|$	$\|E\|$	$\langle k \rangle$	$\langle M \rangle$	M_{min}	M_{max}	$\langle M \rangle$	M_{min}	M_{max}	$\langle M \rangle$	M_{min}	M_{max}	$\langle M \rangle$	M_{min}	M_{max}	$\langle M \rangle$	M_{min}	M_{max}	MSG	ACO	LS
blrand10_1	500	1020	4.08	16.00	16	**16**	16.00	16	**16**	13.65	13	15	14.80	14	**16**	16.00	16	**16**	6.67	0.00	0.00
blrand25_1	500	1020	4.08	32.00	32	**32**	32.00	32	**32**	27.75	26	30	31.85	31	**32**	32.00	32	**32**	6.67	0.00	0.00
blrand40_1	500	1020	4.08	38.00	38	**38**	38.00	38	**38**	33.10	32	35	37.85	37	**38**	37.90	37	**38**	8.57	0.00	0.00
blrand10_2	500	1020	4.08	26.00	26	**26**	25.65	25	**26**	23.85	23	25	25.25	25	**26**	26.00	26	**26**	4.00	0.00	0.00
blrand25_2	500	1020	4.08	35.00	35	**35**	35.00	35	**35**	30.75	29	33	34.75	34	**35**	34.95	34	**35**	6.06	0.00	0.00
blrand40_2	500	1020	4.08	37.00	37	**37**	37.00	37	**37**	32.45	31	34	36.95	36	**37**	36.95	36	**37**	8.82	0.00	0.00
blsdeg10_1	500	1020	4.08	17.00	17	**17**	16.89	15	**17**	14.65	14	16	15.95	15	16	17.00	17	**17**	6.25	6.25	0.00
blsdeg25_1	500	1020	4.08	36.00	36	**36**	36.00	36	**36**	31.55	30	33	35.80	35	**36**	36.00	36	**36**	9.09	0.00	0.00
blsdeg40_1	500	1020	4.08	34.00	34	**34**	34.00	34	**34**	29.00	28	31	33.65	33	**34**	34.00	34	**34**	9.68	0.00	0.00
blsdeg10_2	500	1020	4.08	20.00	20	**20**	19.85	19	**20**	16.90	16	18	19.20	19	**20**	20.00	20	**20**	11.11	0.00	0.00
blsdeg25_2	500	1020	4.08	34.00	34	**34**	34.00	34	**34**	28.45	27	30	32.95	32	**34**	33.90	33	**34**	13.33	0.00	0.00
blsdeg40_2	500	1020	4.08	37.00	37	**37**	37.00	37	**37**	31.75	30	33	36.50	35	**37**	37.00	37	**37**	12.12	0.00	0.00
mesh15_10_1	225	420	3.73	22.00	22	**22**	22.00	22	**22**	20.60	20	**22**	19.65	19	21	21.55	21	**22**	0.00	4.76	0.00
mesh15_25_1	225	420	3.73	36.00	36	**36**	35.10	35	**36**	28.30	27	30	27.70	26	29	32.00	31	33	20.00	24.14	9.09
mesh15_40_1	225	420	3.73	43.00	43	**43**	42.50	42	**43**	30.10	28	32	35.30	32	38	38.80	37	40	34.38	13.16	7.50
mesh15_10_2	225	420	3.73	-	-	-	19.89	19	**20**	19.75	19	**20**	17.50	17	19	19.45	19	**20**	0.00	5.26	0.00
mesh15_25_2	225	420	3.73	35.00	35	**35**	34.70	33	**35**	29.25	29	30	29.20	28	31	33.05	32	34	16.67	12.90	2.94
mesh15_40_2	225	420	3.73	42.00	42	**42**	41.35	41	**42**	29.80	29	32	34.00	33	36	37.60	36	39	31.25	16.67	7.69
mesh25_10_1	625	1200	3.84	-	-	-	47.25	46	**48**	40.70	40	42	32.85	29	36	41.00	39	43	14.29	33.33	11.63
mesh25_25_1	625	1200	3.84	-	-	-	68.30	67	**69**	48.40	47	51	45.00	42	49	55.55	54	59	35.29	40.82	16.95
mesh25_40_1	625	1200	3.84	-	-	-	88.74	88	**90**	54.35	53	58	57.70	53	61	69.30	67	72	55.17	47.54	25.00
mesh25_10_2	625	1200	3.84	-	-	-	44.33	43	**46**	40.05	38	42	30.10	28	33	37.90	36	40	9.52	39.39	15.00
mesh25_25_2	625	1200	3.84	-	-	-	67.22	65	**70**	48.90	47	52	45.60	44	48	54.70	52	59	34.62	45.83	18.64
mesh25_40_2	625	1200	3.84	-	-	-	88.55	87	**90**	54.05	51	57	57.75	54	61	68.85	66	71	57.89	47.54	26.76
steinb4_10	50	100	4.00	5.00	5	**5**	5.00	5	**5**	5.00	5	**5**	5.00	5	**5**	5.00	5	**5**	0.00	0.00	0.00
steinb4_25	50	100	4.00	12.00	12	**12**	12.00	12	**12**	12.00	12	**12**	12.00	12	**12**	12.00	12	**12**	0.00	0.00	0.00
steinb4_40	50	100	4.00	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	19.90	19	**20**	0.00	0.00	0.00
steinb10_10	75	150	4.00	7.00	7	**7**	7.00	7	**7**	7.00	7	**7**	7.00	7	**7**	7.00	7	**7**	0.00	0.00	0.00
steinb10_25	75	150	4.00	18.00	18	**18**	18.00	18	**18**	18.00	18	**18**	17.85	17	**18**	18.00	18	**18**	0.00	0.00	0.00
steinb10_40	75	150	4.00	28.00	28	28	27.65	27	**29**	25.10	24	27	24.35	23	26	27.30	27	28	7.41	11.54	3.57
steinb16_10	100	200	4.00	10.00	10	**10**	10.00	10	**10**	10.00	10	**10**	10.00	10	**10**	10.00	10	**10**	0.00	0.00	0.00
steinb16_25	100	200	4.00	25.00	25	**25**	25.00	25	**25**	25.00	25	**25**	24.35	24	**25**	25.00	25	**25**	0.00	0.00	0.00
steinb16_40	100	200	4.00	36.12	36	**37**	36.00	36	36	33.20	32	34	32.45	32	34	35.95	35	**37**	8.82	8.82	0.00
steinc6_10	500	1000	4.00	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	49.10	47	**50**	50.00	50	**50**	0.00	0.00	0.00
steinc6_25	500	1000	4.00	125.00	125	**125**	122.55	121	124	107.50	106	110	89.90	85	94	104.95	102	108	13.64	32.98	15.74
stienc6_40	500	1000	4.00	145.84	144	**147**	140.40	139	142	114.10	112	117	109.80	106	117	121.40	119	125	25.64	25.64	17.60
steinccl1_10	500	2500	10.00	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	0.00	0.00	0.00
steinc11_25	500	2500	10.00	125.00	125	**125**	125.00	125	**125**	125.00	125	**125**	123.30	122	**125**	125.00	125	**125**	0.00	0.00	0.00
steinc11_40	500	2500	10.00	200.00	200	**200**	200.00	200	**200**	200.00	200	**200**	194.25	190	198	200.00	200	**200**	0.00	1.01	0.00
steinc16_10	500	12500	50.00	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	0.00	0.00	0.00
steinc16_25	500	12500	50.00	-	-	-	125	125	**125**	125.00	125	**125**	125.00	125	**125**	125.00	125	**125**	0.00	0.00	0.00
steinc16_40	500	12500	50.00	-	-	-	200	200	**200**	200.00	200	**200**	200.00	200	**200**	200.00	200	**200**	0.00	0.00	0.00
plan50_10	50	135	5.40	5.00	5	**5**	5.00	5	**5**	5.00	5	**5**	5.00	5	**5**	5.00	5	**5**	0.00	0.00	0.00
plan50_25	50	135	5.40	12.00	12	**12**	12.00	12	**12**	12.00	12	**12**	12.00	12	**12**	12.00	12	**12**	0.00	0.00	0.00
plan50_40	50	135	5.40	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	19.90	19	**20**	0.00	0.00	0.00
plan100_10	100	285	5.70	10.00	10	**10**	10.00	10	**10**	10.00	10	**10**	10.00	10	**10**	10.00	10	**10**	0.00	0.00	0.00
plan100_25	100	285	5.70	25.00	25	**25**	25.00	25	**25**	25.00	25	**25**	25.00	25	**25**	25.00	25	**25**	0.00	0.00	0.00
plan100_40	100	285	5.70	37.00	37	37	37.05	37	**38**	35.80	35	37	34.00	33	36	36.00	35	37	2.70	5.56	2.70
plan200_10	200	583	5.83	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	0.00	0.00	0.00
pan200_25	200	583	5.83	-	-	-	48.95	48	**50**	46.50	46	48	41.80	39	43	45.95	45	48	4.17	16.28	4.17
plan200_40	200	583	5.83	-	-	-	60.65	58	**62**	52.95	52	56	49.35	47	51	55.70	54	58	10.71	21.57	6.90
plan500_10	500	1477	5.91	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	44.95	42	47	50.00	50	**50**	0.00	6.38	0.00
plan500_25	500	1477	5.91	-	-	-	92.29	90	**94**	78.15	76	80	60.95	57	65	78.20	77	80	17.50	44.62	17.50
plan500_40	500	1477	5.91	-	-	-	122.31	119	**124**	92.60	90	95	82.85	78	86	100.15	97	102	30.53	44.19	21.57

Table 2.1: Message-passing and multi-start greedy performances. Columns 1-4 give the characteristics of the benchmark. For each algorithm, columns 1-3 represent the average, the minimum and the max number of accommodated paths over 20 runs of a given set of commodity instance respectively. ACO and LS performances are reported in [43, 44]. Performance comparison between MP and the other algorithms is given in the three last columns, representing the performance ratio $100 \cdot (M_{acc}^{BP}/M_{acc}^{alg} - 1)$ where alg indicates the algorithm used (MSG, ACO and LS respectively). We use as M_{acc}^{MP} the best one between MP with and without reinforcement.

This edge cost f can be generalized to any (non pathological) cost function to address specific needs. They have compared results obtained with their proposed message-passage routine with both a greedy Dijkstra [48] and a multi-commodity flow algorithm. They found a gain in cost of around 20% with respect to the Dijkstra and a smaller one when compared to the multi-commodity. The great advantage of the message-passing is that not only is distributed, uses local information, and does not require fine-tuning of parameters, but is flexible with respect to the choice of cost function. This is important in particular in the case of concave cost functions, i.e. when traffic congestion is rewarded, where multi-commodity flow is not applicable.

In that work they were interested in finding the global optimum. Even though the premises are the same, in this work we ask a different question: what is (if it exists) the Nash equilibrium solution? For Nash equilibrium we mean the concept defined in game theory where each player (communication) will not gain by changing is strategy (path) if all the others' remain the same [49]. This can be formalized by defining $c(\sigma_i|\bar{\sigma}_{\partial i})$ as the cost payed by the player i to adopt the strategy (the variable) σ_i given the neighbors' configuration is $\bar{\sigma}_{\partial i}$. A configuration $(\sigma_i^*, \bar{\sigma}_{\partial i}^*)$ is a pure Nash equilibrium [49] when:

$$c(\sigma_i^*|\bar{\sigma}_{\partial i}^*) = \min_{\sigma_i} c(\sigma_i|\bar{\sigma}_{\partial i}^*) \tag{2.23}$$

Briefly, each player makes the best decision based on his knowledge on the others' strategies. The difference is subtle. Indeed the global optimum is when no single user can be better-off by a change of path without causing the others' paths to get more costly. Therefore in general Nash equilibrium and global optimum do not coincide.
The main challenge concerning these two generalization of the routing problem (global optimum and Nash equilibrium) is how to find the solution given the high number of communications involved and the size of the network. The configuration space is exponential in the number of communications therefore one needs to find a clever way to run through the neighbors configuration in the calculation of the minimum inside (2.6). In [31] they adopted a mean-field approximation. At each iteration step they calculated a mean-field current λ_{ki}^{MF} along edge $(ki) \in \mathcal{E}$ representing the optimal configuration of communications passing along that edge, based on the last message-passing update. The idea is to then update a message $E_{ij}^{\mu}(\bar{I}_{ij})$ $(\forall \mu)$ approximating the currents passing along the neighborhood with the set of $k_i - 1$ mean-field currents $\{\lambda_{ki}^{MF}\}_{k\in\partial i\setminus j}$. We will adopt a similar strategy adapted to a Nash equilibrium formulation in the next sections. We should remark that this is an on-going work [50], for which the theoretical framework has been set up but results from numerical simulations are still to be obtained. Therefore in this thesis we will focus on the theoretical description of the model leaving the numerical tests for a future manuscript [50].

2.3.1 The game theoretical formulation

Here we give the main definitions and principles needed to formulate our problem in the context of game theory. Our model of routing will be introduced only in the next section.
The Nash equilibrium version of the routing problem is often called congestion game in the context of game theory and was first proposed by Rosenthal in 1973 [51]. In this context it is often convenient to define a so called *potential function* $\Phi(\bar{\gamma})$. This is a real valued function of the M-dimensional vector made of the communication paths $\bar{\gamma}$, where each of its entries represents the path taken by communication μ, i.e. $\gamma^{\mu} = \{(ij) \in \mathcal{E} : I_{ij}^{\mu} = 1\}$.
Suppose for a moment that a given communication μ changes its path from γ_1^{μ} to γ_2^{μ}, while the other communications do not change. If the change in cost for the communication $\Delta d^{\mu} = d^{\mu}(\gamma_2) - d^{\mu}(\gamma_1)$ is equivalent to the change of the potential function $\Delta\Phi = \Phi(\gamma_2^{\mu}, \bar{\gamma}^{-\mu}) - \Phi(\gamma_1^{\mu}, \bar{\gamma}^{-\mu})$, where $\bar{\gamma}^{-\mu}$ is the $(M-1)$-dimensional vector where μ is not present, then the

potential function is said to be *exact*. This definition can be generalized to the one of an *ordinal potential* in the case where we don't have an exact equality. In fact if we change a communication path μ such that: $\Delta d^{\mu} < 0$ implies $\Delta\Phi < 0$, then Φ is said to be an ordinal potential. These two definitions are useful because potential functions have nice properties, the most important one being that they admit a pure Nash equilibrium, this is the strategy $\bar{\gamma}^*$ that minimizes $\Phi(\bar{\gamma})$ [52]. Rosenthal proved [51] that a congestion game always admits a potential function. Then, if we start from a given set of paths $\bar{\gamma}$, and at each step which change one of them so to reduce its cost, then also $\Phi(\bar{\gamma})$ will be reduced equally. Given that the potential admits a finite amount of values, eventually it will reach the minimum. Indeed we can define an exact potential as:

$$\Phi(\bar{\gamma}) = \sum_{(ij)\in\mathcal{E}} \sum_{k=1}^{n_{ij}(\bar{\gamma})} f(k) \tag{2.24}$$

where $n_{ij}(\bar{\gamma})$ is the max number of communications passing along edge (ij) given the set of communications $\bar{\gamma}$, i.e. it is equal to $||\bar{I}_{ij}||$ given the set $\bar{\gamma}$; f is the cost as defined in (2.21). The second sum in (2.24) allows to write $\Delta d^{\mu} = d^{\mu}(\gamma_2) - d^{\mu}(\gamma_1) = \Phi(\gamma_2^{\mu}, \bar{\gamma}^{-\mu}) - \Phi(\gamma_1^{\mu}, \bar{\gamma}^{-\mu}) = \Delta\Phi$, this equality being valid for each change involving a single communication from γ_1^{μ} to γ_2^{μ}.
The spin-glass model [24] on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an example of a game that admits an exact potential function. In this model a variable $\sigma_i \in \{-1, +1\}$ is assigned to each node $i \in \mathcal{V}$ of the graph and its cost is represented by $h_i = \sigma_i \sum_{j\in\partial i} J_{ij}\sigma_j$. The set of random variables $\{J_{ij}\}_{(ij)\in\mathcal{E}}$ represents the disorder. One can then define the potential function $\Phi(\bar{\sigma}) = \sum_{j<i} J_{ij}\sigma_i\sigma_j$ and verify that for each arbitrary single-spin flip $\sigma_i \to \sigma_i' = -\sigma_i$ we have $\Delta h_i = \Delta\Phi$. Therefore it is an exact game.
Even though in principle one could reach the Nash equilibrium by performing a set of successive changes to a single communication path at each step, a strategy that is called *best response*, this could take an exponential number of them. Suppose one could define the *neighbor* of a solution of a game, in our case a set $\{\bar{I}_{ij}\}_{(ij)\in\mathcal{E}}$ such that the constraints are satisfied, as another solution which is somehow close to the previous one (one has to specify what is meant by close). For example, in our case a neighbor could be the same set $\{\bar{I}_{ij}\}_{(ij)\in\mathcal{E}}$ but this time one of the communications takes another path, still respecting the constraints. In general this other solution has a different cost (or energy). Then one can define a local protocol where, starting from a given configuration, one replaces it with another one belonging to its neighborhood if this other costs less. One iterates until no neighbor could be found with less cost. This means that a local optimum has been reached. A typical example is the simulated annealing [53]. Assuming that the best response of a player can be computed in polynomial time, it has been shown [54] that the problem of finding Nash equilbria in a congestion game is among the PLS-complete problems [55]. This means that firstly it is among the complexity class PLS (Polynomial Local Search) [52]. Namely, one has to be able to establish if a given solution is locally optimal in polynomial time; if it is not, one should be able to find a less costly one in its neighborhood. PLS-Complete means that there exists a polynomial time reduction from all the other PLS problems to the considered game, such that the local optima of the target problem coincide with the ones of the original game. From these definitions it follows that congestion games, which are in PLS-Complete, do have local optima. However, given that the configuration space could be exponential, it could take exponential time to reach these local optima.

2.3.2 The model

In spite of what was needed in the formulation of the NDP and EDP, in the Nash equilibrium version of the problem it is not enough to parametrize the model using only one variable $\bar{I}_{ij}$.

This time we then define I_{ij} as the net flux passing through an edge $(ij) \in \mathcal{E}$, i.e. it is the equivalent of $||\bar{I}_{ij}||$ of the previous two formulations. For clarity it is convenient to parametrize whether a communication passes or not through an edge using a different variable σ_i^μ (c.f. I_i^μ) so that:

$$\sigma_i^\mu = \begin{cases} 1 & \text{if } \mu \text{ flows from } i \text{ to } j \\ -1 & \text{if } \mu \text{ flows from } j \text{ to } i \\ 0 & \text{if } \mu \text{ does not pass trough } (ij) \end{cases} \tag{2.25}$$

Therefore $I_{ij} = \sum_{\mu=1}^{M} |\sigma_{ij}^\mu|$. One could then define the M-dimensional vector $\bar{\sigma}_{ij} = (\sigma_{ij}^1, \dots, \sigma_{ij}^M)$ denoting the currents' configuration along edge (ij).
The cost paid by each communication to pass along an edge is:

$$c_{ij}^\mu = |\sigma_{ij}^\mu| \phi(I_{ij}) \tag{2.26}$$

where $\phi(I_{ij})$ is a function of the net flow, it could either penalize or favor traffic congestion. Then the global cost of edge (ij) is the sum of each communication's cost: $c_{ij}(I_{ij}) = I_{ij}\phi(I_{ij})$. Finally the total cost of a configuration $\{\bar{\sigma}\}$ is:

$$c(\{\bar{\sigma}\}) = \sum_{(ij)\in\mathcal{E}} I_{ij}\phi(I_{ij}) \tag{2.27}$$

Notice that, with this definition, a linear edge cost $\phi(I_{ij}) = I_{ij}$ would correspond to a non-linear (quadratic) global cost. Hence is enough to fix it linear to obtain a rich behavior. Even in this case one constraint is given by Kirchhoff law at nodes using the same definition Λ_i^μ as in (2.2):

$$\sum_{\jmath \in \partial\, i} \sigma_{ij}^\mu - \Lambda_i^\mu = 0 \qquad \forall \mu = 1, \dots, M \tag{2.28}$$

However this time the constraint characterizing the problem is the Nash constraint: for each node we have to verify that it has the minimal distance (in terms of the cost we defined above) from the source:

$$d_i^\mu = \min_{k\in\partial i}\{d_k^\mu + \phi(I_{ki}^{-\mu} + 1)\} \tag{2.29}$$

where we introduced the variable d_i^μ defined as the distance of node i from the source of communication μ. The variable $I_{ki}^{-\mu}$ represents the total flux on edge (ij) in the absence of communication μ: this is a common notation in game theory where each player estimates the others' behavior and then adapts its strategy consequently. With this in mind, the $+1$ term in (2.29) takes in consideration the presence of μ itself along that edge. This definition of distance should not be confused with the path length. In fact the two would coincide only if $\phi(I)$ is linear. Also in this case one could define an M-dimensional vector $\bar{d}_i$ which has entries as in (2.29).
In order to enforce the Nash constraint, along with the Kirchhoff law, it is convenient to use the vector $(\bar{d}_i, \bar{d}_j, \bar{\sigma}_{ij}, I_{ij})$ as the variable node in the factor graph representation. We then have one type of function nodes Ψ_i acting on the neighborhood of a node i that represent the two constraints; then we have another type of function node ψ_{ij} that acts only on one variable node and this represents both the cost function $c_{ij}(I_{ij})$ and the condition $I_{ij} = \sum_{\mu=1}^{M} |\sigma_{ij}^\mu|$ (this could be interpreted as another constraint). This can be seen as a field acting on the edge (ij). The factor graph is displayed in figure 2.14.

With this setting we can formulate the problem using the *max-sum* equations corresponding to (2.6) but this time we take minus the cost (so the minimum becomes a maximum) and we use two types of messages. One is used to enforce the constraint $I_{ij} = \sum_{\mu=1}^{M} |\sigma_{ij}^\mu|$, whereas the other is the standard one of the message-passing formalism:

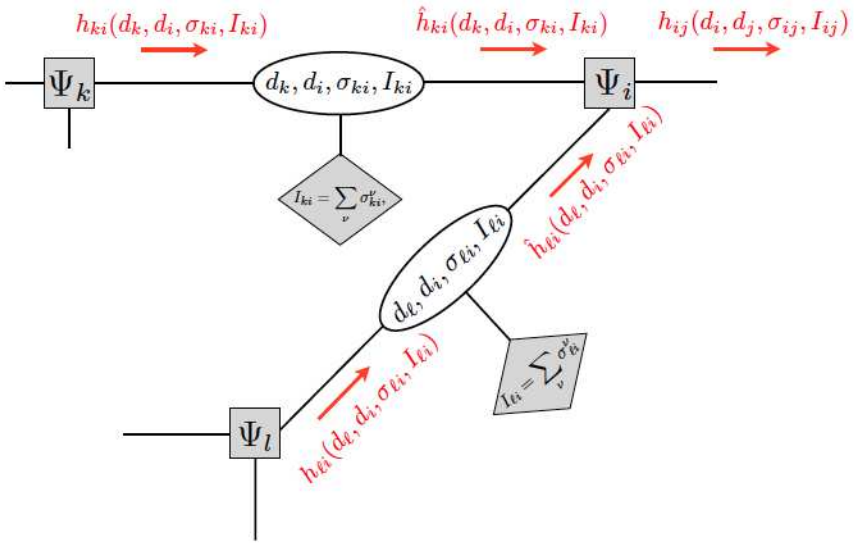


Figure 2.14: Factor graph representation of the Nash equilibrium problem. Variable nodes are the vectors $(\bar{d}_i, \bar{d}_j, \bar{\sigma}_{ij}, I_{ij})$; function nodes are of two types: Ψ_i are acting on the whole neighborhood of i and represent Nash and Kirchhoff constraints; ψ_{ij} acts only on one variable node and represents the cost function and the condition $I_{ij} = \sum_{\mu=1}^{M} |\sigma_{ij}^\mu|$.

$$h_{ij}(\bar{d}_i, \bar{d}_j, \bar{\sigma}_{ij}, I_{ij}) = \max_{\substack{\bar{d}_k, \bar{\sigma}_{ki}, I_{ki}, k \in \partial i \setminus j : \\ constraints}} \left\{ \sum_k \hat{h}_{ki}(\bar{d}_k, \bar{d}_i, \bar{\sigma}_{ki}, I_{ki}) - I_{ij}\phi(I_{ij}) \right\} \quad (2.30)$$

$$\hat{h}_{ij}(\bar{d}_i, \bar{d}_j, \bar{\sigma}_{ij}, I_{ij}) = \begin{cases} h_{ij}(\bar{d}_i, \bar{d}_j, \bar{\sigma}_{ij}, I_{ij}) & \text{if} \quad I_{ij} = \sum_{\mu=1}^{M} |\sigma_{ij}^\mu| \\ -\infty & \text{otherwise} \end{cases} \quad (2.31)$$

Notice that the term $-I_{ij}\phi(I_{ij})$ has to be included inside the max because the constraint $I_{ij} = \sum_{\mu=1}^{M} |\sigma_{ij}^\mu|$ has to be enforced as well as the others. Solving this equation is highly non-trivial. The major difficulty comes from the presence of the term I_{ij} that couples together all the communications, thus preventing to split the message h_{ij} into a sum of M terms. This can be seen by making the following assumption:

$$h_{ij}(\bar{d}_i, \bar{d}_j, \bar{\sigma}_{ij}, I_{ij}) = \sum_\nu h_{ij}^\nu(d_i^\nu, d_j^\nu, \sigma_{ij}^\nu, I_{ij}) \quad (2.32)$$

and plugging it inside both sides of (2.30) (and considering the equivalent assumption valid for (2.31). We obtain:

$$\sum_\nu h_{ij}^\nu(d_i^\nu, d_j^\nu, \sigma_{ij}^\nu, I_{ij}) = \max_{\substack{\bar{d}_k, \bar{\sigma}_{ki}, I_{ki}, k \in \partial i \setminus j : \\ constraints}} \left\{ \sum_\nu \sum_k \hat{h}_{ki}^\nu(d_k^\nu, d_i^\nu, \sigma_{ki}^\nu, I_{ki}) - \sum_\nu |\sigma_{ij}^\nu| \phi(I_{ij}) \right\} \quad (2.33)$$

The flow I_{ki} inside the the first sum on the right-hand side of this equation prevents from splitting the max into a sum of max over the different $\hat{h}^\nu$. In order to allow this decoupling we need to introduce a mean-field approximation.
Suppose indeed that along the neighbors $k \in \partial i \setminus j$, instead of having a variable flux I_{ki}, the

current is fixed to $\hat{I}_{ki} = \sum_\mu |\hat{\sigma}^\mu_{ki}|$, where the $\hat{\sigma}^\mu_{ki}$ are the corresponding background communications contributing to the total background flow. We call this current the *background current*. This can be considered as the optimal flow passing along the neighbor estimated using the messages resulting from the last iteration. This background current acts as a mean-field and allows us to decouple the messages as:

$$h^\nu_{ij}(d^\nu_i, d^\nu_j, \sigma^\nu_{ij}, I_{ij}) = -|\sigma^\nu_{ij}|\phi(I_{ij}) + \max_{d^\nu_k, \sigma^\nu_{ki}: constraints} \sum_{k\in\partial i\setminus j} \hat{h}^\nu_{ki}(d^\nu_k, d^\nu_i, \sigma^\nu_{ki}) \tag{2.34}$$

$$\hat{h}^\nu_{ki}(d^\nu_k, d^\nu_i, \sigma^\nu_{ki}) = h^\nu_{ki}(d^\nu_k, d^\nu_i, \sigma^\nu_{ki}, \hat{I}^{-\nu}_{ki} + |\sigma^\nu_{ki}|) \tag{2.35}$$

Notice the term $\hat{I}^{-\nu}_{ki}$ appearing in the second equation. This is the background flow in the absence of communication ν, i.e. we have $\hat{I}_{ki} = \hat{I}^{-\nu}_{ki} + |\sigma^\nu_{ki}|$. This definition of background flow can be formalized with a self-consistent pair of equations:

$$\hat{I}_{ij} = \sum_\mu |\hat{\sigma}^\mu_{ij}| \tag{2.36}$$

$$\hat{\sigma}^\mu_{ij} = \arg\max_{\sigma^\mu_{ij}} \max_{d^\mu_i, d^\mu_j} h^\mu_{ij}(d^\mu_i, d^\mu_j, \sigma^\mu_{ij}, \hat{I}_{ij}) \tag{2.37}$$

Notice that the background current $\hat{I}_{ij}$ and the corresponding $\hat{\sigma}^\mu_{ij}$ are not symmetric. This is given by the fact that the message h_{ij} itself is not symmetric. In fact h_{ij} and h_{ji} represent two different cavity subgraphs generating from the removal of edge (ij) from the original graph. The equations (2.36) and (2.37) represent the optimal configuration (under the mean-field approximation) along edge (ij) in the cavity graph, subsequent to the last cavity update.
This time the algorithm makes one additional step needed to calculate the background current. We can summarize the algorithmic procedure as follows:

- Initialize randomly the messages $h^\nu_{ij}(d^\nu_i, d^\nu_j, \sigma^\nu_{ij}, I_{ij})$
- Calculate the sets of background currents $\{\hat{I}_{ij}\}$ and $\{\hat{\sigma}^\mu_{ij}\}$ $\forall (ij) \in \mathcal{E}$
- Pick arbitrarily an edge $(ij) \in \mathcal{E}$ and update using (2.34) and (2.35)
- Calculate the new background currents on (ij) using (2.36) and (2.37)
- Repeat for all other edges
- Repeat the whole procedure until messages $h^\nu_{ij}(d^\nu_i, d^\nu_j, \sigma^\nu_{ij}, I_{ij})$ converge

Upon reaching convergence, one collects the converged messages $h^{*\nu}_{ij}(d^\nu_i, d^\nu_j, \sigma^\nu_{ij}, I_{ij})$ and calculates the optimal configuration and its cost. The first one is obtained by combining the in-going and out-going messages on a given edge:

$$I^*_{ij} = \sum_\mu |\sigma^{*\mu}_{ij}| \tag{2.38}$$

$$\sigma^{*\mu}_{ij} = \arg\max_{\sigma^\mu_{ij}} \max_{d^\mu_i, d^\mu_j} \left\{ h^{*\mu}_{ij}(d^\mu_i, d^\mu_j, \sigma^\mu_{ij}, I_{ij}) + h^{*\mu}_{ji}(d^\mu_j, d^\mu_i, -\sigma^\mu_{ij}, I_{ij}) + |\sigma^\mu_{ij}|\phi(I_{ij}) \right\} \tag{2.39}$$

where the last term is added to avoid cost double counting. Notice that this time the resulting currents $\sigma^{*\mu}_{ij}$ and I^*_{ij} are indeed symmetric. The optimal distances $d^{*\mu}_i$ will be the one used to calculate (2.39):

$$(d^{*\mu}_i, d^{*\mu}_j) = \arg\max_{d^\mu_i, d^\mu_j} \max_{\sigma^\mu_{ij}} \left\{ h^{*\mu}_{ij}(d^\mu_i, d^\mu_j, \sigma^\mu_{ij}, I_{ij}) + h^{*\mu}_{ji}(d^\mu_j, d^\mu_i, -\sigma^\mu_{ij}, I_{ij}) + |\sigma^\mu_{ij}|\phi(I_{ij}) \right\} \tag{2.40}$$

Finally the optimal global cost will be:

$$c^* = \sum_{(ij)\in\mathcal{E}} I^*_{ij}\phi(I^*_{ij}) \tag{2.41}$$

The actual implementation of the cavity equations poses some numerical challenges. The major one is how to enforce the constraints efficiently, i.e. without running an extensive search through the configuration space. Indeed enforcing Nash and Kirchhoff has the effect of coupling the variables d_i^μ and σ_{ij}^μ together. Think for instance at the case where for an internal node i (i.e. $\Lambda_i^\nu = 0$) we have incoming and outgoing flows obeying Kirchhoff law: $\exists k_1, k_2 \in \partial i \mid \sigma_{k_1 i}^\nu = +1, \sigma_{k_2 i}^\nu = -1$ and $\forall k \in \partial i \setminus k_1, k_2$ we have $\sigma_{ki}^\nu = 0$. In this case if we want to obey also Nash equilibrium (i.e. $d_i^\nu = \max_{k\in\partial i}\left[d_k^\nu + \phi(I_{ki}^{-\nu}+1)\right]$) we need $d_i^\nu = d_{k_1}^\nu + \phi(I_{k_1 i}^{-\nu}+1)$, $d_{k_2}^\nu = d_i^\nu + \phi(I_{ik_2}^{-\nu}+1)$ and $\forall k \in \partial i \setminus k_1, k_2$ we need $d_k^\nu + \phi(I_{ki}^{-\nu}+1) \geq d_{k_1}^\nu + \phi(I_{k_1 i}^{-\nu}+1)$.

Another non-trivial task is how to parametrize the variables. In fact for a communication made of r edges, d_i^μ could take in principle any value from 0 to $r\phi(M)$. The latter representing the case where the communication path is crowded with M communications on all the r edges. Practically though, this range is reduced because it is quite unlikely that d_i^μ will be near $r\phi(M)$, this is due to the Nash constraint (2.29). Therefore it is reasonable to assume that the length is bounded to have maximal value $d_{max} \ll r\phi(M)$, thus a possible parametrization is $d_i^\mu \in [0, d_{max}]$ with increments of $\phi(1)$ between two successive values in this interval.

At the moment we lack numerical results for testing the validity of this theoretical setup. This is planned to be done in the next future and the results will appear in [50].

Chapter 3

The Matrix Product State approximation for the dynamic cavity method

The dynamic cavity method introduced in section (1.2) is affected by a major problem. In is exact form and if no assumptions are made about the network topology nor on the type of the dynamics, it has an exponential complexity in time t. This allows the numerical implementation only for few time steps, otherwise one is forced to formulate assumptions in order to make the problem solvable. In the following we introduce a method used in quantum mechanics to describe many-particle systems, the Matrix Product States (MPS). We will show how this can be adapted to the cavity formalisms allowing to set up an approximation that scales only polynomial in time. This adapts to describe parallel dynamics and has the potential to allow the treatment of dynamical processes on infinite systems where standard techniques such as Monte Carlo Markov Chain (see section 1.2.3) simulations fail. The results of this work will soon appear in a future manuscript [56].
An important feature of this method is that it allows the estimate of marginal probabilities of various types: either of a single trajectory or of a given configuration at a specific single time step. In this case Monte Carlo becomes extremely inefficient because of the computational cost required to run over a large portion, if not all, of the configuration space. Knowing the probability of a given configuration $\bar{\sigma}^T = (\sigma_i^T, \dots, \sigma_N^T)$ at $t = T$ could be useful in the context of rare events. Consider for instance a dynamical process and a particular event which has a low probability to happen but if it happens this will have major consequences, for instance an earthquake or a bank failure. It should be of great interest to estimate the probability of this to happen. The knowledge of the marginals in thus paramount when considering extreme or rare events.
Another example is the case where one wants to estimate what was the more likely initial condition $\bar{\sigma}^*$ such that a given event $\bar{\sigma}^T$ has been experienced at $t = T$. This is equivalent to ask for:

$$\bar{\sigma}^* = \arg\min_{\bar{\sigma}^0} P(\bar{\sigma}^0|\bar{\sigma}^T) \tag{3.1}$$

where $P(\bar{\sigma}^0|\bar{\sigma}^T)$ is the likelihood of having initial configuration $\bar{\sigma}^0$ conditioned to have the final configuration $\bar{\sigma}^T$. This could be solved using Bayesian maximum likelihood. In this case one assumes that:

$$P(\bar{\sigma}^0|\bar{\sigma}^T) \propto P(\bar{\sigma}^T|\bar{\sigma}^0) \tag{3.2}$$

and therefore one needs to quantify the quantity $P(\bar{\sigma}^T|\bar{\sigma}^0)$ in order to solve the problem. The dynamical cavity method allows for such an estimate but, given the exponential number of possible trajectories starting with a configuration $\bar{\sigma}^0$ and leading to the final $\bar{\sigma}^T$, it is in general

not feasible if considering exact equations as in (1.17).

The remaining of the chapter is divided as follows: in section 3.1 we will introduce the general framework of the MPS as it is used in the context of many-particle physics. The adaptation of this method to the description of the dynamical cavity equations is given in section 3.2. There we will both describe the model and present some preliminary numerical results supporting the method.

3.1 The Matrix Product formalism in many particle physics

In this section we give the theoretical background needed to understand the MPS formalism used to describe the approximation of the dynamic cavity equations given in the next section (3.2). We will set up the main framework used in quantum mechanics and introduced some techniques of linear algebra needed for the actual numerical implementation.

In quantum mechanics the configuration space describing a many-particle system is in general exponential in the number of particles. Therefore it can be computationally unfeasible to find a particular state, for instance the ground state. Nonetheless it is reasonable to think that, if the Hamiltonian describing the system is local, the search for a given state should be focused around a small subset of the entire space. This suggests to find an efficient parametrization of the states so to restrict the search around a sensitive neighborhood inside the complete configuration space. In fact a given state can be written exactly as a product of matrices, the so called *Matrix Product State* representation (MPS) [57]. In order to introduce this parametrization we need two related concepts: the Singular Value Decomposition and the Schmidt decomposition. These two ingredients will eventually enable the MPS decomposition.

Is a standard result of linear algebra that a given $M \times M$ matrix A can be decomposed [58] as:

$$A = USV \tag{3.3}$$

where U, V are unitary matrices, i.e. $U^\dagger U = \mathbb{1}$ and $VV^\dagger = \mathbb{1}$, the symbol $\dagger$ denotes the adjoint of the matrix and $\mathbb{1}$ is the (in this case M-dimensional) identity matrix. Whereas $S = diag(s_1, \dots, s_M)$ and the numbers s_i are called singular values. This can be written also as:

$$A_{ab} = \sum_{k=1}^{M} U_{ak} s_k V_{kb} \tag{3.4}$$

where A_{ab}'s denote the entries of the matrix A. This decomposition is called the *Singular Value Decomposition* (SVD).

One could then use the SVD to rewrite a pure quantum mechanics state, denoted as $|\Psi\rangle$ (using Dirac notation [59]), in form of another decomposition by using the singular values s_i. The state $|\Psi\rangle$ can be thought as a function containing the information about the probability of a given configuration of quantum particles. This allows an analogy with the messages (1.17) in the dynamic cavity method formalism.

In quantum mechanics [59] a state $|\Psi\rangle$ in a space AB is pure if it can be written as:

$$|\Psi\rangle = \sum_{ij} \Psi_{ij} |i\rangle_A |i\rangle_B \tag{3.5}$$

where $\{|i\rangle_A\}$ and $\{|i\rangle_B\}$ are orthonormal basis of the spaces A and B. These could have in principle different dimensions N_A and N_B but here we focus on our case which is $N_A = N_B = M$. The coefficients Ψ_{ij} can be seen as entries of a $M \times M$ matrix Ψ. Decomposing this matrix

using SVD (3.4) leads to:

$$|\Psi\rangle = \sum_{k=1}^{M} s_k \, |a\rangle_A \, |b\rangle_B \tag{3.6}$$

where $|a\rangle = \sum_a U_{ak} \, |i\rangle_A$ and $|b\rangle = \sum_b V^*_{bk} \, |i\rangle_B$. From this we see that due to the orthonormality of U and V also $\{|a\rangle_A\}$ and $\{|a\rangle_B\}$ are orthonormal basis of the spaces A and B. If the sum is restricted to run over the $s_k > 0$ equation (3.6) is called *Schmidt decomposition*. If the number of such positive non-zero singular values is greater than 1, it means that $|\Psi\rangle$ is entangled. Bearing in mind that the overall space AB can be huge, the form (3.6) suggests a convenient approximation of the state $|\Psi\rangle$ into another state $|\tilde{\Psi}\rangle$ defined in a smaller space. In fact, if one orders the singular values in decreasing order of their magnitude then one can neglect the smaller s_i according to some criteria. The dimension M could then be reduced by truncating the original matrix Ψ by keeping only the first $\tilde{M}$ entries of the matrices U and V of the singular value decomposition (3.4), where we assumed that the s_i are ordered as described before. This defines another matrix $\tilde{\Psi}$ so that the approximate state is:

$$|\tilde{\Psi}\rangle = \sum_{k=1}^{\tilde{M} \leq M} s_k \, |\tilde{a}\rangle_A \, |\tilde{b}\rangle_B \tag{3.7}$$

where $|\tilde{a}\rangle$ and $|\tilde{b}\rangle$ are the $\tilde{M}$ dimensional vectors which are made of the first $\tilde{M}$ entries of the original $|a\rangle$ and $|b\rangle$ appearing in (3.6). This procedure is called truncation and the quality of the approximation depends on how it is performed: the bigger the truncation, i.e. the bigger the number of singular values neglected, the worsen the approximation.
It is important to say that the quality of the approximation should be given in terms of the 2-norm $|| \, |\Psi\rangle \, ||_2^2$ of $|\Psi\rangle$, which is the quantity of our interest given the analogy with the messages in the cavity formalism. The 2-norm is defined as:

$$|| \, |\Psi\rangle \, ||_2^2 = \sum_{ij} |\Psi_{ij}|^2 \, \langle i|i\rangle_A \, \langle i|i\rangle_B \tag{3.8}$$

Hence, starting from orthonormal basis $\{|i\rangle_A\}$ and $\{|i\rangle_B\}$, we can write:

$$|| \, |\Psi\rangle \, ||_2^2 = \sum_{ij} |\Psi_{ij}|^2 = ||\Psi||_F^2 \tag{3.9}$$

where $||\Psi||_F$ denotes the Frobenious norm of the matrix Ψ [58]. Equation (3.9) shows that controlling the quality of the truncation, i.e. of the approximation of the state $|\Psi\rangle$ in terms of the 2-norm, is equivalent to control the Frobenious norm of the matrix Ψ. This will be useful when considering matrix products.
Equation (3.8) could also be written in terms of the singular values as:

$$|| \, |\Psi\rangle \, ||_2^2 = \sum_{k=1}^{M} s_k^2 \, \langle a|a\rangle_A \, \langle b|b\rangle_B \tag{3.10}$$

thus the quality of the truncation can be measured by the number of truncated singular values. This also shows that we need orthonormal basis $\{|a\rangle_A\}$ and $\{|a\rangle_B\}$ in order to control the approximation. Supposing this is the case, then we have that:

$$|| \, |\Psi\rangle - |\tilde{\Psi}\rangle \, ||_2^2 = \sqrt{\sum_{k>\tilde{M}} s_k^2} \tag{3.11}$$

Considering a $1D$ chain long L of particles σ_i (with $i = 1, \ldots, L$) we can generalize the state representation (3.5) to the case:

$$|\Psi\rangle = \sum_{\sigma_1, \ldots, \sigma_L} c_{\sigma_1, \ldots, \sigma_L} \, |\sigma_1, \ldots, \sigma_L\rangle \tag{3.12}$$

where there are exponentially many, d^L, coefficients $c_{\sigma_1,\ldots,\sigma_L}$. It is possible [57] to rewrite these coefficients in a more convenient way as a product of matrices. This can be obtained by first building, starting from the left, an initial matrix Ψ of dimension $d \times d^{L-1}$ as:

$$\Psi_{\sigma_1,(\sigma_2,\ldots,\sigma_L)} = c_{\sigma_1,\ldots,\sigma_L} \tag{3.13}$$

and then by performing successive SVD to it combined with some indices rearrangements. Eventually [57] on obtains the matrix product representation:

$$|\Psi\rangle = \sum_{\sigma_1,\ldots,\sigma_L} A^{\sigma_1} \ldots A^{\sigma_L} |\sigma_1, \ldots, \sigma_L\rangle \tag{3.14}$$

where $A^{\sigma_1}, \ldots . A^{\sigma_L}$ are matrices of dimension at most $(1 \times d), (d \times d^2), \ldots, (d^{L/2-1} \times d^{L/2}), (d^{L/2} \times d^{L/2-1}), \ldots, (d^2 \times d), (d \times 1)$ respectively.

This shows that they can blow up exponentially, thus preventing their numerical determination. Similar MPS representations can be obtained [57] by starting with a matrix Ψ as in equation (3.13) but starting ordering indices from the right, i.e. $\Psi_{(\sigma_1,\ldots,\sigma_{L-1}),\sigma_L} = c_{\sigma_1,\ldots,\sigma_L}$, or a combination of the two. Regardless the starting method, in general the matrices obtained in the MPS are subject to a gauge degree of freedom that can be exploited to simplify manipulations.

Nonetheless the main problem remains: the dimension of these matrices can be exponential. Indeed one could reduce this complexity by performing a truncation that keeps only a bounded number $\tilde{M} < M$ of singular values. This will mean to compress the size of the matrices A so to keep only the first $\tilde{M}$ rows and columns, thus obtaining new matrices $\tilde{A}$ which are approximation of the previous ones.

As we said before, the magnitude of the truncation directly affects the quality of the approximation. Hence it is important to always specify what is the rule used to truncate. Two possible truncation criteria can be considered in the MPS routine.

The first fixes a priori the number of singular values kept at each iteration update. This is equivalent to fix the maximum matrix dimension $\tilde{M}$.

The second fixes a threshold L_{max} for the Frobenious norm of the difference in the exact matrix A and its approximated counterpart $\tilde{A}$. Then in this second criterium one cuts the maximum number of singular value so that the threshold norm is not overcome. This can be formalized as follows: one as to find the minimum integer number $\tilde{M} \leq M$ such that:

$$||A - \tilde{A}||_2 = \sqrt{\sum_{i>\tilde{M}} s_i^2} < L_{max} \tag{3.15}$$

where the truncated $\tilde{A}$ is a $\tilde{M} \times \tilde{M}$ matrix with entries $\tilde{A}_{ab} = \sum_{k=1}^{\tilde{M}} U_{ak} s_k V_{kb}$ and the singular values s_i are intended to be ordered in the decreasing order of their magnitude, i.e. $s_1 \geq s_2 \geq \cdots \geq s_M$.

To summarize, in this section we have introduced the MPS representation of a many-particle quantum state. The exact representation involves matrices that in general can have a dimension blowing up exponentially. However, the MPS allows to perform a natural approximation by keeping only the first $\tilde{M}$ singular values and thus compressing the original matrix, with the effect of reducing the complexity from exponential to polynomial. This concept of approximating a matrix through truncations will be the main idea behind the dynamic cavity method approximation which will be the subject of the next section.

3.2 The Matrix Product State representation of parallel dynamics

Here we present an approximation of the dynamic cavity method introduced in section 1.2. The main objective of this approximation is to reduce the complexity from exponential to polynomial

and this is done by exploiting the concept of Matrix Product State (MPS) representation introduced above. The idea is to make an analogy between a pure many-particle state of quantum mechanics and the messages of the cavity method. That an analogy may be foreseen is suggested by the fact that both these quantities contain information about the probability of a given state. In quantum mechanics an entangled state cannot be factorized as a tensor product of single states defined in different spaces. This means that different subsystems cannot be treated as completely independent in an entangled state. Usually this entanglement arises as a result of interactions between degrees of freedom of single-particle states. At the same time under a generic dynamics, a cavity message cannot be factorized in time as a result of the overall dynamics not being Markovian. Motivated by the fact that the complexity of a many-particle quantum state can be reduced using MPS decomposition, we try to exploit the same tool applying it to our problem.

The complexity of the cavity equation (1.17):

$$\mu_{ij}(\bar{\sigma}_i^{t+1}|\bar{\sigma}_j^t) = \frac{1}{Z_{ij}} P_i(\sigma_i^0) \sum_{\{\bar{\sigma}_k^{t-1}\}} \prod_{s=0}^{t} W(\sigma_i^{s+1}|\sigma_i^s, \{\sigma_j^s\}_{j\in\partial i}) \prod_{k\in\partial i\setminus j} \mu_{ki}(\bar{\sigma}_k^t; \bar{\sigma}_i^{t-1}) \tag{3.16}$$

is exponential in time because the path trajectory $\bar{\sigma}_i^t$ can take d^t possible values, where d is the number of values that the variable σ_i^t can take. Therefore it can be numerically implemented only up to few time steps. However the complexity can be substantially reduced if one assumes that it can be factorized in time:

$$\mu_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1}) = \prod_{s=0}^{t} \mu_{ij}^s(\sigma_i^{s+1}|\sigma_j^s) \tag{3.17}$$

This assumption, the *time-factorization* approximation, should be intended valid in the stationary state $t \to \infty$ where is reasonable to expect that messages do not depend on time but it should not be applied to describe transient times where the dynamics is not Markovian. On the computational side this assumption allows a feasible numerical implementation because the complexity is reduced from exponential to polynomial in time.

Our idea is to write the general message $\mu_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1})$ in a factorized form different from (3.17) but that is valid $\forall t$, not only in the stationary state and regardless the properties of either the network topology (directed or undirected) or the form of the dynamics (reversible or irreversible). This factorization is made in form of products of matrices, in analogy with what is done in quantum mechanics to study entanglement between particles. Indeed in section (3.1) we saw how to approximate a generic pure many-particle quantum state by representing it as a product of matrices (3.14) and then compressing them by keeping only the biggest singular values. Here we adapt the same idea to the representation of the messages.

We propose the following Matrix Product States (MPS) factorization ansatz for the dynamic cavity equations:

$$\mu_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1}) = A_{ij}^{(t+1)}(\sigma_i^t) A_{ij}^{(t)}(\sigma_i^{t-1}) \left[\prod_{s=1}^{t} A_{ij}^{(s)}(\sigma_i^{s-1}|\sigma_j^s)\right] A_{ij}^{(0)}(\sigma_j^0) \tag{3.18}$$

where the quantities A's have the following shape: $A_{ij}^{(t+1)}(\sigma_i^t)$ is a $1 \times M$ matrix; $A_{ij}^{(t)}(\sigma_i^{t-1})$ and all the $A_{ij}^{(s)}(\sigma_i^{s-1}|\sigma_j^s)$ are $M \times M$ matrices; $A_{ij}^{(0)}(\sigma_j^0)$ is a $M \times 1$ matrix. Therefore $\mu_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1})$ results in a scalar quantity as it should be.

The matrix dimension M is in general determined by the number of neighbors of i and to the quality of the approximation, thus it may differ for different messages and time steps. At the initial time step one could simply fix $M = 1$ and then, iterating in time, this dimension will in general blow up. The idea is then to control this increase in matrix dimension by performing a truncation based on the singular values of the matrices A's. We will explain this in more details below but first we analyze the equation (3.18).

The equation is saying that the messages can indeed be factorized in time but not as product of messages itself, instead they are factorized as a structured combination of entries of matrices. Each of these matrices being a function of only two successive time steps. This complex combination is supposed to capture the whole information about the correlations in time of the path trajectories. Therefore it is not assuming Markovianity and thus it should be valid $\forall t$, not only for the stationary state but also for the transient times. The fact that the matrices $A_{ij}^{(s)}$ depend on only two time steps and not on the entire trajectory, could allow for a reduction in complexity from exponential to polynomial in time. The product of such matrices *entangles* the dynamics at different time steps together, thus processing the entire history without making any assumption about the dynamics. The only requirement given by the matrix product representation (3.18) is that the update should be made in parallel and with synchronous steps. This is due to the number of matrices produced at each time step that have to be consistent among all the edges in order to then insert them into the product over neighbors $k \in \partial i \setminus j$ in the left-hand side of (3.16).
Assuming the validity of the ansatz (3.18) on can insert it into (3.16) and obtain the update rule consistently:

$$\mu_{ij}(\bar{\sigma}_i^{t+1}|\bar{\sigma}_j^t) = B_{ij}^{(t+2)}(\sigma_i^{t+1})B_{ij}^{(t+1)}(\sigma_i^t)\left[\prod_{s=1}^{t} B_{ij}^{(s)}(\sigma_i^{s-1}|\sigma_j^s)\right] B_{ij}^{(0)}(\sigma_j^0) \tag{3.19}$$

$$= \frac{1}{Z_{ij}} P_i(\sigma_i^0) \sum_{\{\bar{\sigma}_k^{t-1}\}} \prod_{s=0}^{t} W(\sigma_i^{s+1}|\sigma_i^s, \{\sigma_j^s\}_{j\in\partial i}) \times$$

$$\times \prod_{k\in\partial i\setminus j} \left\{ A_{ki}^{(t+1)}(\sigma_k^t)A_{ki}^{(t)}(\sigma_k^{t-1}) \left[\prod_{s=1}^{t} A_{ki}^{(s)}(\sigma_k^{s-1}|\sigma_i^s)\right] A_{ki}^{(0)}(\sigma_i^0) \right\} \tag{3.20}$$

Notice that in $A_{ij}(\sigma_i^{t-1}|\sigma_j^t)$ the time indices are swapped as compared to their order inside the transition probability. This is a necessary choice in order to get consistent representation for the left and right-hand sides of equation (3.19). In particular this allows to perform efficiently the product:

$$W(\sigma_i^s|\sigma_i^{s-1}, \{\sigma_j^{s-1}\}_{j\in\partial i}) \prod_{k\in\partial i\setminus j} A_{ki}^{(s)}(\sigma_k^{s-1}|\sigma_i^s) \tag{3.21}$$

because the variables σ_i^s and σ_k^{s-1} appear with the same time index in both the terms. One could then perform a tensor product exploiting this consistency in the order of the indices.
The ansatz as it is, it's exact to describe the cavity equations (3.16) but the size of the matrices A's increases with time, therefore leading to the exponential complexity of the original problem if this increment is not controlled. The idea is thus to control the matrices' dimension by limiting it, at each time step, to an arbitrary maximum value that one has to fix in order to tune the quality of the approximation, thus reducing the computational complexity. A sensitive dimension control is achieved by rewriting the matrices A in the form of a Singular Values Decomposition (SVD) [58], a linear algebra technique introduced above (3.4). In our case this means to decompose the matrix $A(\sigma|\sigma')$ (where we omitted the superscript (s) and the subscript ij for simplicity) as:

$$[A(\sigma|\sigma')]_{a,b} = \sum_k U_{(\sigma,a),k} s_k V_{k,(\sigma',b)} \tag{3.22}$$

where we are using the notation $U_{m,n}$ to denote the entry of a matrix so that it will be more convenient to represent tensors. In fact in the numerical implementation on treats the matrix $A(\sigma|\sigma')$ as a tensor made of 4 indices: one for each variable σ, σ' and one for each entry a, b of the matrix. This tensor will then be split among the unitary matrices U and V. To achieve this one writes a new variable (σ, a) that will denote the row of the entry $U_{(\sigma,a),k}$. This will be

used to move variables σ among different positions in the matrix product ansatz. In particular this will let us write $A_{ij}^{(s)}(\sigma_i^{t-1}|\sigma_j^t)$ with the first variable (σ_i) appearing at a previous time step compared to the second one (σ_j), which is the opposite of what happens for the transition probabilities $W(\sigma_i^t|\sigma_i^{t-1},\{\sigma_j^{t-1}\}_{j\in\partial i})$.

3.2.1 The MPS decomposition of messages

Here we present the main procedure needed to obtain the matrices B in (3.19) starting from the neighbors' A in (3.20), a series of successive operations on tensors developed by T. Barthel, and allowing to eventually represent the messages as products over matrices.
The idea is to start by building an intermediate set of matrices $C_{ij}^{(s)}(\sigma_i^s|\sigma_j^{s-1})$ using (3.21) and the marginalization over the σ_k^s. For the sake of simplicity from now on we drop the dependence on the variable σ_i^{s-1} inside $W(\sigma_i^s|\sigma_i^{s-1},\{\sigma_j^{s-1}\}_{j\in\partial i})$ so to use instead transition probabilities of the form $W(\sigma_i^s|\{\sigma_j^{s-1}\}_{j\in\partial i})$. The results we will obtain could be generalized to the former case by adding more manipulations of tensors.
The definition of this intermediate matrix $C^{(s)}$ depends upon the position of the time step s inside the interval $[0,t+1]$.
For a *bulk* time step, i.e. $0<s<t$, we define:

$$C_{ij}^{(s)}(\sigma_i^s|\sigma_j^{s-1}) = \sum_{\{\sigma_k^{s-1}\}_{k\in\partial i\setminus j}} W(\sigma_i^s|\{\sigma_j^{s-1}\}_{j\in\partial i}) \prod_{k\in\partial i\setminus j} A_{ki}^{(s)}(\sigma_k^{s-1}|\sigma_i^s) \tag{3.23}$$

whereas at the boundaries ($s=0,t,t+1$):

$$C_{ij}^{(0)}(\sigma_i^0) = P_i(\sigma_i^0) \prod_{k\in\partial i\setminus j} A_{ki}^{(0)}(\sigma_i^0) \tag{3.24}$$

$$C_{ij}^{(t)}(\sigma_i^t|\sigma_j^{t-1}) = \sum_{\{\sigma_k^{t-1}\}_{k\in\partial i\setminus j}} W(\sigma_i^t|\{\sigma_j^{t-1}\}_{j\in\partial i}) \prod_{k\in\partial i\setminus j} A_{ki}^{(t)}(\sigma_k^{t-1}) \tag{3.25}$$

$$C_{ij}^{(t+1)}(\sigma_i^{t+1}|\sigma_j^t) = \sum_{\{\sigma_k^t\}_{k\in\partial i\setminus j}} W(\sigma_i^{t+1}|\{\sigma_j^t\}_{j\in\partial i}) \prod_{k\in\partial i\setminus j} A_{ki}^{(t+1)}(\sigma_k^t) \tag{3.26}$$

The entry of a bulk matrix C can be written as:

$$\left[C_{ij}^{(s)}(\sigma_i^s|\sigma_j^{s-1})\right]_{\bar a,\bar b} = \sum_{\{\sigma_k^{s-1}\}_{k\in\partial i\setminus j}} W(\sigma_i^s|\{\sigma_j^{s-1}\}_{j\in\partial i}) \prod_{k\in\partial i\setminus j} \left[A_{ki}^{(s)}(\sigma_k^{s-1}|\sigma_i^s)\right]_{a_k,b_k} \tag{3.27}$$

so that the indices $\bar a=(a_1,\dots,a_{k_i-1})$ and $\bar b=(b_1,\dots,b_{k_i-1})$ grouped together lead to the final matrix C that has dimension M^{k_i-1}, where k_i is the degree of node i and M the dimension of the matrices A. A graphical representation of a bulk matrix $C_{ij}^{(s)}(\sigma_i^s|\sigma_j^{s-1})$ with $k=z+1$ neighbors is given in figure 3.1.

Given the two sets of M indices $\bar a$ and $\bar b$, and considering the dimension d of the space of a spin variable σ_i^s, the computational cost of computing (3.23) scales as $O(d^3M^{2(k_i-1)})$. Similar considerations can be made for the boundary terms (3.24),(3.25) and (3.26).
Putting all together one finally obtains:

$$\mu_{ij}(\bar\sigma_i^{t+1}|\bar\sigma_j^t) = \left[\prod_{s=1}^{t+1} C_{ij}^{(s)}(\sigma_i^s|\sigma_j^{s-1})\right] C_{ij}^{(0)}(\sigma_i^0) \tag{3.28}$$

Notice that form is not yet the one we want to finally obtain as in the ansatz (3.19), the dependence on time is inverted among σ_i^s and σ_j^{s-1} as compared to what happens for the A's. Supposing to iterate this routine for increasing time t, the dimension of the matrices C will blow up exponentially in time as expected for the exact cavity equations. Therefore now we proceed to decompose the various C's using SVD (3.4) and then compress the resulting matrices so to

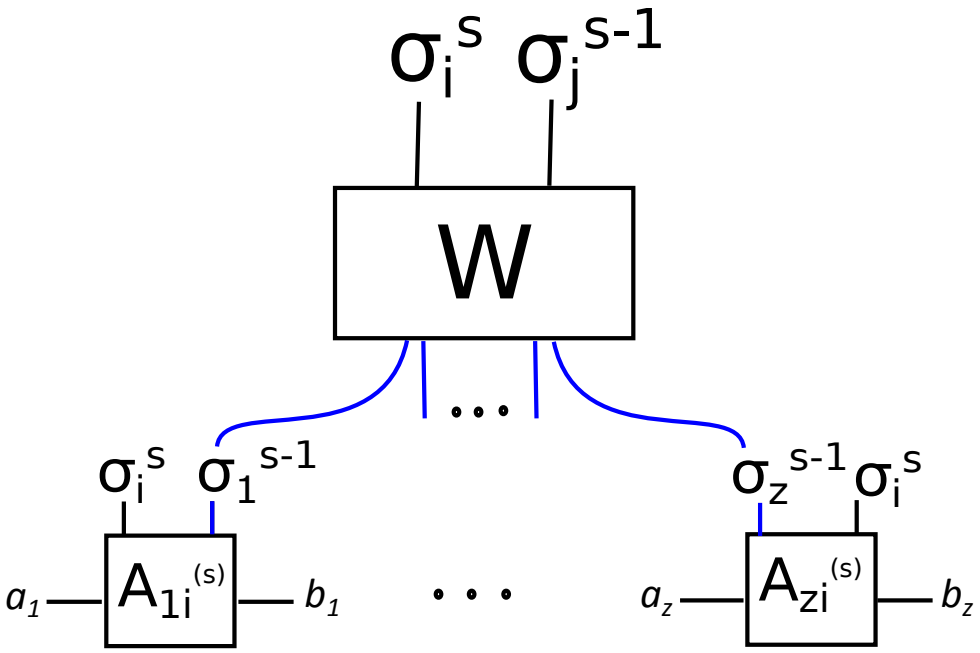


Figure 3.1: Example of a bulk matrix $C_{ij}^{(s)}(\sigma_i^s|\sigma_j^{s-1})$ in the MPS representation (3.23). The blu lines connecting the transition probability $W = W(\sigma_i^s|\sigma_i^{s-1}, \{\sigma_j^{s-1}\}_{j\in\partial i})$ with the matrices A_{ki} represent the marginalization over the variable σ_k^{s-1}. Notice the same dependence on σ_i^s of the A's and W.

control their dimension. One has to be careful though when truncating, because the property of orthonormality among vectors has to be preserved in order to keep the Frobenius norm of the truncated matrix close enough to the original one as in (3.15). In our case this means performing two preparatory steps before the actual truncation. These are two sweeps where we perform SVD for the matrices C in (3.28) first from left to right and then the other way around. In each of these sweeps we do not truncate, but transform the matrices so to enforce orthonormality, first for the right basis and then for the left one. Finally one can proceed with the actual truncation by performing a third sweep from left to right. We leave the details of these calculations in the appendix 2.

Once a given truncation threshold $\tilde{M}$ is chosen, one proceeds by truncating the sum in (3.22) keeping only the first $\tilde{M}$ singular values. At the same time the matrices U, V will be truncated as well, keeping only the first $\tilde{M}$ rows and columns. The more we truncate the more the correlations are neglected. Nonetheless the bigger the truncation the faster the numerical implementation, because the matrix product (3.18) is performed over matrices of reduced dimension. Therefore setting the truncation threshold involves a trade-off between a lower computational complexity and a worse quality of the solution or viceversa. In a sense, the magnitude of the truncation controls the approximation of the time correlations of the trajectories $\bar{\sigma}_i^t$.

The numerical implementation then works as follows. At the first time step $s = 0$ only two matrices have to be built for each message: $A_{ij}^{(0)}(\sigma_j^0)$ and $A_{ij}^{(1)}(\sigma_i^0)$. Their dimension can be arbitrarily initialized, the simplest choice is to fix $M = 1$. The only information given at the first time step is $P_i(\sigma_i^0)$, therefore using the MPS ansatz (3.18) we have:

$$A_{ij}^{(0)}(\sigma_j^0) = 1 \tag{3.29}$$

$$A_{ij}^{(1)}(\sigma_i^0) = P_i(\sigma_i^0) \tag{3.30}$$

This is equivalent to initialize messages:

$$\mu_{ij}(\sigma_i^0) = A_{ij}^{(1)}(\sigma_i^0)A_{ij}^{(0)}(\sigma_j^0) = P_i(\sigma_i^0) \tag{3.31}$$

where one could add a dummy variable σ_j^{-1} in the numerical implementation to keep the structure consistent with the messages obtained in the next iterations.
One proceeds with the next iteration by first building the matrices C using (3.24) and (3.25) such that:

$$C_{ij}^{(0)}(\sigma_i^0) = P_i(\sigma_i^0) \prod_{k\in\partial i\setminus j} A_{ki}^{(0)}(\sigma_i^0) = P_i(\sigma_i^0) \tag{3.32}$$

$$C_{ij}^{(1)}(\sigma_i^1|\sigma_j^0) = \sum_{\{\sigma_k^0\}_{k\in\partial i\setminus j}} W(\sigma_i^1|\{\sigma_j^0\}_{j\in\partial i}) \prod_{k\in\partial i\setminus j} A_{ki}^{(1)}(\sigma_k^0) \tag{3.33}$$

$$= \sum_{\{\sigma_k^0\}_{k\in\partial i\setminus j}} W(\sigma_i^1|\{\sigma_j^0\}_{j\in\partial i}) \prod_{k\in\partial i\setminus j} P_i(\sigma_i^0) \tag{3.34}$$

Notice that at this step the matrices C have still dimension $M = 1$ but indeed they should be rewritten as tensor of the form:

$$C(\sigma|\sigma')_{a,b} = \mathcal{C}_{(\sigma,a),(\sigma',b)} \tag{3.35}$$

which makes the next SVD more convenient to perform as written in (3.22). As a result the new tensor $\mathcal{C}$ has dimension $d \times d$ in this first time step where $a = b = 1$.
To complete this iteration one performs the MPS routine made of two preparatory sweeps to ensure orthogonality and a third one to obtain the ansatz (3.19). At the end of this first step one has a set of 3 matrices B such that:

$$\mu_{ij}(\bar\sigma_i^1|\bar\sigma_j^0) = B_{ij}^{(2)}(\sigma_i^1)B_{ij}^{(1)}(\sigma_i^0)B_{ij}^{(0)}(\sigma_j^0) \tag{3.36}$$

This is nothing but an alternative parametrization of equation (3.34) multiplied by (3.32) consistent with the MPS ansatz (3.18). The same procedure is repeated in the next steps $s > 0$. At the end of each iteration a new set of $t+2$ matrices B is created and in general their dimension increases with time in a way that is controlled by the chosen truncation criterium.

3.2.2 Evaluation of marginals

Eventually the quantities of interest, for instance one-time marginals $P(\sigma_i^t, \sigma_j^t)$ as in (1.21) or more generally $P(\sigma_i^{t_1}, \sigma_j^{t_2})$, can be calculated from the MPS representation (3.18). This evaluation can be performed at the end of each iteration step and the idea is to use the expression (1.16) of the two-variable marginal:

$$P_{ij}(\bar\sigma_i^t, \bar\sigma_j^t) = \mu_{ij}(\bar\sigma_i^t|\bar\sigma_j^{t-1})\mu_{ji}(\bar\sigma_j^t|\bar\sigma_i^{t-1}) \tag{3.37}$$

and then marginalize out over all but the variable of interest.
In our MPS decomposition, the marginalization can be naturally formalized as a series of successive tensor contractions. Namely the sum of products over the scalar entries having same indices between two different tensors A, B, i.e. using Einstein notation [59] this is the generalization of the trace of matrices $A_{ab}B_{bc} = \sum_b A_{ab}B_{bc} = C_{ac}$ in the case of tensors.
In fact (3.37) is a product of matrices $A(\sigma|\sigma')$ but we have already seen that it is often convenient to interpret them as tensors (3.35). Here we use a similar expression for a M-dimensional matrix $A(\sigma|\sigma')$:

$$[A(\sigma|\sigma')]_{a,b} = A_{\sigma,\sigma',a,b} \tag{3.38}$$

hence a set of d^2 matrices of dimension $M \times M$ is transformed in a $d \times d \times M \times M$ tensor. As a result the marginalization over a variable σ_i^s in the calculation of the marginals (3.37) is reduced to a contraction over the common indices of two tensors $[A(\sigma|\sigma')]_{a,c} = A_{\sigma,\sigma',a,c}$ and

$[B(\sigma|\sigma'')]_{c,b} = B_{\sigma,\sigma'',c,b}$:

$$\sum_{\{\sigma\}} A(\sigma|\sigma')B(\sigma|\sigma'') = C(\sigma',\sigma'') \tag{3.39}$$

$$[C(\sigma',\sigma'')]_{a,b} = A_{\sigma,\sigma',a,c}B_{\sigma,\sigma'',c,b} \tag{3.40}$$

$$= \sum_{\sigma,c} A_{\sigma,\sigma',a,c}B_{\sigma,\sigma'',c,b} \tag{3.41}$$

This allows to perform to perform the marginalization over en entire trajectory $\bar{\sigma}_i^{t-1}$ as a series of successive single-time marginalization, thus reducing the complexity of this calculation from exponential to polynomial in time t.

Here we describe the entire routine to calculate $P(\sigma_i^t, \sigma_j^t)$ based on the tensor contractions just described, but this procedure can be adapted to calculate other single-time marginals.

We start by writing (3.37) as a product of two sets of tensors $A^{(s)}$ and $B^{(s)}$, each one having 4 indices: $\left[A^{(s)}(\sigma|\sigma')\right]_{a,b} = A^{(s)}_{\sigma,\sigma',a,b}$ and $\left[B^{(s)}(\sigma|\sigma'')\right]_{a',b'} = B^{(s)}_{\sigma,\sigma'',a',b'}$ so that:

$$\mu_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1}) = A_{ij}^{(t+1)}(\sigma_i^t)A_{ij}^{(t)}(\sigma_i^{t-1}) \left[\prod_{s=1}^{t} A_{ij}^{(s)}(\sigma_i^{s-1}|\sigma_j^s)\right] A_{ij}^{(0)}(\sigma_j^0) \tag{3.42}$$

$$\mu_{ji}(\bar{\sigma}_j^t|\bar{\sigma}_i^{t-1}) = B_{ji}^{(t+1)}(\sigma_j^t)B_{ji}^{(t)}(\sigma_j^{t-1}) \left[\prod_{s=1}^{t} B_{ji}^{(s)}(\sigma_j^{s-1}|\sigma_i^s)\right] B_{ji}^{(0)}(\sigma_i^0) \tag{3.43}$$

In the rest, to simplify notation, we drop the indices ij and ji from the matrices $A^{(s)}$ and $B^{(s)}$ respectively.

The dimension of $A^{(s)}(\sigma|\sigma')$ is M_1 and the one of $B^{(s)}(\sigma|\sigma'')$ is M_2. These two numbers can be different depending on the contractions made to calculate the corresponding messages $\mu_{ij}(\bar{\sigma}_i^t|\bar{\sigma}_j^{t-1})$ and $\mu_{ji}(\bar{\sigma}_j^t|\bar{\sigma}_i^{t-1})$ respectively. We can now rearrange the above products so to pair-up matrices (or tensors using the definition (3.38)) $A^{(s)}(\sigma|\sigma')$ and $B^{(s)}(\sigma|\sigma'')$ sharing the same dependence on σ. Taking into account that their dimension can be different but the trace must be performed over indices having same dimensionality, we should start from the right-most matrices, i.e. the ones corresponding to time $s = 0$. In this case $A^{(0)}(\sigma)$ and $B^{(0)}(\sigma)$ are M_1 and M_2 dimensional vectors respectively, thus we can perform the trace by considering the product of one over the transpose of the other. Here we consider the transpose $\left[A^{(s)}(\sigma|\sigma')\right]^T$ of $A^{(s)}(\sigma|\sigma')$.

Starting at $s = 0$ we have:

$$K^{(0)}(\sigma_i^0, \sigma_j^0) = B^{(0)}(\sigma_i^0) \left[A^{(0)}(\sigma_j^0)\right]^T \tag{3.44}$$

where as usual $K^{(0)}(\sigma_i^0, \sigma_j^0)$ can be seen as a tensor with 4 indices $K_{\sigma_i,\sigma_j,a,b}$. In this case a and b have dimension M_2 and M_1 resulting from the contraction (3.44). Notice that in this first step no marginalization has yet been performed.

We proceed at the next time steps $s = 1, \ldots, t-1$ by contracting further:

$$L^{(s+1)}(\sigma_i^{s+1}, \sigma_i^s) = \sum_{\{\sigma_j^s\}} B^{(s+1)}(\sigma_i^{s+1}|\sigma_j^s)\, K^{(s)}(\sigma_i^s, \sigma_j^s) \tag{3.45}$$

$$K^{(s+1)}(\sigma_i^{s+1}, \sigma_j^{s+1}) = \sum_{\{\sigma_i^s\}} L^{(s+1)}(\sigma_i^{s+1}, \sigma_i^s) \left[A^{(s+1)}(\sigma_j^{s+1}|\sigma_i^s)\right]^T \tag{3.46}$$

where the quantities $L^{(s)}$ and $K^{(s)}$ are new tensors resulting from contractions following the rule (3.39), they have dimension $d \times d \times M_2 \times M_1$. Note that this implies the implementation of successive single-time marginalization as contraction over tensor indices.

The final time step is special as the first one because we have to use the boundary matrices

$A^{(t+1)}$ and $B^{(t+1)}$ which are indeed $1 \times M_1$ and $1 \times M_2$ dimensional vectors:

$$K^{(t+1)}(\sigma_i^t, \sigma_j^t) = B^{(t+1)}(\sigma_j^t)\, K^{(t)}(\sigma_i^t, \sigma_j^t)\, \left[A^{(t+1)}(\sigma^t)\right]^T \tag{3.47}$$

Note that this last tensor has dimension $d \times d$. Eventually we use this resulting tensor to calculate the two-variable joint probability as:

$$P(\sigma_i^t, \sigma_j^t) = \frac{K^{(t+1)}(\sigma_i^t, \sigma_j^t)}{\sum_{\{\sigma_i^t, \sigma_j^t\}} K^{(t+1)}(\sigma_i^t, \sigma_j^t)} \tag{3.48}$$

A diagrammatic representation of this series of tensor contractions is given in figure 3.2.

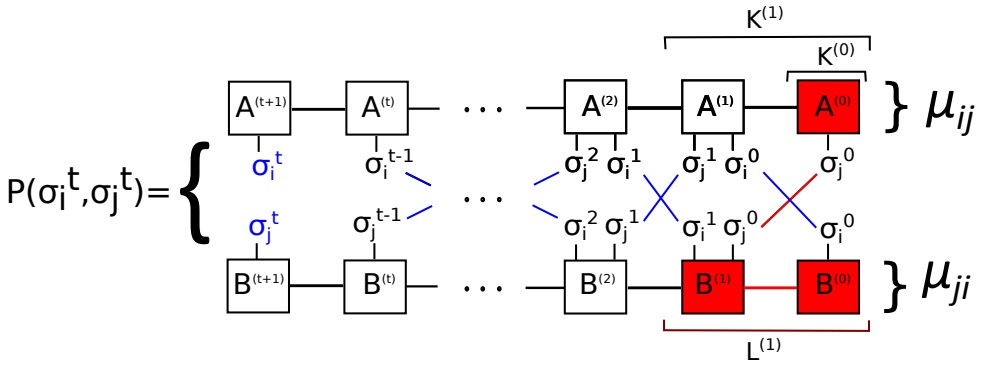


Figure 3.2: Evaluation of two-variable marginal $P(\sigma_i^t, \sigma_j^t)$ in the MPS representation. The red block represent the $L^{(1)}$ tensor as defined in (3.45), horizontal lines are traces of matrices and diagonal ones represent marginalization over the corresponding single-time variable. At the end the remaining variables are σ_i^t and σ_j^t, in fact they are not contracted.

3.2.3 Numerical results

In this section we give the results of numerical simulations performed using Glauber dynamics (1.27) for disordered systems and the majority rule in both the deterministic (1.41) and stochastic (1.42) versions. Note that MPS representation allows to represent the synchronous and parallel update rule, thus we simulate this type of update.

We use random regular graphs as graph topology because we want to focus on evaluating the performance of the algorithm rather then analyze how the dynamics depends on the graph topology. Therefore we use an homogenous graph topology and reserve to use other types of random graphs for future applications.

The simulations are done using the MPS ansatz (3.18) starting with only two matrices A for each message μ_{ij} initialized as in (3.29) and (3.30). The system is the evolved using the definition of the transition matrix given by the model considered and at the end of each iteration step macroscopic observables as those introduced in section 1.2.1 are evaluated using the routine given in section 3.2.2.

We tested our algorithm running it on single instances of graphs and the population dynamic version (1.2.4) as well. We used both the truncation criteria explained above: for the one with a fixed number of singular values $\tilde{M}$ we used $\tilde{M} = 20$; for the one that fixes the norm threshold L_{max} (3.15) we used values of L_{max} from 10^{-3} to 10^{-8}.

For performance comparison we run Monte Carlo Markov chain simulations as described in section 1.2.3. For this we used the same graph instances as for the single instance runs of the MPS cavity equations. We also run the exact dynamic cavity equations (1.17) up to time $t = 6$ (the exponential complexity of the exact case does not allow to solve it for bigger time) but we found identical results for such small times, thus we do not report them.

In figure 3.3 we give the results of the Galuber time evolution of various observables for different truncation criteria and for the Monte Carlo simulations. Here we consider single instances over a 3-regular graph of finite size $V = 1000$ and disorder $\{J_{ij}\}$ distributed as a zero-average Gaussian and variance 1. In addition, we considered $\beta = 1$ and we fix a small initial bias of $\theta_0 = 0.05$. We can notice that both the magnetization and the energy are well reproduced by our MPS approximation, the data in fact are very close to the Monte Carlo ones. When considering smaller quantities, such as $\Delta m(t)$ (1.37) or q_{EA} (1.38) we can notice that the MPS does not perfectly superimpose, and indeed for the lowest value of the approximation, i.e. for higher truncation $L_{max} = 10^{-3}$ the MPS falls apart considerably for increasing times. However, improving the quality of the approximation, i.e. considering smaller L_{max}, the MPS results approach again the Monte Carlo ones. This suggests that a parameter tuning in the sense of the truncation threshold should be performed depending on the model considered. Indeed we found that for lower values of β, i.e. in higher temperature regimes, the worst approximation of $L_{max} = 10^{-3}$ gives good results. Therefore suggesting that the more we lower the temperature the more singular values we must keep in order to maintain a good quality of approximation.

We performed simulations of the Majority rule with stochastic noise and this time our focus was on probing the behavior of the infinite system. Therefore we performed population dynamics simulations of the cavity MPS approximation and compared them with Monte Carlo simulations performed with growing V from 10^2 to 10^4. In figure 3.4 we plot the time-evolution of the total magnetization for different noise values and with a small initial bias of $\theta_0 = 0.01$. Note that smaller values of θ_0 cannot be considered for small system sizes, as for $V = 100$. Here we used only one truncation criterium, namely we fixed $L_{max} = 10^{-8}$ because from parameter tuning we found this as a good trade-off between running time and quality of the approximation. From the picture it is clear that our approximation, with a properly chosen level of truncation, perfectly describes the infinite system size. Results of Monte Carlo simulations tend to approach the MPS ones for largest values of V, however these results are affected by an error that increases with system size. Therefore if one wants to implement Monte Carlo for bigger values of V a larger number of instances should be considered, thus making the implementation considerably slower.

Provided a proper truncation criterium is properly chosen, all these preliminary numerical findings suggest that the MPS approximation of the dynamic cavity method could well describe the time-evolution under a given dynamics rule. Not only one could apply this approximation to analyze single instances, but also it well adapts to address the infinite system size using the population dynamics version.

Here we focused on two types of dynamics but it would be interesting to explore also other applications. Moreover, an interesting question would be to apply this approximation to study inference problems on networks. This would in particular allow to tackle reversible dynamics using the cavity method or to consider the entire time-evolution and not only the stationary state. An opportunity that opens new perspective for future works.

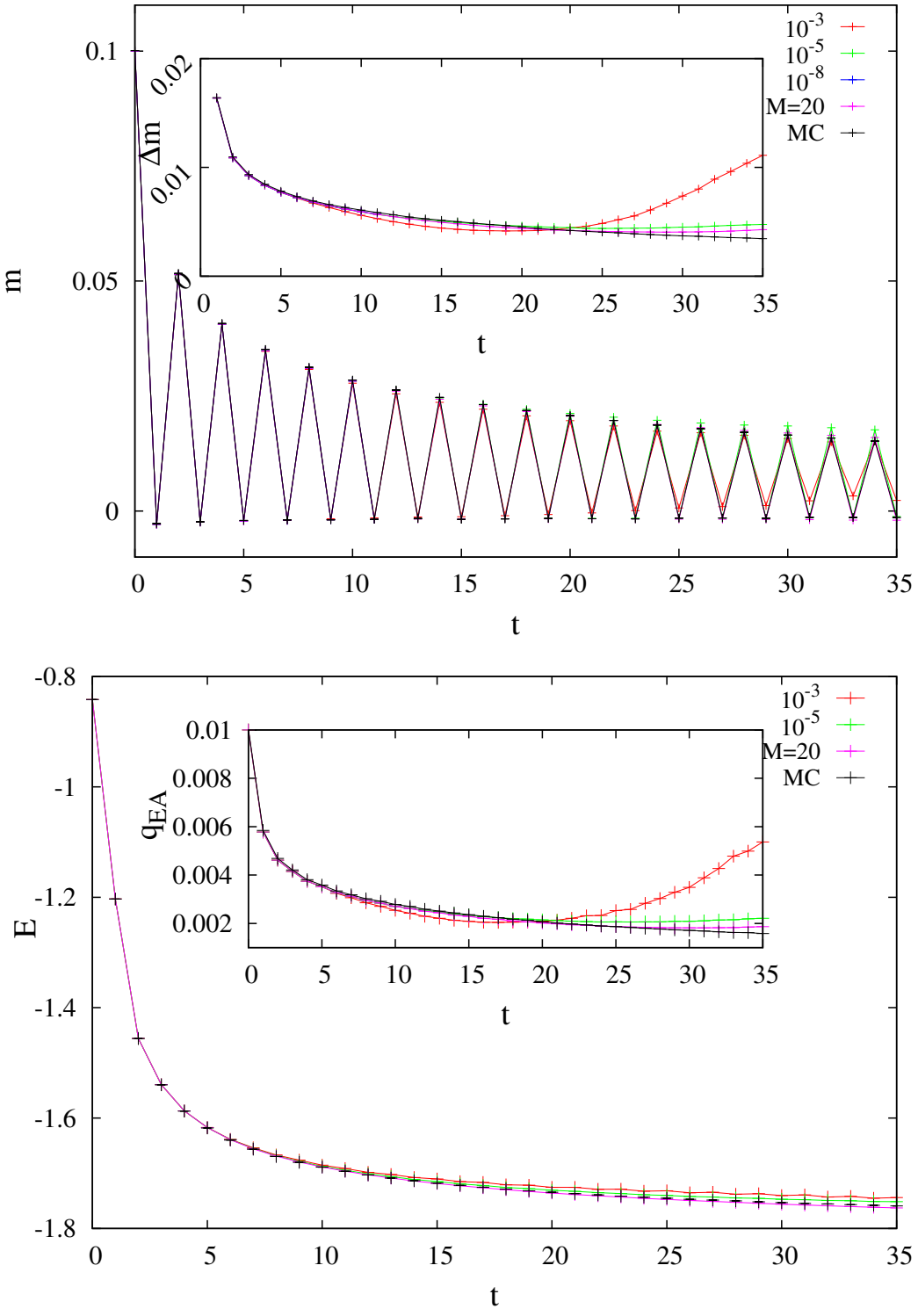


Figure 3.3: Observables evolution in Glauber dynamics. Example of the time evolution predicted by MPS and Monte Carlo (MC) (over 50000 instances) for a 3-regular graph of $V = 1000$. Different curves represent the maximum norm threshold L_{max} applied in the truncation. The curve $M = 20$ is for the truncation criterium with fixed matrix dimension of 20 singular values. We can notice the curve for $L_{max} = 10^{-3}$, after some time steps it starts to diverge. However, by increasing the threshold the quality of the MPS approximation improves. The parameters are: $\beta = 1.0$, $\theta_0 = 0.05$. Inset: $\Delta m(t)$ as defined in (1.37), this helps to notice the performance for different truncation criteria with respect to the Monte Carlo (MC) simulations.

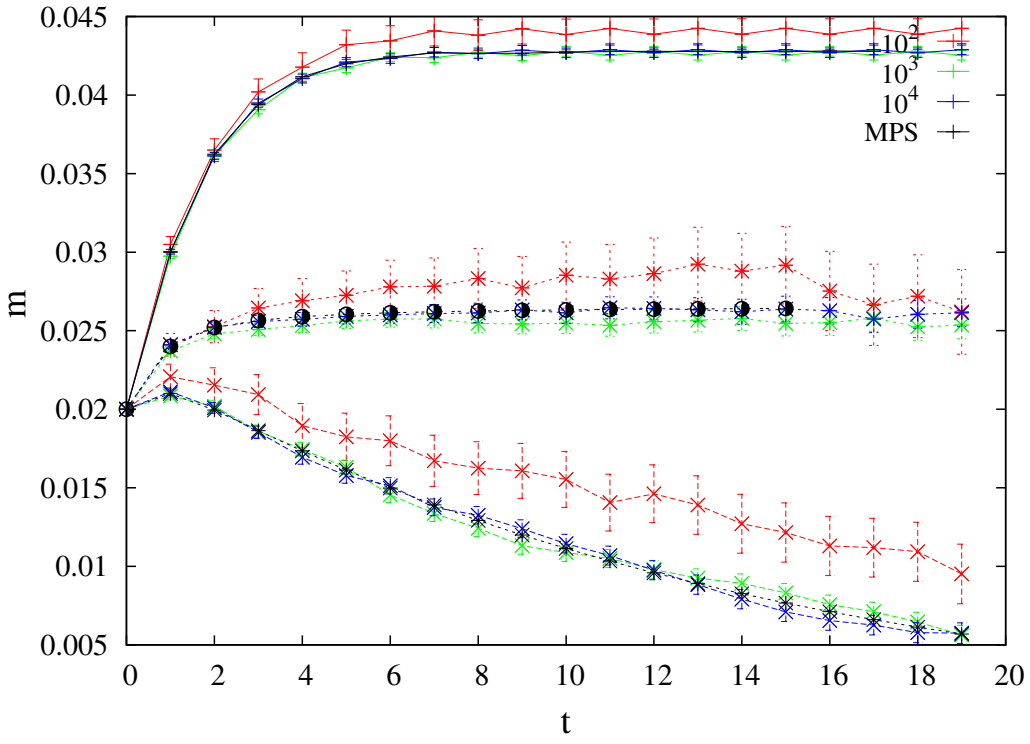


Figure 3.4: Magnetization evolution in Majority rule with noise. Example of the time evolution predicted by the infinite system MPS (population dynamics) and Monte Carlo (MC) (over 10000 instances) for a 3-regular graph of $V = 100, 10^3, 10^4$. For the MPS we fixed a truncation threshold $L_{max} = 10^{-8}$. Three values of the noise Q where considered: from top to bottom $Q = 0.00, 0.10, 0.15$. We can notice that the finite size Monte Carlo tends towards the MPS results for increasing system sizes. The initial bias is $\theta_0 = 0.01$.

Chapter 4

Random walks on networks

In this chapter we study several problems involving random walks on networks. A random walk is defined as a set of successive random steps made by a walker on a given topology, in this work we focus on the case where the topology is a random networks. The random steps are made from one node i to one of its neighboring sites $j \in \partial i$ following a transition rule specified by the model. This can be biased or not, in this introduction we will consider only the case of unbiased walks in order to fix the ideas. In fact the generalization to the biased case will be done by starting from the unbiased results; this approach will be followed when studying the rare event statistics in section 4.3.1.
One of the most important features of a random walk is its cheap implementation. In fact a walker makes a step with a certain probability that depends only on local information. This means that it is not needed to know the situation on the entire network and this translates in network scalability. At the same time if a random walk is used as a strategy for data transmission in communication networks, this feature could allow to achieve fast result and efficiently in terms of energy requirements, given that it is not needed to store big routing tables. This make them suited to be used in networks where the cost of information is high.
There are in general many quantities of interest when studying random walks on graphs [60] but here we focus on the ones studied in [61] and [62].
From a random walker prospective, the two main differences between moving on a regular lattice and on a random network are the graph heterogeneity and the absence of a metric. While a regular lattice has an homoegenuos topology with a fixed number of neighbors for each node, a generic random graph is instead characterized by a degree probability distribution $P(k)$. This gives the probability for a given node to have k neighbors and, a part from the regular graph case where k is fixed for all nodes, in general this number varies with $i \in \mathcal{V}$. This heterogeneity requires a specific treatment when studying random walks and generalizations from the regular lattice case cannot be performed.
The main quantity needed to characterize networks is the adjacency matrix A, a $N \times N$ (where $N = |\mathcal{V}|$) matrix that contains all the information about the graph topology. Therefore is not surprising that many of the techniques developed to study random walks on networks are based on spectral properties of this or other related matrices.
Finally, a random network usually lacks of a universally agreed concept of distance, besides the shortest path length. Therefore the machinery used to study random walks on regular lattices, which relies on being able to characterize it through its spatial coordinates, is not applicable to networks.
In the rest of the chapter we will take these facts into account and exploit different techniques used in linear algebra, probability and large deviation theory to study several statistics associated to random walks on networks.
The rest of the chapter is organized as follow: the general model of random walks on networks

is given in 4.1. In section 4.2 we study the average number of distinct sites visited by a random walker. We conclude by studying rare events of random walks on networks by using the large deviations formalism in section 4.3.1.

4.1 The model

Here we set up the general framework needed to characterize a random walk on graphs. This will be used in all the next sections were several applications will be studied.

Given a random graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with $V = |\mathcal{V}|$ nodes and $E = |\mathcal{E}|$ edges, we denote the neighbourhood of a node $i \in \mathcal{V}$ by ∂i, and its degree, i.e. the number of neighbours, by $k_i = |\partial i|$. An overall characterization of the graph topology is then provided by the distribution of the degrees k_i, which we write as $P(k)$. A more detailed study of different network topologies is given in appendix 3.

Introducing matrix notation we define the graph adjacency matrix A as the matrix of dimension $N \times N$ with entries

$$a_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

The nonzero entries of A thus indicate which pairs of nodes are connected by an edge. We do not consider self-loops, thus $a_{ii} = 0$. The resulting matrix is column-stochastic, meaning that:

$$\sum_i a_{ij} = 1 \tag{4.2}$$

The same statement is in general not true for rows.

Throughout the rest of the thesis we will assume that the graph is singly connected. Should the original random graph have disconnected pieces, we discard all except for the largest connected component. This requirement is usually considered when studying random walks on networks because the presence of small disconnected cluster would constraint the walker to stay on them. This would give trivial results and hinder the more interesting behavior of a walk performed on the giant cluster.

A random walk on a graph is a path $\gamma = \{i_0, i_1, \ldots, i_n\}$ made up of successive random steps between adjacent nodes i_n on the graph, starting from a given node $i_0 \in \mathcal{V}$. Steps are performed according to a transition probability from a node i to an adjacent node j given by:

$$w_{ij} = \frac{a_{ij}}{k_i} \tag{4.3}$$

in the case of unbiased walk. The biased case could be generalized by introducing a weight dependent on the adjacent node j on the right-hand side of (4.3).

All adjacent neighbours of i then have equal probability of being reached in a step starting from i. In matrix notation we define the transition matrix W as the matrix with entries w_{ij}. Defining also D as the diagonal matrix with entries $\delta_{ij} k_i$, we have the relation:

$$W = D^{-1} A \tag{4.4}$$

We denote the probability of reaching node j in n steps starting from node i as $G_{ij}(n)$.

With these definitions, given an n-step random walk $\gamma = \{i_0, i_1, \ldots, i_n\}$, the probability of reaching node i_n starting from node i_0 along this path is the product:

$$\prod_{i=0,\ldots,n-1} \frac{1}{k_i} = \frac{1}{k_0} \frac{1}{k_1} \times \cdots \times \frac{1}{k_{n-1}} \tag{4.5}$$

In general, in order to compute $G_{ij}(n)$ one has to consider all possible random walks connecting i to j in n steps. Using the transition matrix W we can write this probability as:

$$G_{ij}(n) = [W^n]_{ij} = [(D^{-1}A)^n]_{ij} \tag{4.6}$$

In this section we will derive an expression for $G(n)$, the matrix with entries $G_{ij}(n)$, where the dependence on the graph size for large graphs is explicit. Here we will for the first time have to restrict the type of graph: as explained below, we require that the eigenvalue spectrum of A has a nonzero gap.

To transform to a symmetric matrix whose properties are simpler to understand, we rewrite this as

$$\hat{G}(z) \;=\; D^{-1/2}\hat{R}(z)D^{+1/2} \tag{4.7}$$

in terms of the matrix

$$\hat{R}(z) = (\mathbb{1} - zD^{-1/2}AD^{-1/2})^{-1} \tag{4.8}$$

This matrix is now clearly symmetric, and we can diagonalize it as

$$\hat{R} = P\mathbf{\Lambda}P^T \tag{4.9}$$

where the matrix P has as columns the eigenvectors of $\hat{R}$ and $\mathbf{\Lambda}$ is a matrix containing the eigenvalues of $\hat{R}$ on the diagonal.

In terms of the normalized adjacency matrix $M = D^{-1/2}AD^{-1/2}$ [63], one has

$$\hat{R}(z) = (\mathbb{1} - zM)^{-1} \tag{4.10}$$

In the following we use Dirac bra-ket notation [64] to denote the eigenvectors $|u_k\rangle$ of M. If $|u_k\rangle$ is one such eigenvector and λ_k the corresponding eigenvalue, then we have the eigenvalue equation:

$$M\,|u_k\rangle = \lambda_k\,|u_k\rangle \tag{4.11}$$

and it follows that

$$\hat{R}(z)\,|u_k\rangle \;=\; (1 - z\lambda_k)^{-1}\,|u_k\rangle \tag{4.12}$$

In words, $\hat{R}(z)$ has the same eigenvectors $|u_k\rangle$ as M but with corresponding eigenvalues $1/(1 - z\lambda_k)$.

From spectral graph theory [63] we know that the z-independent matrix M has eigenvalues all lying in the range $[-1, 1]$. By direct substitution into the eigenvalue equation for M one sees that the vector with entries $u_{1,i} = c\sqrt{k_i}$ is an eigenvector with eigenvalue $\lambda_1 = 1$. The constant c is found from the normalization condition $\langle u_1|u_1\rangle = \sum_{i=1}^{V} u_{1,i}^2 = 1$ as $c^{-1} = \sqrt{V\,\langle k\rangle}$ where $\langle k\rangle = \sum_{j\in V} k_j/V$ is the average degree of the graph. If the graph is singly connected then there are no other eigenvectors with eigenvalue 1, so we can order the eigenvalues as

$$1 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_V \geq -1 \tag{4.13}$$

The fact that the eigenvalues lie between -1 and 1 can also be seen from the Perron-Frobenius theorem [65, 66], given that the entries of $|u_1\rangle$ are all positive and $\lambda_1 = 1$. The theorem in fact states [67] that a positive matrix always admits an eigenvalue λ_1 which is grater than all the others' moduli. Moreover, the eigenvector $|u_1\rangle$ corresponding to this maximal λ_1 has all positive entries $u_{1i} > 0$.

Splitting off the contribution coming from λ_1, we can now write the eigenvector decomposition of $\hat{R}(z)$ as

$$\hat{R}(z) \;=\; |u_1\rangle\,\langle u_1|\,\frac{1}{1-z} + \sum_{k=2}^{V} |u_k\rangle\,\langle u_k|\,\frac{1}{1-z\lambda_k} \tag{4.14}$$

and clearly the first term will be dominant in the limit $z \to 1$ that we will need to consider in section 4.2.

With the shorthand

$$C(z) \;=\; \sum_{k=2}^{V} |u_k\rangle\,\langle u_k|\,\frac{1}{1-z\lambda_k} \tag{4.15}$$

for the second term, we can then write

$$\hat{R}_{ij}(z) = \frac{\sqrt{k_i k_j}}{V\langle k\rangle}\frac{1}{1-z} + C_{ij}(z) \tag{4.16}$$

From equation (4.7) we have:

$$\hat{G}_{ij}(z) = \sqrt{\frac{k_j}{k_i}}\hat{R}_{ij}(z) \tag{4.17}$$

so the analogous representation for $\hat{G}(z)$ reads:

$$\hat{G}_{ij}(z) = \frac{k_j}{V\langle k\rangle}\frac{1}{1-z} + \sqrt{\frac{k_j}{k_i}}C_{ij}(z) \tag{4.18}$$

4.2 The average number of distinct sites visited by a random walker

In this section we will study an application of the formalism introduced above. Namely we will analyze the large-n behavior of the average number of distinct sites $S(n)$ visited by a random walker on large networks. In particular we are interested to find an expression for the constant prefactor B which characterizes the asymptotic linear behavior $S(n) = Bn$, in terms of the graph topology. To achieve this goal we will first use the generating functions formalism to relate the graph adjacency matrix to the prefactor. Secondly we will derive message-passing equations to calculate its value through numerical simulations. The analysis of numerical results will be done showing scaling behaviors with system size and graph connectivity.
Solving this problem could give important insights on the geometry of the coverage of nodes of the graphs performed by a random walker. It finds direct applications in a variety of fields: from chemistry with target decay [68] and trapping problems [69] in chemical reactions and in the problem of annealing of point defects in crystals [70], to relaxation problems in disordered systems [71] or in problems of dynamics on the internet [72, 73].

4.2.1 Results on various topologies

We denote by $F_{ij}(n)$ the probability of reaching site j for the first time after n steps for a random walk starting at site i; note that for the case $i = j$ we define "reaching" as "returning to" so that $F_{ii}(0) = 0$. We also define $H_{ij}(n)$ as the probability that site j has been visited at least once in n steps by a random walker starting at site i, and let $q_j(n)$ be the probability that a walker starting at site j does not return to it within n time steps.

With these definitions the average number of distinct sites visited by time n (i.e. after n steps), starting at node i, can be written as:

$$S_i(n) = \sum_{j\in\mathcal{V}} H_{ij}(n) \tag{4.19}$$

Now if a node j has been visited at least once in a walk of n steps starting at node i, we can call the time of the final visit of the walk $m \leq n$ and by definition the walk then never returns to j in the remaining $n - m$ steps.

Thus we can write the convolution:

$$H_{ij}(n) = \sum_{m=0}^{n} G_{ij}(m)q_j(n-m) \tag{4.20}$$

The generating function (or z-transform) of a quantity $f(n)$ is defined as $\hat{f}(z) = \sum_{n=0}^{\infty} z^n f(n)$, with $z \in [0, 1)$, and has the property that the z-transform of a convolution is the product of

the z-transforms. The z-transform of (4.20) is then

$$\hat{H}_{ij}(z) = \hat{G}_{ij}(z)\hat{q}_j(z) \tag{4.21}$$

We now want to write everything in terms of $\hat{G}_{ij}(z)$ and so need to find a relation linking $\hat{q}_j(z)$ to $\hat{G}_{ij}(z)$, which we do via the first passage time probability $F_{jj}(n)$. The probability of returning to node j for the first time after exactly n steps can be written as:

$$q_j(n-1) - q_j(n) = F_{jj}(n) \tag{4.22}$$

Taking the z-transform of this expression and noting that $q_j(0) = 1$, $\hat{q}_j(z) = \sum_{n=0}^{\infty} z^n q_j(n)$ and $\hat{F}_{jj}(z) = \sum_{n=1}^{\infty} z^n F_{jj}(n)$ we have:

$$z\sum_{n=1}^{\infty} q_j(n-1)z^{n-1} - \sum_{n=1}^{\infty} q_j(n)z^n = \sum_{n=1}^{\infty} F_{jj}(n)z^n \tag{4.23}$$

$$z\hat{q}_j(z) - [\hat{q}(z) - 1] = 1 - (1-z)\hat{q}_j(z) = \hat{F}_{jj}(z) \tag{4.24}$$

Hence:

$$\hat{q}_j(z) = \frac{1 - \hat{F}_{jj}(z)}{1 - z} \tag{4.25}$$

We now relate the generator $G_{jj}(n)$ to the first passage time probability $F_{jj}(n)$. The probability of arriving at node j in n steps starting at the same node j, can be seen as the sum of the probabilities grouped according to how often j is visited overall: we can reach j for the first time after n steps; or a first time at $n_1 < n$ and a second time after another $n - n_1$ steps; or a first time at $n_1 < n$, a second time after another $n_2 - n_1$ steps and a third time after a final $n - n_2$ steps, and so on. Mathematically this can be written as:

$$G_{jj}(n) = F_{jj}(n) + \sum_{n_1=0}^{n} F_{jj}(n_1)F_{jj}(n-n_1) + \sum_{n_2=0}^{n}\sum_{n_1=0}^{n_2} F_{jj}(n_1)F_{jj}(n_2-n_1)F_{jj}(n-n_2) + \dots \tag{4.26}$$

To make the convolution structure clearer, we have included the extreme values (e.g. $n_1 = 0$ and $n_1 = n$ in the first sum) here even though – because $F_{jj}(0) = 0$ – they do not contribute. Taking the z-transform of both sides one sees that

$$\hat{G}_{jj}(z) = 1 + \hat{F}_{jj}(z) + \hat{F}_{jj}^2(z) + \dots = \frac{1}{1 - \hat{F}_{jj}(z)} \tag{4.27}$$

Substituting this result into (4.21) using (4.25) we obtain:

$$\hat{H}_{ij}(z) = \hat{G}_{ij}(z)\frac{1 - \hat{F}_{jj}(z)}{1 - z} = \frac{1}{(1-z)}\frac{\hat{G}_{ij}(z)}{\hat{G}_{jj}(z)} \tag{4.28}$$

This can now be inserted into (4.19) to give finally the z-transform of the average number of distinct sites visited starting from site i:

$$\hat{S}_i(z) = \frac{1}{1-z}\sum_{j\in\mathcal{V}}\left[\frac{\hat{G}_{ij}(z)}{\hat{G}_{jj}(z)}\right] \tag{4.29}$$

One sees that the underlying quantity of central interest for our problem is $\hat{G}_{ij}(z)$. The result of equation (4.29) is valid in general, i.e. regardless of the graph topology. We note that to understand the large n-behaviour of $S_i(n)$ we need to consider $\hat{S}_i(z)$ near $z = 1$. Specifically, if as expected for $V \to \infty$ we have $S_i(n) = Bn$ for large n, then the z-transform will diverge for $z \to 1$ as $\hat{S}_i(z) = B/(1-z)^2$. To calculate B we thus need to understand the behaviour of $\hat{G}_{ij}(z)$ for $z \to 1$.

The problem of finding $S(n)$ in the large-n limit has been addressed for various types of topologies. For d-dimensional lattices it has been shown [74, 75, 76] that, for $d > 3$, $S(n)$ grows linearly in time as $S(n) = n/W(d)$ with a prefactor $1/W(d)$ that depends on the lattice's dimension. For smaller dimension the growth is slower: $S(n) = \sqrt{8n/\pi}$ for $d = 1$ and $S(n) =$

$\pi n/\ln n$, for $d = 2$. Further studies have characterized the same quantity when multiple walkers are moving together [77, 78].
For Bethe lattices of connectivity k the behavior is again linear and the prefactor depends on the connectivity as $S(n) = [(k-2)/(k-1)]n$ [79]. Finally the case of graphs different from lattices or random graphs has been tackles using the concept of spectral dimension $\tilde{d}$. Under certain assumptions $S_i(t) \sim t^{\min\{1,\tilde{d}\}}$ for $t \to \infty$ and $\tilde{d} \neq 2$. Examples of graphs for which it was possible to calculate the spectral dimension $\tilde{d}$ are decimable fractals, bundled structures, fractal trees and d-simplex. See [80, 81] for an overview. Nonetheless the determination of the prefactor remains an open questions for these complex types of graphs.

4.2.2 Results on networks

In this section we focus our attention of networks topologies. We will adapt the main results derived in the previous section to this case.
As we said before, few results are known about the average number of distinct sites visited by a random walker when the topology is a random network, this problem being only recently studied. In particular it has been found that for Scale-Free graphs (SF) [82, 83] (in the time regime $n \gg 1$) one recovers the linear behaviour $S(n) \sim n$ seen in both Bethe lattices and d-dimensional lattices for $d \geq 3$. However, there is very limited information on the prefactor B describing this linear behavior $S(n) = Bn$ on random networks. Indeed all the studies referred to above are based on a scaling ansatz and on the analysis of numerical simulations; neither provides a theoretical framework that fully characterizes the prefactor B to the same extent as has been achieved for lattices.
We start by recalling the form (4.18) taken by the entries of $\hat{G}_{ij}(z)$ in the case of networks:

$$\hat{G}_{ij}(z) = \frac{k_j}{V\langle k\rangle}\frac{1}{1-z} + \sqrt{\frac{k_j}{k_i}}C_{ij}(z) \tag{4.30}$$

This can be substituted in (4.29) to obtain:

$$\hat{S}_i(z) = \frac{1}{1-z}\sum_{j\in V}\left\{\frac{k_j}{\hat{R}_{jj}(z)V\langle k\rangle(1-z)} + \frac{\sqrt{\frac{k_j}{k_i}}C_{ij}(z)V\langle k\rangle(1-z)}{k_j + C_{jj}(z)V\langle k\rangle(1-z)}\right\} \tag{4.31}$$

This expression has to be taken in two limits: the thermodynamic $V \to \infty$ and the large time limit corresponding to $z \to 1$. The order of taking the two limits is important to get physical results, as we explain in more detail below. In the following we will consider first the limit $V \to \infty$ and then the limit $z \to 1$. Note that the denominators in the two terms of (4.31) are identical but written in two different forms that will make the limit procedure clearer.

The large V-limit is simple to take in (4.16), giving $\lim_{V\to\infty}\hat{R}_{jj}(z) = C_{jj}(z)$. We are assuming implicitly here that $C(z)$ has a well-defined limit for $V \to \infty$. This requires in particular that λ_2 stays away from 1, i.e. that the spectrum of M has a nonzero gap $1 - \lambda_2$ between the leading and first subleading eigenvalue for $V \to \infty$. This is generally true for regular [84, 85], ER [86, 87] and scale-free [88, 86] random graphs, but not for lattices, where the eigenvectors are Fourier modes whose eigenvalue approaches 1 smoothly in the large wavelength (zero wavevector) limit.

In the second term of (4.31), the first term in the denominator can be neglected for $V \to \infty$ at fixed $z < 1$, giving

$$\lim_{V\to\infty}\hat{S}_i(z) = \frac{1}{1-z}\sum_{j\in V}\left\{\frac{k_j}{C_{jj}(z)V\langle k\rangle(1-z)} + \frac{\sqrt{k_j}C_{ij}(z)}{\sqrt{k_i}C_{jj}(z)}\right\} \tag{4.32}$$

Now we take the limit $z \to 1$, in which the second term becomes negligible compared to the first. With the assumption of a nonzero gap, $C_{jj}(z)$ also has a finite limit for $z \to 1$ so that we

can define

$$\lim_{z\to 1}\left[\lim_{V\to\infty}\hat{R}_{jj}(z)\right]=\lim_{z\to 1}C_{jj}(z)=R_j \tag{4.33}$$

and get finally

$$\lim_{V\to\infty}\hat{S}_i(z)=\frac{1}{V\langle k\rangle(1-z)^2}\sum_{j\in V}\frac{k_j}{R_j} \tag{4.34}$$

as the asymptotic behaviour for $z\to 1$.

This has exactly the $1/(1-z)^2$ divergence we were expecting, and gives us the prefactor of the large n-asymptote of the number of distinct sites visited:

$$\lim_{V\to\infty}S_i(n)=B\,n \tag{4.35}$$

where

$$B=\frac{1}{V\langle k\rangle}\sum_{j\in V}\frac{k_j}{R_j} \tag{4.36}$$

This is the main result of this section, an expression for the topology-dependent for the constant prefactor characterizing the linear behavior (4.35) of $S_i(n)$ in the large time and thermodynamic limits.

Starting from expression (4.36) we can make three observations. Firstly, if we had inverted the order of taking the limits and fixed V while taking $z\to 1$, then we would have had $\hat{R}_{jj}(z)=k_j/[V\langle k\rangle(1-z)]$ to leading order. The second term in (4.31) would have disappeared in the limit, so that

$$\hat{S}_i(z) \;=\; \frac{1}{1-z}\sum_{j\in V}\frac{k_j}{\hat{R}_{jj}(z)V\langle k\rangle(1-z)}=\frac{1}{1-z}V \tag{4.37}$$

to leading order near $z=1$. This $1/(1-z)$ divergence of $\hat{S}_i(z)$ implies $\lim_{n\to\infty}S_i(n)=V$, a result which is clear intuitively: if we keep the graph size finite then in the limit of large times the random walk will cover the entire graph, i.e. visit all nodes at least once.

Secondly, from equation (4.33) we can see that the information one needs to calculate B resides in the quantities $C_{jj}(z)=\sum_{k=2}^{V}u_{k,j}^2/(1-z\lambda_k)$, where the $u_{k,j}$ are the components of the eigenvectors $|u_k\rangle$ of M and the λ_k the eigenvalues. So knowing the full spectrum of M and the associated eigenvector statistics would in principle solve our problem of determining B. While this is feasible computationally for finite and not too large V, we are not aware of a method that would work in the thermodynamic limit $V\to\infty$.

Thirdly, although the index i appears on the left hand side of equation (4.35), representing the initial node of the walk, it does not appear on the right. This means that the average number of distinct sites visited in the large n limit does not depend on the starting node, and therefore we can drop the index i from the left hand side of (4.35). In particular, even for graphs with broad degree distributions such as scale-free graphs, the number of distinct sites visited will be the same whether we start the walk from a hub (a node with high degree) or a dangling end of the graph (a node with degree one) – provided of course n is large enough.

Expression (4.36) shows that the quantity of interest for our problem is the ratio k_j/R_j. The degree k_i is given by the graph degree distribution $P(k)$ and if one considers single instances of a graph this quantity is fixed at the beginning. The quantity R_j instead is given by (4.33) and has to be determined. Therefore the original problem reduces to the another one: the calculation of the quantity $\hat{R}_{jj}(z)$. Unfortunately this is not a straightforward task to accomplish, even though we know the entries of the inverse $\hat{R}^{-1}_{ij}(z)=\delta_{ij}-za_{ij}(k_ik_j)^{-1/2}$.

Indeed we could find the value R_j either by calculating $\lim_{z\to 1}C_{jj}(z)$ where $C_{jj}(z)=\sum_{k=2}^{V}u_{k,j}^2/(1-z\lambda_k)$ or by directly inverting the matrix $\hat{R}^{-1}(z)=[\mathbb{1}-zD^{-1/2}AD^{-1/2}]$. However both of these two methods are prohibitive computationally, already for individual graphs of large size V and

even more so if in addition we want to average the results over an ensemble of random graphs. The aim of this section, then, is to find a viable alternative method that will allow us to characterize the value of $\hat{R}_{jj}(z)$, and thus calculate $\lim_{n\to\infty} S(n)$ through (4.35) and (4.36).

In the following we will propose and describe a method to address this problem of characterizing the diagonal entry $\hat{R}_{jj}(z)$. This has already been deployed in the context of spectral analysis to calculate the sparse random matrix spectra [89].
That a connection to spectral problems should exist is suggested by looking at the definition of the resolvent $R(z, M)$ of a random matrix M. This is defined as:

$$R(z, M) = (M - z\mathbb{1})^{-1} \tag{4.38}$$

Therefore the shape of $z\hat{R}(z) = (z^{-1}\mathbb{1} - D^{-1/2}AD^{-1/2})^{-1}$ suggests that, up to a trivial rescaling, $\hat{R}(z)$ has the structure of a resolvent (with parameter z^{-1}) for the random matrix $D^{-1/2}AD^{-1/2}$. It is from such resolvents that spectral information is normally derived, in an approach that in the statistical physics literature goes back to at least Edwards and Jones [90]. Accordingly the two steps we will need to take mirror closely those used to find resolvents of sparse random matrices in [89]: we first write the $\hat{R}_{jj}(z)$ as variances in a Gaussian distribution with covariance matrix $\hat{R}^{-1}(z)$, and then exploit the fact that this distribution has a graphical model structure to derive cavity equations from which these variances can be found.

4.2.3 The auxiliary multivariate Gaussian distribution

As we pointed out before, we have en exact expression for the inverse $\hat{R}^{-1}(z) = [\mathbb{1} - zD^{-1/2}AD^{-1/2}]$, but inverting it to find a similar expression for $\hat{R}(z)$ as a function of the adjacency matrix A is not straightforward. Thus we may prefer to work directly with $\hat{R}^{-1}(z)$ and from this derive information about the diagonal entries $\hat{R}_{jj}$ (4.33). We can achieve this by considering the definition of a multivariate Gaussian distribution.
A multivariate Gaussian random variable is a n-dimensional vector $\bar{x} = (x_1, \ldots, x_n)$, with entries x_i being random variables such that their joint distribution $P(\bar{x}) := P(x_1, \ldots, x_n)$ is Gaussian [15]:

$$P(\bar{x}) \propto e^{-\bar{x}^T \Sigma^{-1} \bar{x}/2} \tag{4.39}$$

where the matrix Σ is called the covariance matrix and has the property that $\Sigma_{ij} = \langle (x_i - \mu_i)(x_j - \mu_j) \rangle$, where μ_1, μ_j are the mean values of the variables x_i, x_j respectively; the symbol $\propto$ means proportional to, thus we omitted the normalization. This implies that the variance of a zero-mean variable x_j is $\langle x_j^2 \rangle = \Sigma_{jj}$. Moreover we have the property [15] that the marginal probability distribution $P(x_j)$ of the single random variable x_j, which is defined as:

$$P(x_j) := \int P(\bar{x}) \prod_{i \neq j} dx_i \tag{4.40}$$

is itself Gaussian distributed. Therefore we can write:

$$P(x_j) \propto e^{-\frac{(x_j - \mu_j)^2}{2v_j}} \tag{4.41}$$

where $\mu_j = \langle x_j \rangle$ is the mean value and $v_j = \langle x_j^2 \rangle$ is the variance.
In the expression (4.39) it appears the inverse Σ^{-1} of the covariance matrix, but the variance of a zero-mean variable $\langle x_j^2 \rangle$ is equal to the diagonal entry of Σ itself. The analogy with our problem now becomes clear: we have an expression for the inverse $\hat{R}^{-1}(z)$ but the quantity of interest is the diagonal entry $R_j = \lim_{z\to 1} \left[\lim_{V\to\infty} \hat{R}_{jj}(z) \right]$ (4.33). This suggest us that if we were able to consider $\hat{R}(z)$ as the covariance matrix Σ of a multivariate Gaussian distribution we could characterize the joint probability distribution (4.39) using its known inverse $\hat{R}^{-1}(z)$ and then explicitly find the variances $\langle x_j^2 \rangle$ to obtain our quantity of interest $R_{jj}(z)$ through

the relation $\hat{R}_{jj} = \Sigma_{jj} = \langle x_j^2 \rangle$; in this way we could obtain the diagonal entries R_j without performing an expensive matrix inversion. Fortunately we can make this characterization.
Indeed if a matrix is symmetric and positive semi-definite then it can be a candidate for a covariance matrix [15] of a multivariate Gaussian distribution. Given that, for $0 \leq z < 1$, $\hat{R}(z) = [\mathbb{1} - zD^{-1/2}AD^{-1/2}]^{-1}$ is symmetric with all non-negative eigenvalues (4.13), then it is positive semi-definite and it can be a considered as the covariance matrix associated to a multivariate distribution of the form (4.39).
We can thus define a mapping from our original problem of finding the entries of $\hat{R}(z)$ to an auxiliary problem of finding the variances of N random variables x_i jointly distributed through a multivariate Guassian distribution. The link between these two problems is given by the equation:

$$\hat{R}_{jj}(z) = \Sigma_{jj} = \langle x_j^2 \rangle \tag{4.42}$$

It follows that the quantity of interest in this auxiliary problem is the variance $\langle x_j^2 \rangle$.
We start by defining a vector of random variables $(x_1, \ldots, x_V)$ and assign to this the zero mean Gaussian distribution with covariance matrix $\hat{R}(z)$:

$$P(\bar{x}) \quad \propto \quad e^{-\bar{x}^{\mathrm{T}}\hat{R}^{-1}(z)\bar{x}/2} = e^{-\bar{x}^T(\mathbb{1}-zD^{-1/2}AD^{-1/2})\bar{x}/2} \tag{4.43}$$

The marginal distribution of any component of the vector, obtained by integrating $P(\bar{x})$ over all other components, is then also Gaussian:

$$P(x_j) \propto e^{-x_j^2/(2v_j)} \tag{4.44}$$

with variance $v_j = \langle x_j^2 \rangle = \hat{R}_{jj}(z)$. Our goal is now to calculate these marginal variances efficiently, i.e. without a full matrix inversion. The key property of the probability distribution (4.43) is that it can be written in the form

$$P(\bar{x}) \quad = \quad \prod_{i \in \mathcal{V}} e^{-x_i^2/2} \prod_{(ij) \in \mathcal{E}} e^{z x_i x_j (k_i k_j)^{-1/2}} \tag{4.45}$$

As this factorizes into contributions associated with the nodes and edges of the underlying graph, is an example of a graphical model similar to the one (4.45) introduced in section 1.1. In that chapter we have seen that in such a graphical model marginal distributions can be obtained using message-passing, or cavity, equations. In this case the situation simplifies though, because the general expression for a factor graph representation (4.45) reduces to the one of a pair-wise model where the functional nodes $\psi_a(x_{\partial a})$ is a function of only two variables of the form:

$$\psi_a(x_{\partial a}) \sim \psi_{ij}(x_i, x_j) \tag{4.46}$$

The $\phi_i(x_i)$ can be treated as an external field acting on variable x_i.

In our case we have:

$$\psi_{ij}(x_i, x_j) \quad = \quad e^{z x_i x_j (k_i k_j)^{-1/2}} \tag{4.47}$$

$$\phi_i(x_i) \quad = \quad e^{-x_i^2/2} \tag{4.48}$$

In this case one needs only one type of message as in (1.8). This auxiliary graphical model is represented in figure 4.1.

To calculate the marginal distribution of x_j, we follow the procedure described in section 1.1 where we suppose the graph is a tree and imagine to first remove all edge factors $\psi_{ij}(x_j, x_i)$ from $P(\bar{x})$, where i runs over all neighbours of j. The tree is now split into subtrees rooted at each neighbour i, and one can define the cavity marginal of i, $\nu_{i \to j}(x_i)$ as the marginal that is obtained from a (suitably renormalized) probability distribution containing only the factors from the relevant subtree. To get the marginal of x_j, we now just need to reinstate the missing edge factors as well as the node factor at j and integrate over the values of the nodes that we

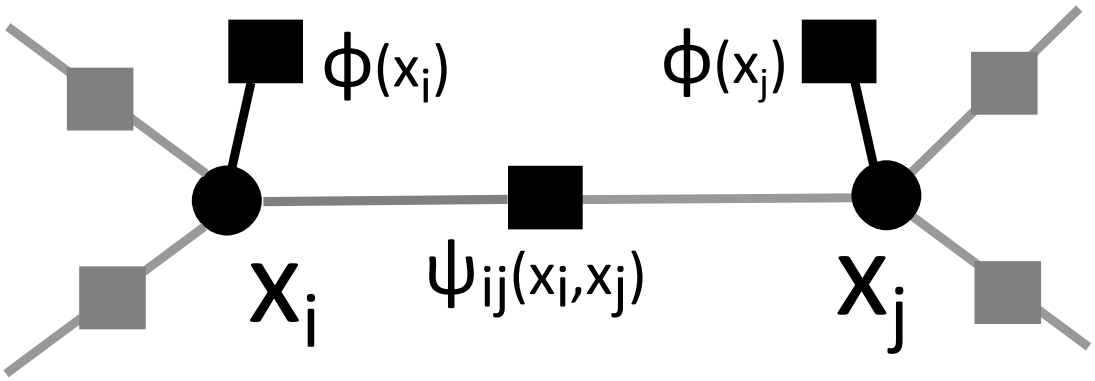


Figure 4.1: Multivariate Gaussian factor graph. The auxiliary system of jointly multivariate Gaussian variables x_i is represented as a pairwise graphical model. The variances $\langle x_i^2 \rangle$ are the quantity of interest because they correspond to $\hat{R}_{jj}(z)$ (4.42).

have not yet marginalized over, namely, the neighbours i:

$$P(x_j) \propto \phi_j(x_j) \prod_{i \in \partial j} \int dx_i \, \psi_{ji}(x_j, x_i) \nu_{i \to j}(x_i) \tag{4.49}$$

Our messages sent from i to j are the quantities $\nu_{i \to j}(x_i)$, also called cavity marginals: each message tells node j what the marginal of its neighbour i would have been if the edge between them had been severed.

The cavity marginals can now be obtained from an analogous relation. To get $\nu_{i \to j}(x_i)$, one can think of removing all edges connecting i to its neighbours l other than j; note that the edge connecting i to j has already been taken out in the definition of the cavity marginal. This generates independent subtrees rooted at the neighbours l, and the marginals at these nodes are $\nu_{l \to i}(x_l)$. Reinstating removed edge factors and marginalizing over neighbours then yields

$$\nu_{i \to j}(x_i) \propto \phi_i(x_i) \prod_{l \in \partial i \setminus j} \int dx_l \, \psi_{il}(x_i, x_l) \nu_{l \to i}(x_l) \tag{4.50}$$

One can then sweep through the tree in a way that calculates each message once messages have been received from all neighbours except the intended recipient of the message. The structure of equation (4.50) suggests that a convenient way to perform the iteration is to start at leaf nodes, where simply $\nu_{i \to j}(x_i) \propto \phi_i(x_i)$. Once all messages have been found, the marginals can be deduced from (4.49).

As explained in section 1.1, expression (4.50) is exact on trees because of the absence of loops, i.e. of correlations between neighbors of a given node. On a general graph this is an approximate expression which becomes more and more accurate for locally tree-like structures. This is the case of random graphs in the thermodynamic limit $V \to \infty$ because loops, which correspond to correlations, increase weakly (logarithmically) in V thus leading to a locally tree-like topology.

Specializing now to our auxiliary Gaussian graphical model, the cavity marginals must also be Gaussian as follows from a general property of multivariate Gaussian distributions. We can then write them as:

$$\nu_{l \to i}(x_l) \propto e^{-x_l^2/(2v_l^{(i)})} \tag{4.51}$$

which defines the cavity variances $v_l^{(i)}$.

Now consider the integral inside (4.50), note that thanks to the properties of Gaussian

integrals we have:

$$\int dx_l\, \nu_{l\to i}(x_l) e^{z\frac{x_l x_i}{\sqrt{k_l k_i}}} = \int dx_l\, e^{-\frac{x_l^2}{2v_l^{(i)}}} e^{z\frac{x_l x_i}{\sqrt{k_l k_i}}} \tag{4.52}$$

$$\propto e^{\frac{z^2}{2}\frac{x_i^2 v_l^{(i)}}{k_l k_i}} \tag{4.53}$$

where we have omitted the normalization.
Using this result we can develop further the product in (4.50) to obtain:

$$\nu_{i\to j}(x_i) \propto e^{-\frac{1}{2}x_i^2} \prod_{l\in\partial i\setminus j} e^{\frac{z^2}{2}\frac{x_i^2 v_l^{(i)}}{k_l k_i}} = e^{-\frac{1}{2}x_i^2[1-\frac{z^2}{k_i}\sum_{l\in\partial i\setminus j}\frac{v_l^{(i)}}{k_l}]} \tag{4.54}$$

which resembles the typical formula of a zero-mean Gaussian distribution with variance given by:

$$v_i^{(j)} = k_i\left(k_i - z^2\sum_{l\in\partial i\setminus j}\frac{v_l^{(i)}}{k_l}\right)^{-1} \tag{4.55}$$

Repeating the same calculations as in (4.54) but extending the product over all the neighbors, we obtain the expression for the marginal (4.49):

$$P(x_j) \propto e^{-x_j^2/2} \prod_{i\in\partial j} e^{\frac{z^2}{2}\frac{x_j^2 v_i^{(j)}}{k_j k_i}} = e^{-\frac{1}{2}x_j^2[1-\frac{z^2}{k_j}\sum_{i\in\partial j}\frac{v_i^{(j)}}{k_i}]} \tag{4.56}$$

which has analogously the form of a zero-mean Gaussian variable with variance:

$$v_j = k_j\left(k_j - z^2\sum_{i\in\partial j}\frac{v_i^{(j)}}{k_i}\right)^{-1} \tag{4.57}$$

These two relations are the direct analogues of Eqs. (11) and (12) in [89].

The variances v_j, when calculated in the limit $z\to 1$, are the quantity of interest for our problem as $v_j = \langle x_j^2\rangle = R_j$. They are known once the cavity variances have been obtained by solving (4.55).

In practice we use the rescaled cavity variances

$$m_{i\to j} = \frac{v_i^{(j)}}{k_i} \tag{4.58}$$

as messages from node i to node j. With this definition and using (4.55) for $z\to 1$ the cavity equations are:

$$m_{i\to j} = \left(k_i - \sum_{l\in\partial i\setminus j} m_{l\to i}\right)^{-1} \tag{4.59}$$

We solve these by iteration according to

$$m_{i\to j}^{(t+1)} = \left(k_i - \sum_{l\in\partial i\setminus j} m_{l\to i}^{(t)}\right)^{-1} \tag{4.60}$$

where t represents a discrete iteration time step. This equation corresponds to the general pairwise-model message-passing equation (1.8) encountered in section 1.1.
Starting from a given graph $\mathcal{G}$, a suitably chosen convergence criterion and a maximum iteration time $T_{\max}$, the algorithm then works as following:

1. Initialize the messages $m_{i\to j}^{(0)}$ randomly.

2. Run through all edges (ij) and find for each the updated messages $m_{i\to j}^{(t+1)}$, $m_{j\to i}^{(t+1)}$ from (4.60).

3. Increase t by one.

4. Repeat steps 2 and 3 until either convergence is reached or $t = T_{\max}$.

If convergence is reached, i.e. the preset convergence criterion is satisfied, one can collect the results and calculate the variances v_j using (4.57) and (4.58):

$$v_j = k_j \left(k_j - \sum_{i \in \partial j} m_{i\to j} \right)^{-1} \tag{4.61}$$

where $m_{i\to j}$ are the converged messages.

If we identify $v_j = \langle x_j^2 \rangle = R_j$ we can then also express directly the prefactor (4.36) in the linear asymptote in the number of distinct sites visited, $S(n) = Bn$, as

$$B = \frac{1}{V\langle k \rangle} \sum_{j \in V} \frac{k_j}{v_j} \tag{4.62}$$

$$= \frac{1}{V\langle k \rangle} \sum_{j \in V} \left(k_j - \sum_{i \in \partial j} m_{i\to j} \right) \tag{4.63}$$

There is one subtlety here that we have glossed over: the variances v_j are the full marginal variances $\hat{R}_{jj}(z)$, which from (4.16) have the form $k_j/[V\langle k \rangle(1-z)] + C_{jj}(z)$. In the calculation of B we need $R_j = \lim_{z\to 1} C_{jj}(z)$, where the contribution $\propto (1-z)^{-1}$ has been removed. Above we have taken the limit $z \to 1$. This requires implicitly that $1 - z$ needs to lie in the range $1/V \ll 1 - z \ll 1$ where the divergent contribution to $\hat{R}_{jj}(z)$ is still small enough to be neglected compared to $C_{jj}(z)$. That it is then allowable nevertheless to set $z = 1$ directly in the cavity equations that we solve is something that has to be checked numerically: we do indeed always find finite marginals v_j from converged solutions for the cavity marginals. The divergent solution also exists as a separate fixed point, namely the trivial solution $m_{i\to j} \equiv 1$ of (4.59), but is not accessed in this iterative solution method.

Regular graph case.

For a generic random topology equations (4.60) are difficult to solve analytically. Nevertheless for random regular graphs, for which the degree is the same k for all nodes, one can suppose that the nodes are all equivalent. Therefore one can assume that the messages and all the quantities of interest are as well all the same and thus we can drop the indices:

$$k_i = k \tag{4.64}$$

$$m_{i\to j} = m \tag{4.65}$$

$$v_j = v \tag{4.66}$$

First of all we substitute into (4.59) to get:

$$m = \left(k - \sum_{l \in \partial i \setminus j} m \right)^{-1}$$

$$= [k - (k-1)m]^{-1} \tag{4.67}$$

We obtain a second order equation in m:

$$m^2(k-1) - mk + 1 = 0 \tag{4.68}$$

with solutions $m = 1/(k-1)$ or $m = 1$. The first solution is the one we require; the second one is the trivial solution discussed above that gives divergent variances in (4.61). From $m = 1/(k-1)$

one can find the cavity variances and from there the full variances. In fact, substituting this solution into (4.61) we get:

$$\begin{aligned} v &= k\,[k - \sum_{k\in\partial i} m]^{-1} \\ &= k\,[k - km]^{-1} \\ &= k\,\left[k\left(1 - \frac{1}{k-1}\right)\right]^{-1} \end{aligned}$$

So that we finally obtain the full variances:

$$v = \frac{k-1}{k-2} \tag{4.69}$$

We substitute into the expressions (4.62) for the prefactor B to obtain:

$$\begin{aligned} B &= \frac{1}{Vk}\sum_{j\in V}\frac{k}{v} \\ &= \frac{1}{Vk}\frac{Vk(k-2)}{k-1} \end{aligned}$$

And eventually the prefactor becomes:

$$B = \frac{k-2}{k-1} \tag{4.70}$$

This result agrees with the one derived for Bethe lattices of connectivity k [79]. This is as expected, given that the cavity method is exact on tree graphs.
Therefore we can conclude that the large time limit of the average number of distinct sites of a random walk on a k-regular graph is:

$$\lim_{n\to\infty} S(n) = \left(\frac{k-2}{k-1}\right) n \tag{4.71}$$

4.2.4 Numerical results

We performed numerical simulations to test the predictions from our cavity approach for the number of distinct sites visited. We used four types of graph structures: regular random graphs (Reg), Erdős-Rényi (ER) [33], scale-free (SF) using a preferential attachment scheme [83] and the RER. The detailed definition of these topologies can be found in appendix 3. For each of these graph topologies we investigated three fixed sizes $V = 10^3, 10^4, 10^5$ and different average degrees. For ER graphs we only used the giant connected component of each graph sampled, but the average degrees we consider are large enough ($\langle k\rangle \geq 4$) for this to reduce V by at most by 2%. The other types of graph have only one connected component by construction.
We performed two types of simulations.
The first one aims at testing the validity of the cavity predictions (4.63), we denote this MP (message-passing). Therefore we evaluated the cavity equations by iterating equation (4.59). These depend on the graph topology through the realization of $\{k_i\}$ of a given graph instance. The iterative solution converged quickly, in typically around 10 iteration steps. Once convergence was reached, we collected the cavity messages to calculate the full marginal through (4.61). As for the convergence criterium we used the following rule: convergence is reached if for y consecutive times we have:

$$\max_{(ij)\in\mathcal{E}} |m_{i\to j}^{(t+1)} - m_{i\to j}^{(t)}| < \epsilon \tag{4.72}$$

where we set $y = 10$ and $\epsilon = 10^{-5}$. Finally, the results for B were averaged over $1,000$ different graph instances for $V = 10^3, 10^4$ and 100 instances for the bigger graphs of size $V = 10^5$.
The second type of simulation is a Markov chain simulation of an unbiased random walk, we

denote it by RW (random walk). Starting from the same graph instances of the MP simulations, a starting node $i_0 \in \mathcal{V}$ is chosen randomly and then a walk is performed as a series of jumps with transition probability $w_{ij} = a_{ij}/k_i$ as in (4.3). We keep track of the number of distinct visited sites as the walk progresses. For each instance of a given graph type, only a single walk was performed starting from a randomly chosen initial site. From this description we can see that in this case there is an additional source of randomness arising from the particular random walk trajectory that is obtained on a given graph. On the contrary, in the MP simulations randomness was due only to the topology of each graph.

Here we briefly comment on how the cavity predictions depend on graph size V. We argued that the method should become exact in the limit $V \to \infty$, and so a priori should extrapolate our predictions for B to this limit. We found, however, that for our relatively large graph sizes the predictions for different V agreed within the error bars. Thus we did not perform a systematic extrapolation and simply used the predictions for $V = 10^4$, as the largest graph size for which we could obtain a statistically large sample (1000 graph instances) of data. The fact that already $V = 10^3$, our smallest size, is large enough to obtain results that are essentially indistinguishable from those for $V \to \infty$ is consistent with findings from cavity predictions in other contexts, see e.g. [91, 92].
An alternative approach to evaluating the cavity predictions would have been to move from specific graph instances to solving the limiting ($V \to \infty$) integral equations for the distribution of messages across the graph. These equations can be read off more or less directly from the cavity equations, see e.g. [91, 93], or obtained from replica calculations [94] and then solved numerically using population dynamics. Given the good agreement between the predictions for our three different V this approach would be expected to give identical predictions, so we did not pursue it.

Our first task is to verify that the cavity equations do indeed correctly predict the prefactor B for random walks on large graphs, therefore we compared the average number of sites $S(n)$ obtained from the Markov chain RW versus Bn, with B the value taken from the cavity MP simulations. If these two agree then the data points should lie on the diagonal $y = x$. We see in figure 4.2 that this is indeed the case, for graphs of size $V = 10^4$. Here we used ER graphs of different average connectivities $k = 4$, 7 and 10. Similar levels of agreement are obtained for the other graph ensembles and sizes. We conclude that the numerical data thus fully support our argument that the cavity predictions will be exact for large V, and show that in fact V does not have to be excessively large to reach good quantitative agreement between the predictions and direct simulations.

4.2.5 Dependence on graph topology.

We next look more systematically at how the prefactor B in the large n-behaviour $S(n) = Bn$ depends on the topology of the graphs we study. In figure 4.3 we report the dependence of the cavity prediction for B on average node degree $\langle k \rangle$, for the four different graph ensembles we studied. We found that for each graph type a hyperbolic fit of the form $B(\langle k \rangle) = \frac{\langle k \rangle - c_1}{\langle k \rangle - c_2}$ gives a good description of the data, with the parameters c_1, c_2 dependent on the graph topology but best fit values always satisfying $c_1 = c_2 + 1$. Thus we could interpret the generic graph result as the one for a regular graph with effective degree $\langle k \rangle - c_2 + 1$. This is intriguing as it suggests that the effect of changing the average degree is quite similar between the different graph types.

Comparing the various graph ensemble we found that the prefactor B is smallest for regular graphs, given a fixed average connectivity. This finding suggests that heterogeneity in the node degrees generically seems to *increase* the number of distinct sites a random walk will visit, a result that seems to us non-trivial. We argue that it would be interesting to investigate as a broader conjecture: could there be a lower bound $B \geq (\langle k \rangle - 2)/(\langle k \rangle - 1)$?

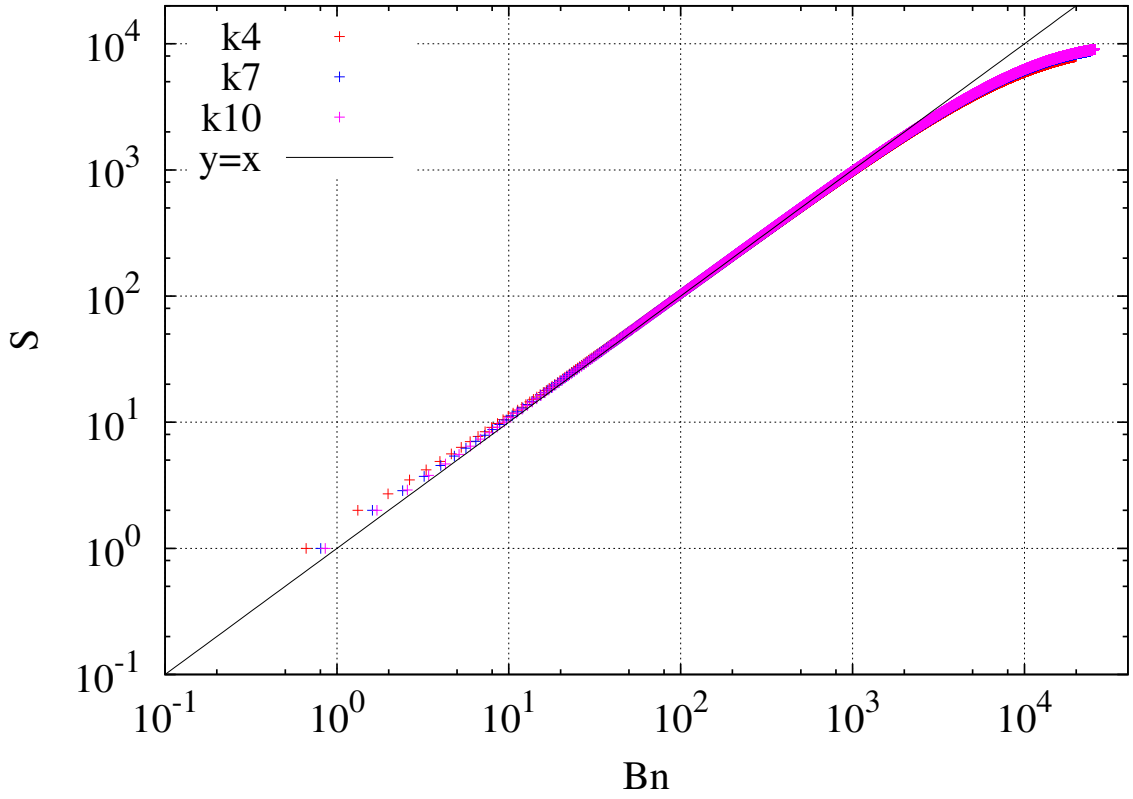


Figure 4.2: Average number of distinct sites visited, $S(n)$ for random walks on ER graphs of size $V = 10^4$. $S(n)$ is plotted against Bn with the prefactor B as predicted by the cavity method (4.63), for different average degrees $\langle k \rangle = 4, 7, 10$ as shown in the legend. In the linear regime, before the random walk starts to saturate the graph, data points lie on the diagonal, showing excellent agreement between predictions and direct simulations.

It is know that the spectral gap of a given graph, the difference between the first two eigenvalues, is maximal for regular random graphs [95, 96]. Thus one may wonder, if the conjecture about the lower bound were correct, whether this is related to the spectral gap.
We may attempt to argue on this by looking at the prefactor (4.36). Indeed the spectral gap appears to impact the numerator of this expression through equation (4.33) and by using the definition $C_{jj}(z) = \sum_{k=2}^{V} u_{k,j}^2/(1 - z\lambda_k)$. Nonetheless the gap contribution could be balanced off by the square of the eigenvector entries $u_{k,j}^2$ of the matrix R which can be of order $O(1)$ or $O(1/V)$ depending on the eigenvector localization or delocalization respectively. For instance scale-free graphs have been shown empirically to be localized (when considering the adjacency matrix), i.e. only a few eigenvector entries are non-zero and these correspond to the high degree nodes [86], whereas for ER graph the amplitude of the eigenvalue entries is evenly distributed among all the nodes; this difference can be detected for instance by calculating the inverse participation ratio [94, 86].
Taking all these considerations into account, we conclude that one would then need to consider these two aspects at the same time in order to make a more rigorous statement. Unfortunately the absence of a general analytical characterization for either the eigenvalues or the eigenvector entries makes this difficult.
A concluding remark regarding the topology dependence is the following. Suppose that for various graph ensembles the average degree $\langle k \rangle$ is fixed. Can we characterize the increase in B with some measure of spread of degrees such as the variance $\langle k^2 \rangle - \langle k \rangle^2$? For our admittedly

limited choice of graph ensembles it is certainly true that the scale-free graphs (SF), which have the broadest degree distributions, also give the largest B. Below them are the ER graphs. The RER graphs, finally, with their character intermediate between regular and ER, also have prefactors B that lie between those of the ER and regular graphs.

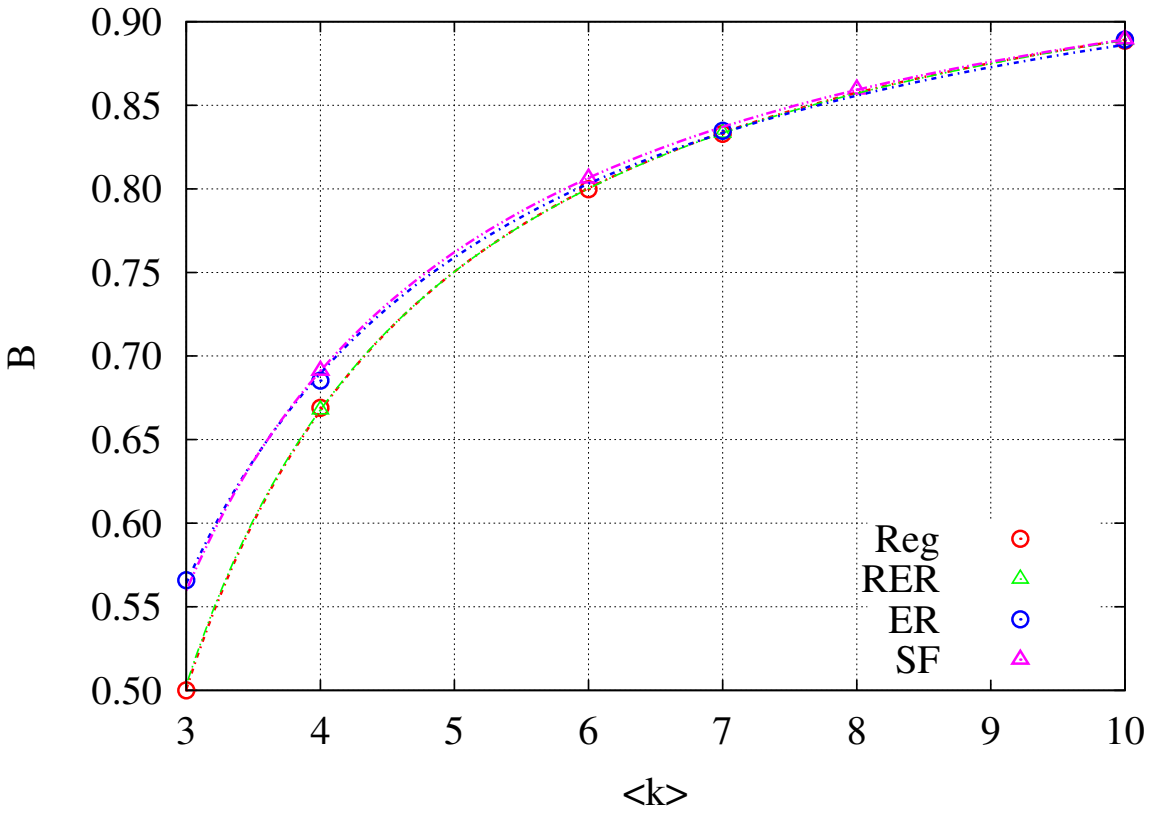


Figure 4.3: Prefactor B predicted by cavity method as a function of average degree, for different graph types as shown in the legend. The lines represent hyperbolic fits; see text for details. Note that the results for Reg and RER are essentially on top of each other, and the same is true for ER and SF.

4.2.6 Finite-size effects and scaling.

In this paragraph we interpret and use the results of the numerical simulations to describe finite-size effects. By this we mean to find a proper analytical description of the behaviour of $S(n)$ on graphs of large but finite size V. In fact our derivation of B and its prediction using cavity techniques was done taking a large V-limit so cannot make statements about this regime. Instead in the following we will comment on the numerical results using physical intuition and construct a suitable finite-size scaling ansatz that captures the different aspects provided by the numerical analysis.

A first observation concerns time. This follows from the interpretation of results plotted in figure 4.4, which shows results of $S(n)/V$ as a function of n/V for fixed graph size $V = 10^4$ and graphs with $\langle k \rangle = 4$. Here we can distinguish a number of time regimes. Initially $S(n)$ is linear in n with prefactor 1. This is greater than the large n prediction Bn with a prefactor $B < 1$, because the random walker has not yet had much opportunity to return to previous sites; in particular one has, trivially, $S(1) = 1$, ignoring the starting site v_0. A second regime is seen for larger n: in this case one finds the predicted linear growth with prefactor $B < 1$, i.e.

$S(n) = Bn$. Finally, once Bn becomes comparable to V, a crossover to sublinear growth takes place, and eventually $S(n)$ approaches V as the walker visits all sites for asymptotically large n. Plots for other graph sizes and average degrees look qualitatively identical.

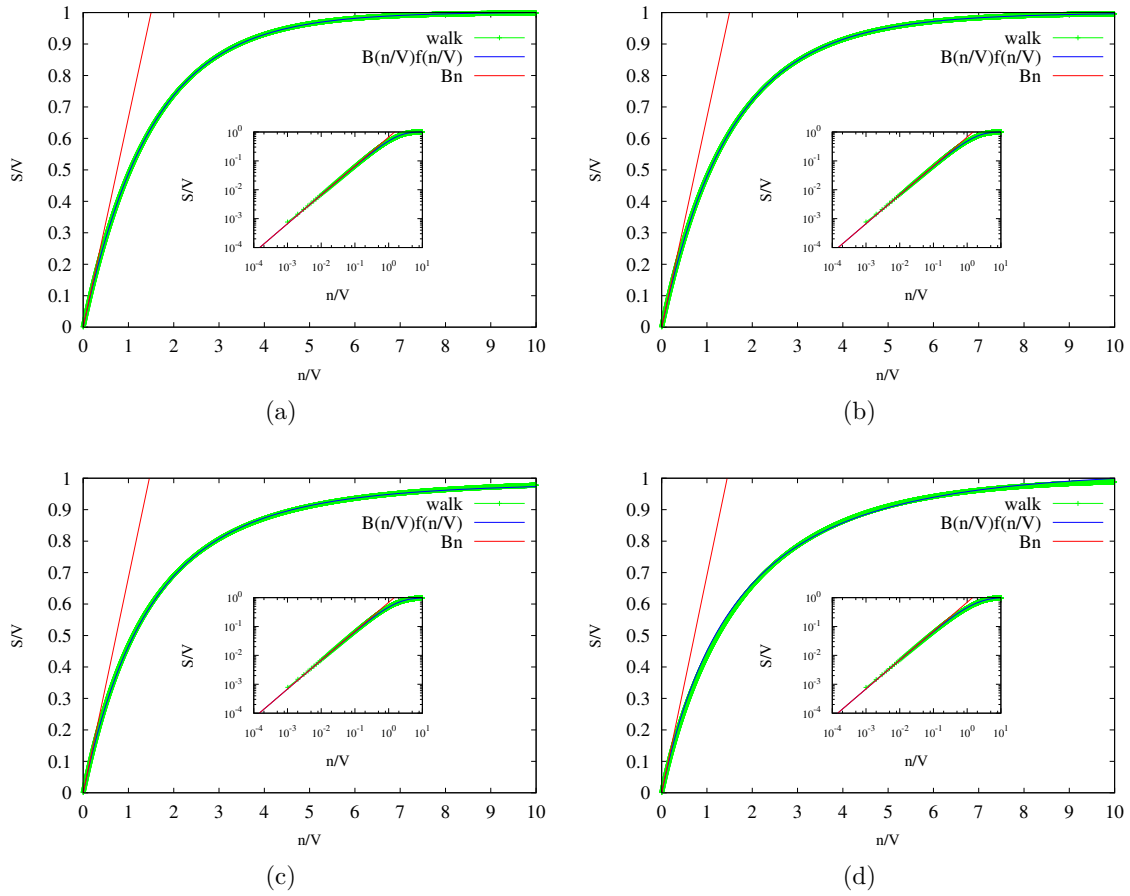


Figure 4.4: Finite size effects: we show the walker behavior by plotting $S(n)/V$, i.e. the fraction of distinct sites visited, derived from direct simulations vs n/V. Results are from averages over 1000 instances of graphs of fixed size $V = 10^4$ and average degree $\langle k \rangle = 4$, for different graph topologies: a) Regular; b) RER; c) ER; d) SF. The dashed red lines show the cavity predictions Bn for the linear growth with n, a regime which is clearer in the log-log plot insets. Beyond that one observes a slow crossover, with $S(n)/V$ eventually approaching unity. Solid lines show our phenomenological scaling fits.

These results suggested us to propose as a plausible scaling function that encompasses the various regimes (without the initial small n-piece) the following ansatz

$$S(n, V) = Bn\, f\left(\frac{Bn}{V}\right) \tag{4.73}$$

where the limiting behaviour of the scaling function must be

$$f(x) \approx \begin{cases} 1 & x \ll 1 \\ \frac{1}{x} & x \gg 1 \end{cases} \tag{4.74}$$

to reproduce $S(n, V) \approx Bn$ and $S(n, V) \approx V$ when n is much smaller and much larger than V, respectively. We tested the validity of this ansatz (4.73) by plotting $S(n)/(Bn)$ vs Bn/V with B predicted from the cavity equations. These results can be seen in figure 4.5; here we used graph sizes $V = 10^3, 10^4, 10^5$ and two values for the average degree $\langle k \rangle = 4, 10$. This

plot allows us to directly have a graphical representation of the scaling function $f(x)$. Very good agreement is seen between the three different graph sizes: these all collapse onto the same curve, except the initial regime discussed above where $S(n) \approx n$ and hence $S(n)/(Bn) > 1$. Beyond this we observe a plateau at $S(n)/(Bn) = 1$, which in a different guise verifies our claim above that the cavity method does indeed predict the prefactor B correctly. For $x = Bn/V$ growing towards unity, the curves drop below this plateau as expected, indicating the start of the saturation regime. Asymptotically the scaling function $f(x)$ then approaches $1/x$, reflecting the final saturation of $S(n)$ at the upper bound V.

The same figure reveals another interesting and unexpected result, which indeed was not required by our ansatz per se. The plots for graphs of different average degree show a good data collapse. This is suggesting that using Bn/V as the argument of the scaling function seems sufficient to absorb all the variation with $\langle k \rangle$, without further changes in $f(x)$. The only exception is provided by the scale free graphs, which we discuss in more detail below.

Encouraged by the good agreement of the numerical data with the ansatz (4.73), we attempted to find simple fits to the scaling function $f(x)$. The simulation data show that the crossover starts off with a roughly exponential departure from the small x-plateau $f(x) \approx 1$, which suggests a scaling function of the form:

$$f(x) = a/\ln(b + (e^a - b)e^{ax}) \tag{4.75}$$

where a and b are fitting parameters. Figure 4.5 shows that this form fits the data extremely well, and except for the scale-free graphs the fits can be performed even with fixed $b = 1$, leaving a single fit parameter.

The case of SF requires a separate comment. For this type of graph in fact we see results behaving differently than the other graph ensembles. Firstly, the data in figure 4.5 do not collapse perfectly for different V in the intermediate regime where $x = Bn/V$ is order unity or somewhat smaller. Secondly, the crossover in $f(x)$ is slower, with $f(x)$ lower in the crossover region than for the other three graph types. We conjecture that both of these effects are due to the presence of many small loops in SF graphs, for example triangles (loops of length 3). To support this hypothesis, we calculated the average number of triangles present in the different types of graph, taking averages over 100 graph instances of size $V = 10^3$. We found results in the same range for Reg, ER and RER graphs, where the average percentage of nodes that are part of at least one triangle does not exceed 2%, 7% and 37% for $\langle k \rangle = 4, 6, 10$ whereas for SF graphs the relevant fractions of nodes reach 9%, 24% and 51% for the same average degrees. These results confirm that SF graphs generated via preferential attachment contain a higher number of short loops than the other topologies. In fact it has been shown by spectral arguments [86] that, even though the fraction of nodes in triangles will tend to zero for $V \to \infty$, the growth rate of the number of loops of length $l \geq 4$ exceeds all polynomial growth rates, thus these graphs do not become locally treelike for large V. Therefore it is somewhat surprising that the cavity predictions for B are quantitatively accurate even for SF graphs. This feature was nonetheless already encountered in the EDP problem 2.2.1. There for very loopy graphs, meshes or planar graphs, we could still find solutions of the optimisation problem using message-passing with reinforcement. Remarkably these solutions were better than the other alternative algorithms.

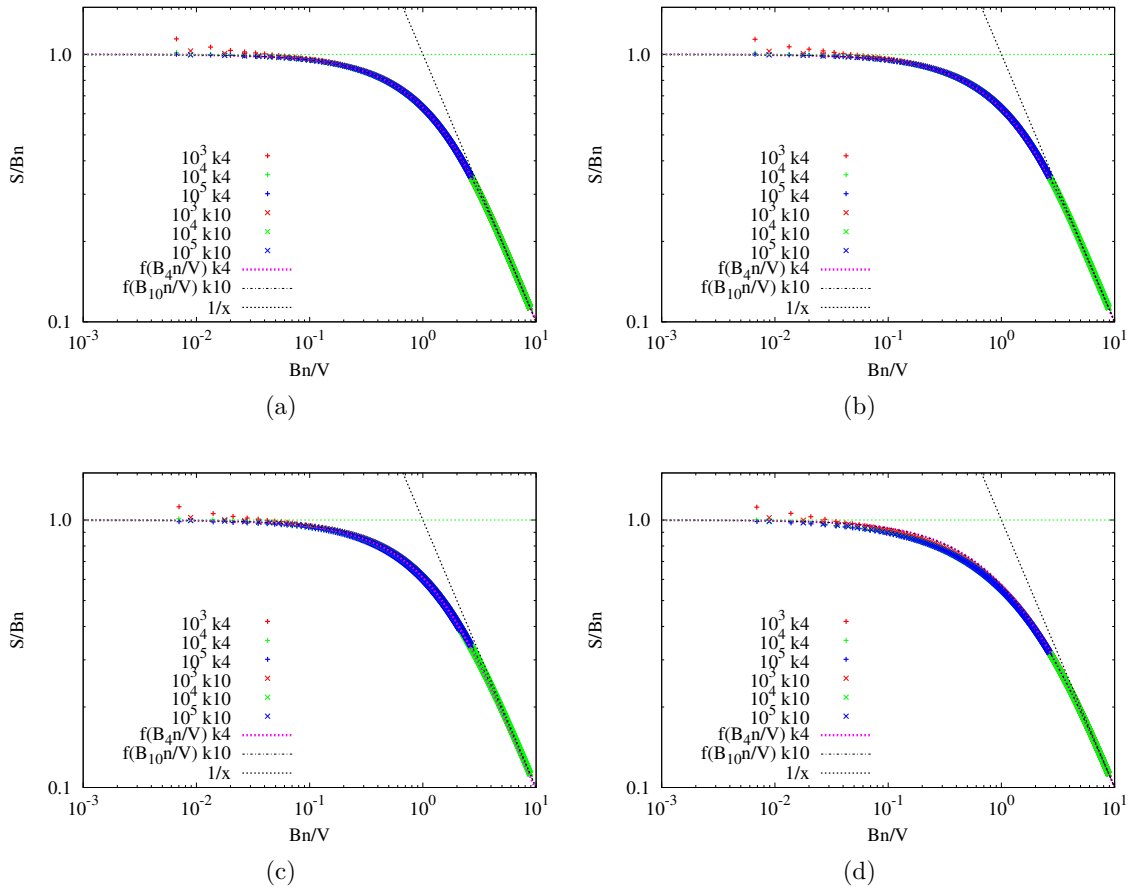


Figure 4.5: Finite-size scaling of number of distinct sites visited, showing $y = S(n)/(Bn)$ versus $x = Bn/V$. Data from direct simulations (symbols), with B predicted from the cavity equations, are shown for graphs of sizes $V = 10^3, 10^4, 10^5$ and average degrees $\langle k \rangle = 4, 10$. The graph topologies are: (a) Regular; (b) RER; (c) ER; (d) SF. Very good collapse onto a master curve $y = f(x)$ is seen between the different average degrees and – in (a,b,c) – also different V. The initial plateau at $y = 1$ shows the agreement between direct simulations and cavity predictions. For larger x saturation sets in, with $f(x) \approx 1/x$ asyptotically (dotted black line).

4.3 The rare events statistics of random walks on networks

In this section we focus our attention to study properties of rare events associated to unbiased random walks on networks. A rare event could have serious consequences on the networks situation, think for instance at the situation where in a communication network a router is overloaded. Considering cybersecurity, one could model the path made by viruses or spam emails as a random walk. A rare event could be represented by the case where the malicious attack succeeds to infect many sensible nodes, an event that could have catastrophic consequences. Studying rare events statistic associated to a random walk on a network could highlight some important aspects related to this dynamic process that are usually hindered by the average behavior of the walker. Our goal here is to analyze this topic by exploiting large deviation techniques. We want to characterize the statistics associated to rare event statistics of *path averages* or, equivalently, of time integrated variables and see how the results vary with graph topology. This approach is different from the one recently adopted to analyzed rare events for

biased random walks in complex networks in [97]. That work addressed rare fluctuations in single node occupancy for an ensemble of independent (biased) walkers in the stationary state of the system. Instead, rare event statistics of path averages has been looked at for instance in the context of kinetically constrained models of glassy relaxation [98]; relations to constrained ensembles of trajectories were explored in [99] for Glauber dynamics in the 1-d Ising chain. In these works they use large deviation formalism as a tool to explore dynamical phase transitions, a topic that here comes out as the outcome of the numerical study of our theory. Indeed we observe two types of dynamic phase transitions, whose details depend on the nature of the observable for which the rare events statistics is studied. In this work we focus the numerical studies on the degree of the nodes visited by the random walker, but the theoretical analysis will be addressed in the general case. An analytical approximation of the Legendre transform of the large deviation rate function will be given for graphs of large average connectivity. In the rest of the chapter we will discuss the main ideas presented in [62]. Our main focus is on the interplay between rare event statistics, the heterogeneity of the underlying system, and dynamic localization phenomena.

4.3.1 Large deviation theory

In this section we introduce the main ideas behind the large deviation formalism. In particular we will describe the main tools that we will use to address our problem in the next sections.
Large deviation theory [100, 101] studies the exponentially decay of probabilities of large fluctuations in random systems. In general in these systems the probability of a given event concentrates around the average. However it is important to obtain a finer characterization of the fluctuations around this average or the decay in probability towards the tail of the distribution. This is the main objective of large deviation theory.
The starting point of this theory is the formulation of the *large deviation principle* [100, 101]. This states that a probability distribution function $P(x_n)$ of a random variable x_n satisfies a large deviation principle if the following limit exists:

$$\lim_{n\to\infty} -\frac{1}{n}\ln P(x_n) = I(x_n) \tag{4.76}$$

where $I(x)$ is called the rate function. This principle can be interpreted by saying that the behavior for large n of $P(x_n)$ is dominated by a decaying exponential and that the rate of this decay is controlled by $I(x_n)$. Formally this can be written also as:

$$P(x_n) \propto e^{-nI(x_n)+o(n)} \tag{4.77}$$

where $o(n)$ means corrections sub-linear in n. This shows that the function $I(x)$ controls the dominant exponential term and the remaining $o(n)$ terms are neglected. Indeed $P(x_n)$ concentrates on the points corresponding to the zeros of $I(x_n)$. These points are the typical values in the large n limit. Therefore, concerning our problem of studying rare events, we are interested to determine this function. Unfortunately, proving that $P(x)$ follows the large deviation principle (4.76) or finding an analytical representation of $I(x)$ could be both hard tasks. Nonetheless one could overcome this problem by using the Gärtner-Ellis theorem [101] and the concept of limiting cumulant generating function $\lambda(k)$. The latter is a quantity defined as:

$$\lambda(k) = \lim_{n\to\infty} \frac{1}{n}\ln E[e^{nkx_n}] \tag{4.78}$$

where the symbols $E[x]$ means the expected value of x_n and k is a real parameter.
The theorem says [100, 101] that, if $\lambda(k)$ is differentiable, then x_n satisfies the large deviation principle (4.76) and the rate function $I(x_n)$ is determined as the Legendre transform [102, 103]

of $\lambda(k)$ as:

$$I(x) = \sup_{k \in R}\{kx - \lambda(k)\} \tag{4.79}$$

where sup is the supremum. This in particular shows that it is possible to calculate $I(x)$ without knowing explicitly $P(x)$, it is sufficient to know the cumulant generating function $\lambda(k)$ and check that this is differentiable. This is indeed the strategy that we will adopt in the following.

In general one can calculate the Legendre transform (4.79) by using the property of $\lambda(k)$ of being differentiable [102, 103]. In this case one could define:

$$F(x,k) = kx - \lambda(k) \tag{4.80}$$

and take the derivative with respect to k forcing it to be zero in order to calculate the maximum:

$$\frac{\partial}{\partial k} F(x,k) = 0 \tag{4.81}$$

But the form of $F(x,k)$ implies that:

$$\lambda'(\tilde{k}) = x \tag{4.82}$$

In general there could be more than one $\tilde{k}$ such that (4.82) is valid. However, if $\lambda'(k)$ is continuos and monotonically increasing for increasing k and it goes as $\lambda'(k) \to -\infty$ for $k \to -\infty$ and $\lambda'(k) \to +\infty$ for $k \to +\infty$ then there is a unique $\tilde{k}$. This (or these if there are more than one) value $\tilde{k}$ of k is the one that corresponds to the supremum in (4.79) so that we can finally write:

$$I(x) = \tilde{k}x - \lambda(\tilde{k}) \tag{4.83}$$

We have seen that the knowledge of $\lambda(k)$ is sufficient to derive the rate function. One has to be careful though. Indeed in all our considerations we were assuming that $\lambda(k)$ was differentiable $\forall k \in R$. This is one of the main hypothesis needed in order to apply the Gärtner-Ellis theorem (4.79). However, even if this was not the case, for instance when $\lambda(k)$ has one (or more) non-differentiable point k^*, the theorem could still be applied to the subset of the domain of $\lambda(k)$ where this is differentiable. One could say that the Gärtner-Ellis theorem is applied piece-wise in this case.

The fact that $\lambda(k)$ is not differentiable in some points, is a signature that the rate function either is not strictly convex, i.e. admits linear parts, or is not convex [101]. These two cases cannot be distinguished and in fact $I(x)$ cannot be calculated as the Legendre transform of $\lambda(k)$. If one tries to do this, i.e. applies (4.79) in the non differentiable points of $\lambda(k)$, one only finds the *convex envelope* of $I(x)$ [101]. This is defined as the double Legendre-Fenchel transform of I. The problem is that the rate function coincides with its convex envelope only if it is convex. If $I(x)$ is not convex than it can have more than one minimum and rare events are indeed rarer than what predicted by the convex enveloped calculated from $\lambda(k)$. If instead $I(x)$ is convex but not strictly, i.e. admits a linear part, than the exponential decay of $P(x)$ is slower than the strictly convex case where $I(x)$ has the shape of a parabola.

4.3.2 Large deviation theory adopted to rare events on networks

The theory of large deviations introduced above could be adapted to describe the rare events statistics associated to functions defined on vertices $i \in \mathcal{V}$ along paths made by an unbiased random walker on the network. We consider functions that take the form $f(\epsilon_i)$, where ϵ_i are quenched random variables associated with the nodes of the graph. These variables could either be stochastic or deterministic. In the first case one could take them to be either independent or correlated to the node degree k_i. In the deterministic case the non-trivial choice is to take them as functions of the degree $f(k_i)$. In this work we focus on the latter, reserving the stochastic case for future works. Nevertheless we will describe a general model valid in both cases and we will point out when the results will be restricted to the deterministic case.

Consider a path $\bar{i}_l = (i_0, \ldots, i_l)$ of length l of a random walk over a network, with $\imath_n$, $n = 0, \ldots, l$ denoting the visited nodes $i_n \in \mathcal{V}$ at each step n. We are interested to study the statistics of empirical averages:

$$\hat{\phi}_l = \frac{1}{l} \sum_{n=1}^{l} f(\epsilon_{i_n}) \tag{4.84}$$

where $\{i_n\}$ are the nodes of the path $\bar{i}_l$. The mean of $\hat{\phi}(\epsilon_l)$ is defined as:

$$\begin{aligned} \bar{\phi}_l &= \frac{1}{l} \sum_{\bar{i}_l} P(\bar{i}_l) \sum_{n=1}^{l} f(\epsilon_{i_n}) && (4.85) \\ &= \frac{1}{l} \left\langle \sum_{n=1}^{l} f(\epsilon_{i_n}) \right\rangle && (4.86) \end{aligned}$$

where $P(\bar{i}_l)$ is the probability of having a path $\bar{i}_l$ given a walk with unbiased transition probability as in (4.4) and an arbitrary distribution of initial conditions. The mean $\bar{\phi}_l$ could be computed using the moment generating function $Z_l(s)$ of the form:

$$\begin{aligned} Z_l(s) &= E\left[e^{s\sum_{n=0}^{l} f(\epsilon_{i_n})}\right] && (4.87) \\ &= \sum_{\bar{i}_l} P(\bar{i}_l) e^{s\sum_{n=0}^{l} f(\epsilon_{i_n})} && (4.88) \end{aligned}$$

and then noting that the mean (4.85) could be written as:

$$\bar{\phi}_l = \frac{d}{ds} \frac{1}{l} Z_l(s)|_{s=0} \tag{4.89}$$

We expect that for large l the empirical average (4.84) is highly peaked around the mean (4.85). However focusing our attention on the mean may hinder some interesting behavior that could be seen only when considering fluctuations around this mean. Therefore in the following we will explore the behavior of $\hat{\phi}_l$ when this deviates from the mean. Formally, we are interested in quantifying the probability of rare events, i.e. in quantifying $P(\phi)d\phi = Prob(\hat{\phi} \in [\phi, \phi + d\phi))$. To achieve this we will deploy the large deviation formalism introduced above. In what follows, for simplicity of notation, we will use the variable ϕ instead of $\hat{\phi}_l$ taken in the limit $l \to \infty$.
We start by defining the cumulant generating function:

$$\begin{aligned} \psi_l(s) &= \frac{1}{l} \ln E\left[e^{s\sum_{n=0}^{l} f(\epsilon_{i_n})}\right] && (4.90) \\ &= \frac{1}{l} \ln Z_l(s) && (4.91) \end{aligned}$$

and assume that this quantity exists and is differentiable.
In this case one can define the equivalent of the limiting cumulant generating function (4.78) specific to our problem:

$$\psi(s) = \lim_{l \to \infty} \psi_l(s) \tag{4.92}$$

Therefore one could use the Gärtner-Ellis theorem (4.79) to imply the validity of the large deviation principle (4.77). This means that $P(\phi)$ has an asymptotic behavior (valid for large l) of the form:

$$P(\phi) \sim \exp\{-lI(\phi)\} \tag{4.93}$$

where the exponential decay is controlled by the rate function $I(\phi)$.
Moreover this quantity can be determined as the Legendre transform of $\psi(s)$:

$$I(\phi) = \sup_{s \in R}\{s\phi - \psi(s)\} \tag{4.94}$$

and we can conclude that $P(\phi)$ concentrates around the zeros of $I(\phi)$, i.e. the typical values

of ϕ. In what follows we are interested to find how ϕ deviates from these typical values. We could calculate $I(\phi)$ by taking the derivative of (4.94) with respect to s so that we recover (4.82),(4.83) and eventually we could write:

$$\psi'(s^*) = f\phi \tag{4.95}$$

$$I(\phi) = s^*\phi - \psi(s^*) \tag{4.96}$$

or equivalently we could write the parametric expression:

$$I(\psi') = s^*\psi' - \psi(s^*) \tag{4.97}$$

where s^* is the solution of (4.95). This last equation is particularly convenient in the case where finding an analytical expression for $I(\phi)$ is difficult. In this case one can nonetheless evaluate it numerically by first evaluating both $\psi(s)$ and $\psi'(s)$. Then one could plot I as a parametric function of ψ'.

4.3.3 Evaluation using the spectrum of a deformed transition matrix

Following the considerations made above, the whole problem is translated to the determination of the limiting cumulant generating function $\psi(s)$ and its Legendre transform, i.e. the rate function $I(\phi)$. Here we will tackle this problem by relating $\psi(s)$ to the eigenvalues of a deformed transition matrix $W(s)$ (or its symmetric counterpart $\tilde{W}(s)$) that results from a modification of the unbiased walk transition matrix W as in (4.4). The idea is that rare events under W are mapped into typical events under the deformed dynamics regulated by $W(s)$.
In what follows it is convenient to adopt the convention of denoting by W_{ij} as the transition probability of jumping from j to i, instead of the contrary as done in section 4.2.
First of all notice that the path probability $P(\bar{i}_l)$ can be written in terms of W as:

$$P(\bar{i}_l) = \left[\prod_{n=0}^{l-1} W_{i_{n+1}i_n}\right] p_0(i_0) \tag{4.98}$$

In this way one could express compactly the moment generating function $Z_l(s)$ defined in (4.87) as:

$$Z_l(s) = \sum_{\bar{i}_l} \left[\prod_{n=1}^{l} e^{sf(\epsilon_{i_n})} W_{i_n i_{n-1}}\right] p_0(i_0) \tag{4.99}$$

$$= \sum_{i_n, i_0} [W^l(s)]_{i_l, i_0} p_0(i_0) \tag{4.100}$$

where we have introduced a (first) deformed transition matrix $W(s)$ with entries:

$$W_{ij}(s) = e^{sf(\epsilon_i)} W_{ij} \tag{4.101}$$

Notice that this matrix is in general not symmetric.
Now we consider the large l limit of $Z_l(s)$ and for this purpose it is convenient to evaluate it in terms of the spectral decomposition of $W(s)$. Denoting by λ_α the eigenvalues of $W(s)$ and by $\mathbf{v}_\alpha$ and $\mathbf{w}_\alpha$ the corresponding right and left eigenvector (respectively), one has:

$$Z_l(s) = \sum_\alpha \lambda_\alpha^l (\mathbf{1}, \mathbf{v}_\alpha)(\mathbf{w}_\alpha, \mathbf{p}_0) \tag{4.102}$$

where $\mathbf{1} = (1, \ldots, 1)$ and $\mathbf{p}_0 = (p_0(i_1), \ldots, p_0(i_V))$. We follow the convention in [62] to denote inner product using brackets $(\cdot, \cdot)$. Notice that the eigenvalues and eigenvectors depend implicitly on s, even though we dropped this dependence when using the notation λ_α. Indeed one has $\lambda_\alpha = \lambda_\alpha(s)$.
Taking the ln of this expression we obtain the cumulant generating function $\psi_l(s)$ as defined in

(4.91) as a function of the spectrum of $W(s)$:

$$\psi_l(s) = \ln \lambda_1 + \frac{1}{l}\left[(\mathbf{1}, \mathbf{v}_1)(\mathbf{w}_1, \mathbf{p}_0) + \sum_{\alpha \neq 1} \left(\frac{\lambda_\alpha}{\lambda_1}\right)^l (\mathbf{1}, \mathbf{v}_\alpha)(\mathbf{w}_\alpha, \mathbf{p}_0)\right] \tag{4.103}$$

where we separated on purpose the leading eigenvalue λ_1 from the rest of the spectrum $\lambda_1 > |\lambda_2| \geq \cdots \geq \lambda_N$. Here this series of inequalitis is a consequence of the Perron-Frobenius theorem [67].

This spectral decomposition allows to determine the large l limit as a function of only the leading eigenvalue λ_1. In fact the contributions of all the other λ_α (with $\alpha \neq 1$) in the determination of $\psi(s)$ are negligible when $l \to \infty$. This can be seen by taking in (4.103) first $l \to \infty$, because of the term $\left(\frac{\lambda_\alpha}{\lambda_1}\right)^l \to 0$, and only after we can consider the thermodynamic limit $V \to \infty$. This order in taking the limits is important to insure that the product $(\mathbf{1}, \mathbf{v}_\alpha)(\mathbf{w}_\alpha, \mathbf{p}_0)$ is a finite quantity, therefore allowing to discard the sum inside (4.103).

Bearing in mind that $\lambda_1 = \lambda_1(s)$, eventually we can write:

$$\psi(s) = \ln \lambda_1(s) \tag{4.104}$$

and thus:

$$I(\phi) = \sup_s \{s\phi - \ln \lambda_1(s)\} \tag{4.105}$$

and everything resolves to the evaluation of the leading eigenvector of the matrix $W(s)$ defined in (4.101). One could equivalently write parametric expressions by taking the derivative with respect to s:

$$\psi'(s^*) = \frac{\lambda_1'(s^*)}{\lambda_1(s^*)} = \phi \tag{4.106}$$

$$I(\psi') = s^*\psi' - \psi(s^*) \tag{4.107}$$

These last two equations are particularly convenient when the numerical evaluation of I is the only viable way to determine the rate function.

The case $s = 0$ is trivial. Indeed the column-stochasticity, i.e. the property $\sum_i W_{ij}(s) = 1$, yields a left row eigenvector $\mathbf{w}_1 = (1, \ldots, 1)$, corresponding to the maximum eigenvalue $\lambda_1 = 1$. Whereas the corresponding right column eigenvector $\mathbf{v}_1$ has entries $v_{1i} \propto k_i$.

For the case $s \neq 0$ such closed form expressions are in general not known.

As we already mentioned, this matrix is not symmetric. Therefore it could be convenient to symmetrize it in order to compute λ_1 efficiently. This is obtained by defining another deformed matrix $\tilde{W}(s)$ applying a similarity transform to $W(s)$ as:

$$\tilde{W}(s) = D^{-1/2} W(s) D^{1/2} \tag{4.108}$$

The matrix $\tilde{W}_{ij}(s)$ has entries:

$$\tilde{W}_{ij}(s) = e^{\frac{s}{2}f(\epsilon_i)} \frac{a_{ij}}{\sqrt{k_i k_j}} e^{\frac{s}{2}f(\epsilon_j)} \tag{4.109}$$

This matrix has the same spectrum of $W(s)$. In fact, denoting by $\mathbf{u}_\alpha$ an eigenvector of $\tilde{W}$, we have the eigenvalue equation:

$$\tilde{W}(s)\mathbf{u}_\alpha = \lambda_\alpha \mathbf{u}_\alpha \tag{4.110}$$

$$D^{-1/2} W(s) D^{1/2} \mathbf{u}_\alpha = \lambda_\alpha \mathbf{u}_\alpha \tag{4.111}$$

and multiplying on the left by $D^{1/2}$ both the sides of the equation we obtain:

$$W(s) D^{1/2} \mathbf{u}_\alpha = \lambda_\alpha D^{1/2} \mathbf{u}_\alpha \tag{4.112}$$

which is the eigenvalue equation of $W(s)$ for the eigenvalue λ_α and corresponding eigenvector $D^{1/2}\mathbf{u}_\alpha$. The same reasoning can be made using the left eigenvector $\mathbf{w}_\alpha$ of $W(s)$. Thus $\tilde{W}(s)$ and $W(s)$ have the same spectrum, and in particular the same leading λ_1. The eigenvectors

are not the same instead, nonetheless they can be related by:

$$\mathbf{v}_\alpha = D^{1/2}\mathbf{u}_\alpha \tag{4.113}$$
$$\mathbf{w}_\alpha = \mathbf{u}_\alpha^T D^{-1/2} \tag{4.114}$$

where $\mathbf{u}_\alpha^T$ denotes the row vector corespondent to $\mathbf{u}_\alpha$.

Ideally, for the non-zero s, one would like to find simple analytical results as for the $s = 0$ case. Unfortunately, the graph heterogeneity makes this a difficult task to be addressed. Moreover the strategy of performing a direct matrix diagonalization of either $W(s)$ or $\tilde{W}(s)$ could be daunting for large N. This is valid even if one considers alternative methods that calculates only the first eigenvalue as the Lanczos algorithm [104]. Hence we are interested to find a fast viable approximation.
We start by noting that the form of $W(s)$ in (4.101) suggests us the following ansatz for the left eigenvector $\mathbf{w}_1$:

$$\lambda_1 w_j = \frac{1}{k_j}\sum_{i\in\partial j} w_i e^{sf(\epsilon_i)} \tag{4.115}$$

This defines a system of V equations, i.e. one equation $\forall i \in \mathcal{V}$. These equations can be solved iteratively using Lagrange multiplier formalism combined with the cavity method [105], though we found that using a Lanczos algorithm [104] to determine the largest eigenvalue of $\tilde{W}(s)$ is usually more efficient than a simple forward iteration of (4.115).
In general though, as already mentioned before, one could numerically determine λ_1 and from this calculate $\psi(s)$ using (4.104). Eventually one could determine the rate function using the parametric expression (4.107) but first one has to calculate $\psi'(s)$. This could be done numerically but fortunately there exists an analytical exact formula to characterize this derivative starting from the eigenvalue equation for $\tilde{W}(s)$ and using (4.106). Here we can use $\tilde{W}(s)$ because the spectrum is the same as the one of $W(s)$ and all we need to calculate $\psi'(s)$ is λ_1.
In fact, considering the first eigenvalue λ_1 and the corresponding eigenvector $\mathbf{v}$, the eigenvalue equation $\tilde{W}(s)\mathbf{v} = \lambda_1\mathbf{v}$ yields:

$$\lambda_1 = (\mathbf{v}, \tilde{W}(s)\mathbf{v}) \tag{4.116}$$
$$= \sum_i v_i \sum_j \tilde{W}_{ij}(s) v_j \tag{4.117}$$

Note that we also have:

$$\tilde{W}'_{ij}(s) = \left[\frac{f(\epsilon_i)}{2} + \frac{f(\epsilon_j)}{2}\right]\tilde{W}_{ij}(s) \tag{4.118}$$

where the last equation is symmetric in ij.
Now we consider (4.106). Taking the first order derivative with respect to s of (4.117) and combining it with (4.118), eventually we obtain:

$$\psi'(s) = \sum_i v_i^2 f(\epsilon_i) \tag{4.119}$$

This formula could be used to evaluate numerically $\psi'(s)$ by using the components v_i of the first eigenvector, thus avoiding to perform the numerical derivation of $\psi(s)$.

4.3.4 Degree-based approximation

One could attempt to solve analytically the system of equations (4.115) by considering a degree-based approximation, a method that has been often used to study epidemic processes on networks [106, 107]. In this case one assumes that eigenvalue's entries w_i only depend on the

degree k_i of node i:

$$w_i = w(k_i) \tag{4.120}$$

thus nodes with same degree will correspond to equal eigenvector entries. This could be rephrased by saying that all nodes with a given degree are considered statistically equivalent.
This approximation is expected to be valid when the average degree $c = \langle k \rangle = \sum_{i \in \mathcal{V}} k_i / V$ is large enough and the degree distribution is not too heterogenous.
If this is the case we could write the eigenvalue equation (4.115) by using the law of large numbers as:

$$\lambda_1 \, w(k) = \sum_{k'} P(k'|k) \, w(k') \, e^{sf(k')} \tag{4.121}$$

where $P(k'|k)$ is the probability for a node of degree k to have neighboring node of degree k'. In an Erős-Renyí graph and more generally in any configuration model ensemble or in a graph where two-point correlations are not present we have that $P(k'|k)$ is independent on k. In fact one has [106, 107]:

$$P(k'|k) = P(k') \frac{k'}{\langle k \rangle} \tag{4.122}$$

In this case the right-hand side of (4.121) does not depend on k and thus the left-hand side neither. This implies that the $w(k)$ are in fact independent on k.
The eigenvalue equation (4.121) then simplifies to:

$$\lambda_1 w = w \sum_{k'} P(k') \frac{k'}{\langle k \rangle} \, e^{sf(k')} \tag{4.123}$$

$$\tag{4.124}$$

and one can drop the w so to eventually obtain:

$$\lambda_1 = \left\langle \frac{k}{\langle k \rangle} e^{sf(k)} \right\rangle \tag{4.125}$$

where the average is over the degree distribution $P(k)$. From this expressionone could calculate $\psi = \ln \lambda_1$ as in (4.104).
We found that this approximation yields excellent results for large mean connectivities $c = \langle k \rangle$ on ER graphs and more generally for configuration models without low degree nodes. This can be seen in figure 4.6 where we plot a comparison with numerical simulations for ER with $c = 30$ and for $f(k_i) = k_i/c$. Numerical simulations are used to determine $\langle \ln \lambda_1 \rangle$, where the average is over 1000 instances of these types of ER graphs.

4.3.5 Eigenvector localization

The exponential form (4.101) of the deformed transition matrix suggests that for large values of $|s|$ the random walk may be highly localized around nodes where $sf(\epsilon_i)$ is very large. This feature can be quantitatively investigated be defining the inverse participation ratio of the eigenvector corresponding to the largest eigenvalue λ_1 of $W(s)$. Denoting by v_i its components, this is defined as [108]:

$$\mathrm{IPR}[\mathbf{v}] = \frac{\sum_i v_i^4}{\left[\sum_i v_i^2 \right]^2} \tag{4.126}$$

One expects $\mathrm{IPR}[\mathbf{v}] \sim N^{-1}$ for a delocalized vector, whereas $\mathrm{IPR}[\mathbf{v}] = \mathcal{O}(1)$, if $\mathbf{v}$ is localized. In fact a delocalized N-dimensional vector with norm $||\mathbf{v}|| = 1$ has homogenous entries, each one of the order $v_i \sim 1/\sqrt{N}$. This gives a denominator ~ 1 in (4.126) and a numerator $\sim N^{-1}$. On the contrary, a localized $\mathbf{v}$ has most of the entries zero or close to zero and only a few

$v_i = \mathcal{O}(1)$ contributing the most to the norm.

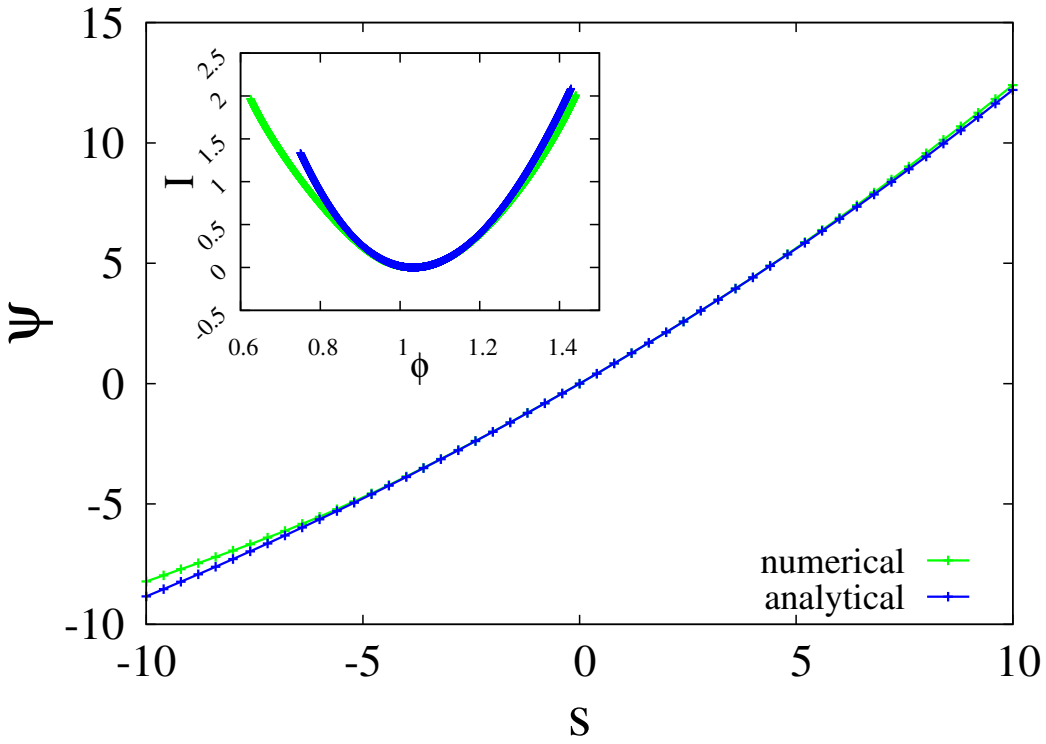


Figure 4.6: Degree-based approximation. We plot $\psi = \ln \lambda_1$ as a function of s for ER networks with $c = 30$ and $f(k_i) = k_i/c$ the renormalized degree. The green curve is the result of the large-degree approximation (4.125) for this graph ensemble. In blue we plot results from numerical simulations over 1000 realizations of the graph topology. INSET: the rate function $I(\phi)$ resulting from the numerical and analytical degree-based approximation.

4.3.6 Numerical results

We performed numerical simulations to evaluate λ_1 and the IPR[$\mathbf{v}$] for several instances of random network of type Erdős-Rényi (ER) [33] and scale-free (SF) under the preferential attachment scheme [83]. See appendix 3 for more details about the graph construction. We used various average connectivities c and system sizes V.
We looked at two different definitions for $f(k_{i_t})$: a normalized degree $f(k_i) = k_i/c$ and a binary function $f(k_i) = 1, (0)$ if $k_i > c$, $(k_i \leq c)$, respectively.
We restricted ourselves in the simulations to connected graphs, in order to prevent isolated nodes or small disconnected clusters (e.g. dimers) dominating λ_1 and thus the IPR for negative s, as these would represent trivial instances of rare events, where a walker starts and is thus stuck on a small disconnected component of the graph. Therefore, after having generated a graph, we kept only the giant component and calculated λ_1 and $\mathbf{v}$ on it.
The first type of results concerns λ_1. To evaluate this quantity one could for instance use Lanczos algorithm [104] that allows to evaluate efficiently only the the first n eigenvalues of interest, if the system size is small enough($V < 4000$). Otherwise we found that solving the eigenvalue equation (4.121) iteratively as proposed in [105] is faster for bigger system sizes.
Once we have λ_1 this can be used to calculate the relevant quantities $\psi(s)$, $\psi'(s)$ and $I(\phi)$ using

(4.104), (4.119) (for this we also need the eigenvector components v_i) and (4.107) respectively.

In figure 4.7 we plot two examples of rate functions for the two choices of $f(k_i)$. We plot the I as parametric functions of ϕ as in (4.105). We could notice that for $f(k_i) = k_i/c$ the curve seems to show a linear behavior with slope a on the left of the minimum for increasing system sizes. This is a signal that $\psi(s)$ may not be differentiable in $s^* = a$ as N increases. Therefore we analysed the behavior of $\psi(s)$ for the largest system size ($N = 6400$) and indeed noticed the existence of this non-differentiable point which we estimated to be $s^* = 0.302 \pm 0.021$ for $c = 6$ and $s^* = -0.061 \pm 0.008$. In figure 4.8 we plot the linear fit on the left side of the rate function with slope s^* and in the inset we display $\psi(s)$ for the two different average connectivities. We can notice that in s^* the tangents take two different values approaching this point from the left and from the right. This behavior seems to smooth out with increasing c, and indeed this is confirmed by the inset of figure 4.6 where for high average connectivity $c = 30$ shows a rate function without linear parts. On the contrary for the smallest connectivity $c = 3$ the linear part is the sharpest and for this it is more evident the presence of the complex hull of the otherwise parabolic behavior, as can be seen in figure 4.6 (bottom).

Also the right branch of the rate function seems to have a slow increasing slope as for the linear left branch. However the result of a quadratic fit on the right side of the curve seems to better describe its behavior instead of a linear one. For the binary case the rate function shows a regular behavior too for increasing N and in fact $\psi(s)$ (not plotted) is always differentiable in this case.

The presence of a non-differentiable point of $\psi(s)$ implies that one cannot apply the Gärtner-Ellis theorem (4.79) in the point $s = s^*$. The linear part of the curve of $I(\phi)$ represents its convex envelope, i.e. the Legendre transform of $\psi(s)$. This coincides with the actual rate function $I(\phi)$ only if the rate function is convex (or strictly convex). If this was the case then the exponential decay of $P(\phi)$ would be slower that the strictly convex case where $I(\phi)$ is a parabola. This allows to differentiate small deviations (I is parabolic) and large deviations (I has a linear part) from the typical value corresponding to the minimum of I. If instead $I(\phi)$ is not convex, than it admits more than one minimum and this translates into $P(\phi)$ having rare events which are rarer than what is predicted by the convex envelope calculated from $\psi(s)$.

The presence of this linear part, thus of a non-differentiable point of $\psi(s)$, is a signature of a dynamic phase transition concerning the largest eigenvalue of $W(s)$. In fact $\lambda_1(s)$ cannot display a non analyticity unless it is degenerate. Unless you have a cycle, in a finite system the largest eigenvalue of $W(s)$ will be non degenerate, but as the system size grows, a degeneracy may develop; the transition becoming sharp as $N \to \infty$.

To better explore this conjecture we monitored the second largest eigenvalue $\lambda_2(s)$ and indeed saw an avoided *level crossing* near the estimated s^*. By this we mean that the two eigenvalues start far apart from each other and then λ_2 approaches more and more λ_1 as $s \to s^{*-}$ without ever reaching it. Then they start to get apart again when $s > s^*$, on the right side of the transition point.

A better signature of this level crossing and degeneracy is obtained by monitoring a *relaxation time* so defined:

$$\tau = -\frac{1}{\log\left(\frac{\lambda_2}{\lambda_1}\right)} \tag{4.127}$$

From this definition we see that a divergence arises for $\lambda_2 \to \lambda_1$, i.e. for a level crossing; whereas an increasing gap between the first two eigenvalues would give $\tau \to 0$. In figure 4.9 we can notice a divergence close to s^* for the case of ER graphs of average connectivity $c = 6$; this seems to keep diverging more sharply for increasing N. Another smaller divergence seems to develop for a positive value of s, a signal of a linear branch of the rate function on the right

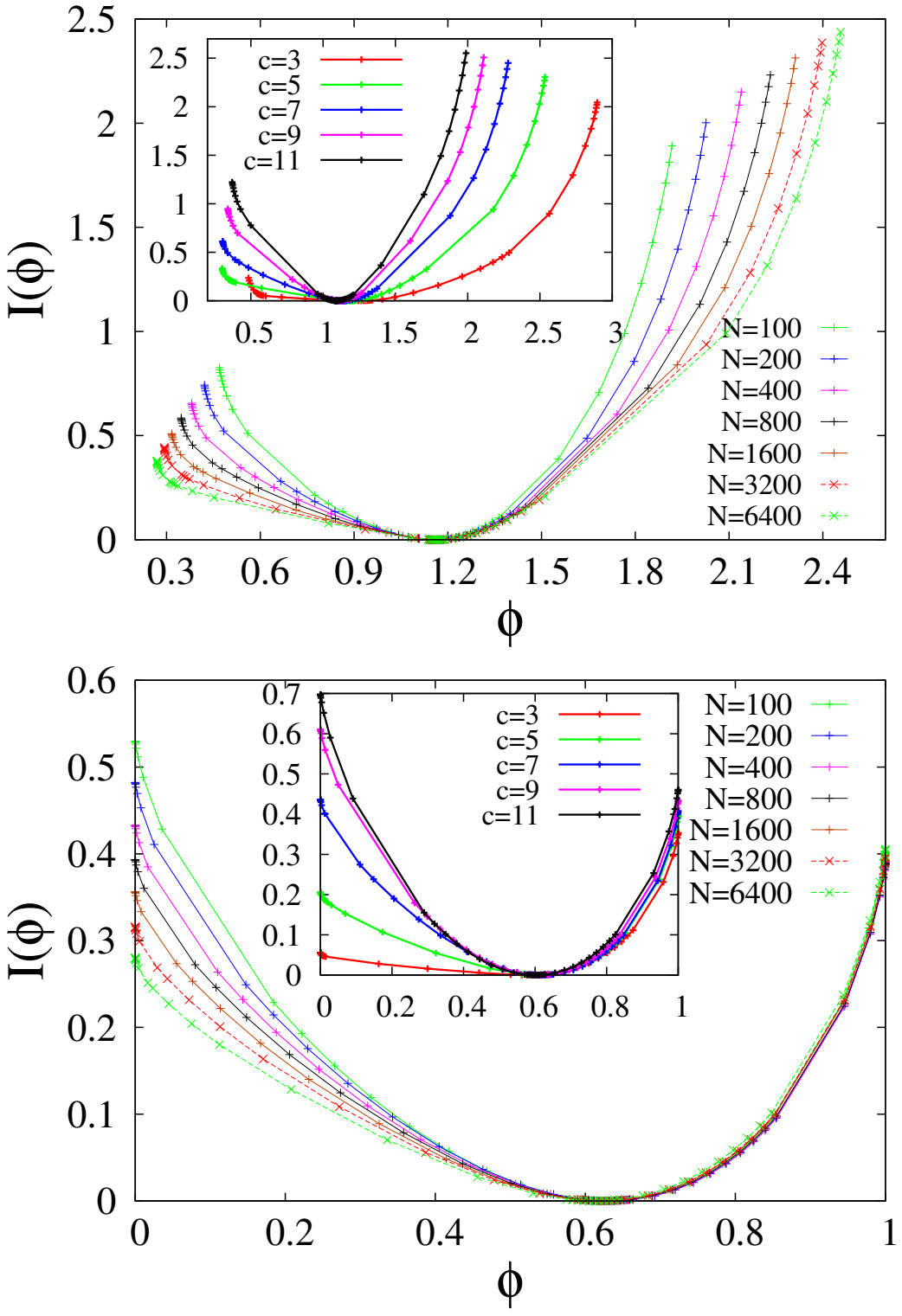


Figure 4.7: Rate function $I(\phi)$ as a function of ϕ. We use ER graphs with $c = 6$, $f(k_{i_t})$ the renormalized degree (top) and the binary function (bottom). Results are averages over 1000 realizations of each graph topology. Different colors represent different system sizes. Insets: the same curves for a fixed size $N = 3200$ and different connectivities.

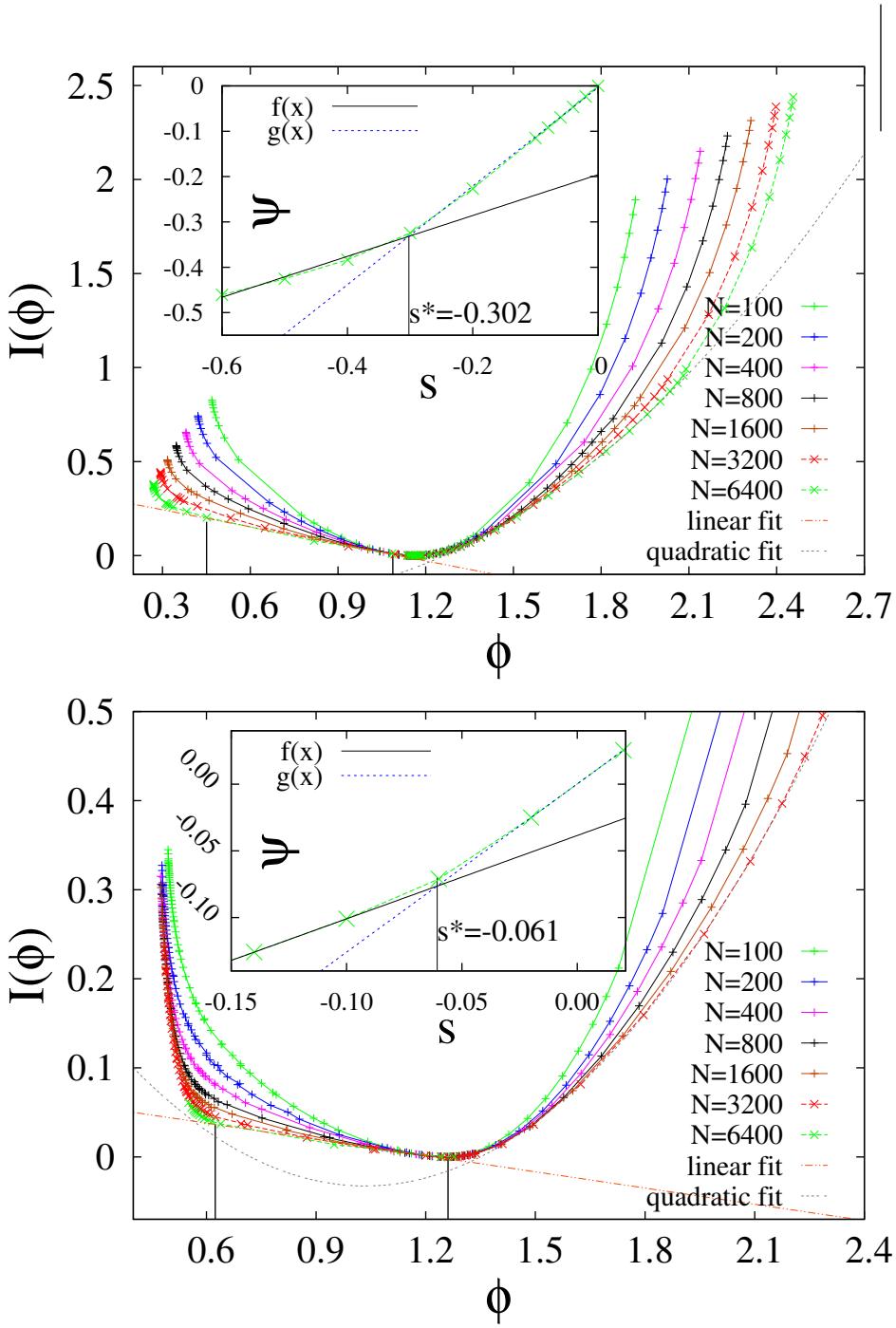


Figure 4.8: Rate function linear behavior. $I(\psi')$ exhibits a linear behavior for $\psi' \in (a_1, a_2)$, these corresponds to the vertical lines. This results are for ER graphs of $c = 6$ and different system sizes. From a linear fit we estimated $a_1 = 0.45 \pm 0.04$ and $a_2 = 1.09 \pm 0.01$, these are plotted as vertical lines. The linear curve *linear fit* with slope $s^* = -0.302 \pm 0.021$ is superimposed to show the agreement with the estimated non-differentiable point of $\psi'(s)$. Inset: $\psi(s)$ is plotted around the non-differentiable point s^*. The two tangents, denoted with $f(x)$ and $g(x)$, are displayed. These are the results of linear fits. Their resulting slopes a_1 and a_2 correspond to the extremes of the interval where the rate function is linear.

side of the curve. However, as we have already seen if figure 4.8, indeed there seems to be no linear envelope, and a slowly quadratic increase better fits the data.

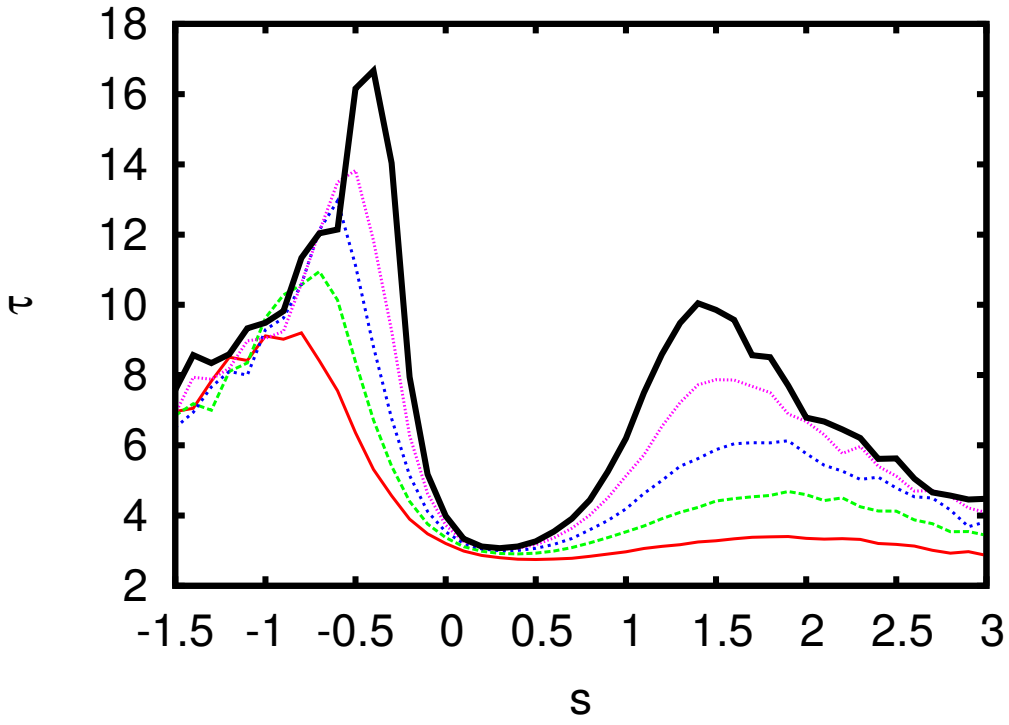


Figure 4.9: Relaxation time. The quantity (4.127) is plotted as a function of s for ER graphs of increasing system sizes (from the bottom to the top curves) and $\langle k \rangle = 6$. A divergence peak seems to develop around the estimated transition point $s^* = -0.301$ signaling the presence of a non analytic point of $\psi'(s)$. A smaller peak for a positive value of s seems seems to develop as well, however the corresponding behavior of the rate function around that point seems to be quadratic.

We can conclude that all these findings unveil the presence a first type of dynamic phase transition related to a *switching between modes* phenomenon, as the deformation parameter s used to select rare events is varied. Namely, the level crossing of the first two largest eigenvalues, implies that the *mode* which realizes the typical rare event changes at the given s to another mode, and the original one is no longer the dominant rare event. Thus it can be interpreted as a dynamic phase transition in the rare events ensemble.

The second type of result concerns the IPR (4.126), the quantity playing the role of a signature of eigenvector localization. In figure 4.10 we give results of numerical evaluations of this quantity for the two different expressions for $f(k_i)$. In the case of $f(k_i) = k_i$ (top) we see high localization for big values of $|s|$. This shows that in this interval the walk concentrates on very high (or very low) degree nodes. On the contrary for values of $s \in [-0.1, 0.7]$ the IPR is almost zero, thus showing delocalization. The case of the binary $f(k_i)$ (bottom) also shows both regimes but in this case we have that the localization happens only for negative values of s and small c, whereas for positive values the IPR goes to zero. Localization persists at negative s up to $c \sim 10$. This is related to the binary nature of the function. The random walk at large $|s|$ can be understood as a random walk on an effective network where the low (or high) degree nodes have been replaced by an absorbing state (this effectively corresponds to the random walker falling into a hole). The extended range of localization at negative s regime can be understood as a consequence of this absorbing state and the low degree of the "allowed" nodes: the effective graph is more strongly fragmented than mere random removal of nodes would suggest, and the

random walk is attracted to those connected components of the effective network where the probability of falling into a hole is lowest. On low-c networks, this limits the choices to small chains at the periphery of the network and thus creates significant localization. Thus the system exhibits a *localization transition*, indicating that rare large fluctuations of path averages are typically realized by trajectories that remain localized on small subsets of the network.

We show the transition between the two phases in the insets of the two figures. There we take logarithmic scale and plot the IPR as a function of $1/N$. We can notice that in the localized regime the IPR tends to a constant value with increasing system size, i.e. decreasing $1/N$. Whereas in the delocalized regime we have a linear behavior that persists at all N.
The existence of a localized phase in either the choices of $f(k_i)$ highlight the presence of a *dynamic cluster*, a set of consecutive nodes that have very high (or low) degree. We call this concept dynamic to not confuse it with the static clusters usually characterizing the topology. Here these structures arise only when performing a random walk under the deformed dynamics (4.101) and depend on the shape of $f(k_i)$. The static clusters instead are sets of nodes characterized by given topological properties, such as their mean degree with respect with the overall graph's one. They are extensively used in the studies of community detection on networks [109, 110] and their existence is hence independent on the random walk. These dynamic clusters arise as a consequence of the transition matrix deformation and are detected by studying the leading eigenvector.

This set of results follows from a preliminary work on rare events associated to random walks on networks that opens new questions that needs to be understood.
First of all here we focused our attention of deterministic functions $f(k_i)$ of the node degree. It may be an interesting generalization to consider stochastic functions instead. We indeed have obtained some preliminary results for binary functions taking values independently on k_i, however these results are not reported here because they require a deeper understanding and a thorough interpretation.
Secondly, we considered functions taking values on nodes. The same formalism could be adapted to a dual case where variables are instead on edges. This could be of interest for instance when studying routing problems on networks subject to capacity constraints or having costs represented by weights. A rare event in this case could be associated to a walk performed on overcrowded (or completely empty) edges. The arguments made above seem to adapt to this case but a deeper analysis has to be formulated in order to state more rigorous results. We leave all these open issues for further studies.

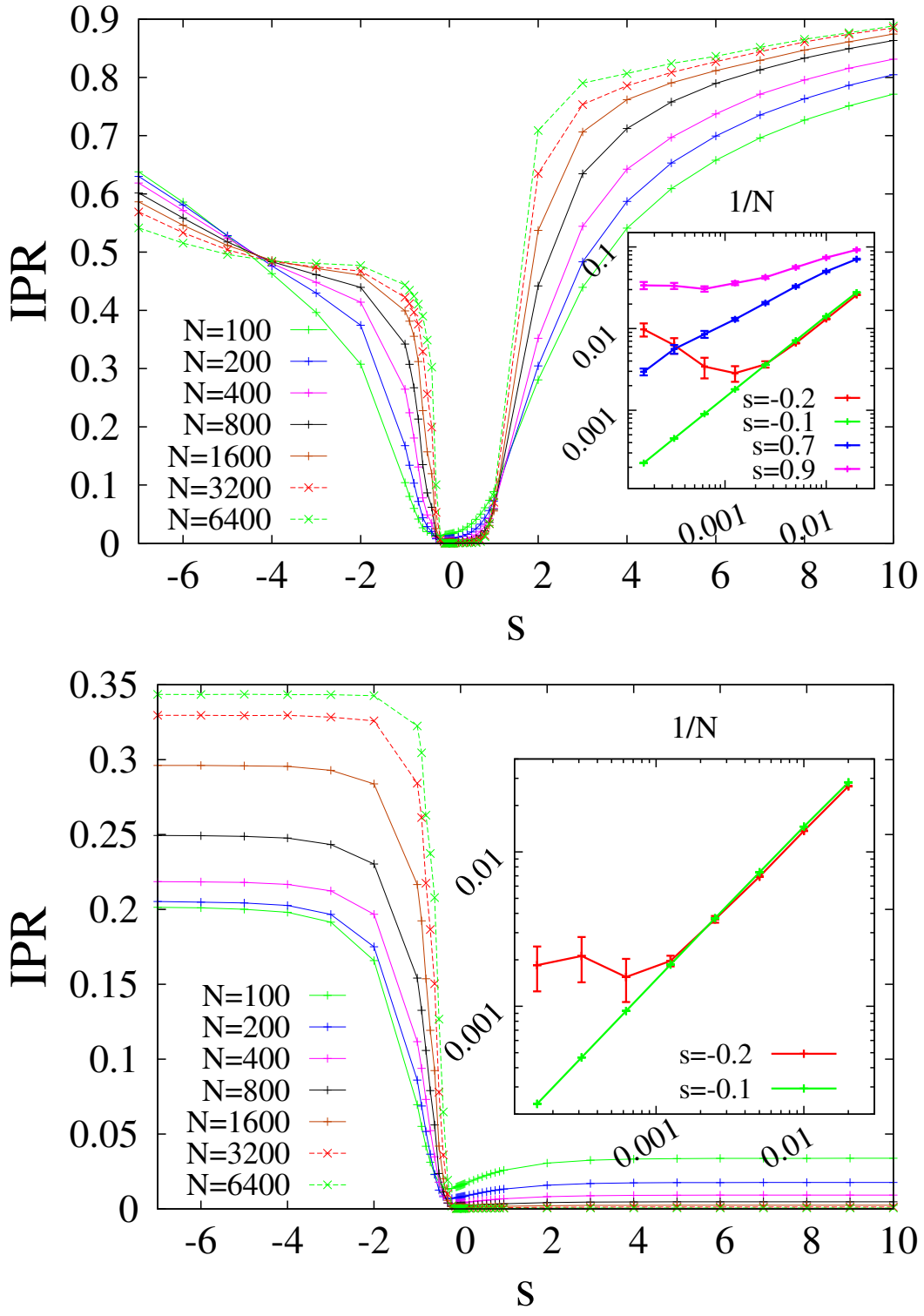


Figure 4.10: IPR[**v**] as a function of s for ER with $c = 6$, $f(k_{i_t})$ the renormalized degree (top) and the binary function (bottom). Results are averages over 1000 realizations of each graph topology. Different colors represent different system sizes. In the insets, $\frac{1}{N}$ scaling in the non-localized and localized regimes, respectively linear and approximatively constant

Chapter 5

Non-equilibrium Statistical Mechanics of the Heat Bath for Two Brownian Particles

In this chapter we will described a model whose development has begun before the official start of the PhD but that has been finally completed only during the first year of the thesis. Therefore it is a Chapter independent form the others and is based on the work published in [111]. Here we outline the main ideas and focus on the novel outcomes resulting from this work, leaving the details of the computation to the reference [111].
The model is composed of two mesoscopic brownian particles that interact through a potential $U_0(X_1, X_2)$, where X_1 and X_2 are their respective positions. At the same they are surrounded by a thermal bath. The latter is represented as an infinite number of microscopic particles that interact with both of the brownian ones, leading to a frictional memory force that causes the dynamics to be non-Markovian. It is an out-of-equilibrium system that we solved extending Zwanzig model [112]. Zwanzig presented an analytically solvable microscopic model of a single Brownian particle in an arbitrary potential. The heat bath around a single heavy particle has been modeled by many light mass particles each of which is coupled to the heavy particle through a Hookean spring. Despite the simplicity of considering only one brownian particle, that model has revealed to be very instructive in providing a concrete picture of the microscopic origin of the generalized Langevin equation. That inspired us to expect that solvable microscopic models including more than one heavy particles will help us to have clear idea about the non-local aspects of the heat bath. Therefore we studied a two-particle system using a similar Mori-Zwanzig formalism [113, 114, 115] that leads to a neatly solvable model and eventually yields to a generalized Langevin equation. This result unveils the appearance of a bath-mediated effective potential $U_{BbB}(X_1, X_2)$ that acts on the two brownian particles, so that the overall potential controlling their interaction will be the sum:

$$U(X_1, X_2) = U_0(X_1, X_2) + U_{BbB}(X_1, X_2) \tag{5.1}$$

In addition, we found that this induced potential $U_{BbB}(X_1, X_2)$ its directly related to the friction kernel and the thermal noises through the microscopic parameters and its analytic expression can be derived from the equations of motion. This allowed us to address the work needed in order to change the friction kernel, a quantity which has not been accessible on the level of Langevin dynamics.

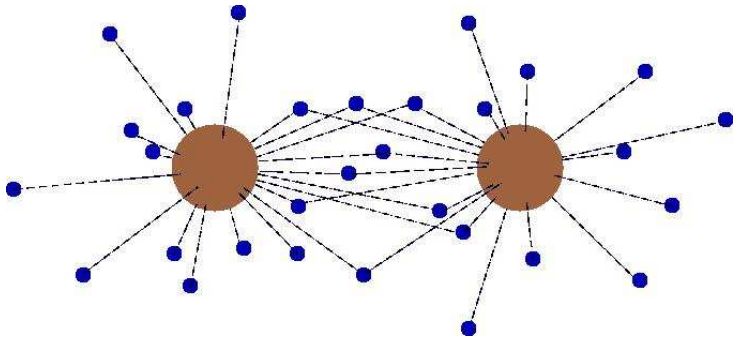


Figure 5.1: Hamiltonian model of two Brownian particles. Two mesoscopic particles are surrounded by an heat bath modeled as an infinite set of microscopic harmonic oscillators.

5.1 The model

The starting point of the Mori-Zwanzig formalism [113, 114, 115] is to write down an Hamiltonian that captures all the interactions present in the system, so that as a next step one could apply Hamilton equations to it.
In our system the whole Hamiltonian consists of two heavy (Brownian) particles of mass M_J with $J = 1$ and 2 and many light " gas" particles of mass m_i, where the index i distinguishes each gas particle. In figure 5.1 the heavy particles and the gas particles are schematically shown. Each gas particle is linked at least to one of the heavy particles $J = 1$ or 2, through the Hooke's spring with the spring constant, $m_i\omega_{i,J}^2(> 0)$, and the natural length $optimisation_{i,J}$. Some of the gas particles can be linked to both heavy particles, when $\omega_{i,1}^2\omega_{i,2}^2 > 0$. In figure 5.1 these links are represented by dashed lines. The heavy particles can interact with each other as well as with some external potentials, such as the force field of optical pincers. For the sake of solvability, the actual model is limited in the one-dimensional space while figure 5.1 gives the general idea in higher dimensions.
All these premises are represented as the following Hamiltonian, $H = H_0 + H_b + H_{bB}$:

$$H_0 = \frac{P_1^2}{2M_1} + \frac{P_2^2}{2M_2} + U_0(X_1, X_2) \tag{5.2}$$

$$H_b = \sum_i \frac{p_i^2}{2m_i}\,, \qquad H_{bB} = \sum_i \frac{m_i}{2} \sum_{J=1}^{2} \omega_{i,J}^2 (q_i - \ell_{i,J} - X_J)^2 \tag{5.3}$$

where the pairs (X_J, P_J) and (x_i, p_i) denote, respectively, the positions and momenta of the heavy (J) or light (i) particles. The bare potential energy for the heavy particles, $U_0(X_1, X_2)$, may reflect the distant-force interaction between them as function of $X_1 - X_2$, as well as external force fields on the individual heavy particles such as of optical pincers. That q_i and X_J appears as a simple difference, $(q_i - X_J)$, reflects the translational symmetry of the heat bath. In the original Zwanzig model [112] the heat bath was essentially translationally invariant with any prefactor of X_J and any natural lengths of each spring because these parameters could be absorbed by redefining q_i and the associated parameters, m_i or $\omega_{i,J}$. In the present model, however, such redefinitions are impossible in the presence of light particles coupled to the both heavy particles, i.e. with $\omega_{i,1}^2\omega_{i,2}^2 > 0$. By contrast, when $\omega_{i,J} = 0$ for a J value, the natural length $\ell_{i,J}$ is merely a dummy parameter.
One can thus derive the Hamilton equations of the Hamiltonian H for both the Brownian

particles and the light mass ones. The Brownian particles obey the following dynamics:

$$M_J \frac{d^2 X_J}{dt^2} = -\frac{\partial U_0}{\partial X_J} + \sum_i m_i \omega_{i,J}^2 (q_i - X_J - \ell_{i,J}) \tag{5.4}$$

whereas the equations for $(q_i(t), p_i(t))$ reads:

$$m_i \frac{d^2 q_i}{dt^2} = -m_i \sum_{J=1}^{2} \omega_{i,J}^2 (q_i - X_J - \ell_{i,J}) \tag{5.5}$$

Given the initial values of (q_i, p_i) we can solve for $q_i(t)$ assuming that the histories of X_J ($J = 1$ and 2) up to the time t are given. We assume that at $t = 0$ the bath variables In order to assure the compatibility with the initial canonical equilibrium of the heat bath, we assume the vanishing initial velocity for the Brownian particles $dX_J/dt|_{t=0} = 0$. Upon substituting the solutions for $q_i(t)$ the dynamics of $X_J(t)$ is rigorously reduced to

$$\begin{aligned} &M_J \frac{d^2 X_J(t)}{dt^2} \\ &= -\frac{\partial (U_0 + U_{BbB})}{\partial X_J} - \sum_{J'=1}^{2} \int_0^t K_{J,J'}(t-\tau) \frac{dX_{J'}(\tau)}{d\tau} d\tau + \epsilon_J(t), \end{aligned} \tag{5.6}$$

where we obtain the bath-mediated static potential supplementing U_0 as:

$$U_{BbB}(X_1, X_2) = \frac{k_{BbB}}{2} (X_1 - X_2 + L_{BbB})^2 \tag{5.7}$$

with

$$k_{BbB} = \sum_i \frac{m_i \omega_{i,1}^2 \omega_{i,2}^2}{\tilde{\omega}_i^2}, \quad L_{BbB} = \frac{1}{k_{BbB}} \sum_i \frac{m_i \omega_{i,1}^2 \omega_{i,2}^2 (\ell_{i,1} - \ell_{i,2})}{\tilde{\omega}_i^2}, \tag{5.8}$$

under the definition, $\tilde{\omega}_i^2 = \omega_{i,1}^2 + \omega_{i,2}^2$.
Note that U_{BbB} depends on X_1 and X_2 though the difference $X_1 - X_2$ therefore it possess translational invariance.

The friction kernel $K_{J,J'}(s)$ is found to be

$$K_{J,J'}(s) = \sum_i \frac{m_i \omega_{i,J}^2 \omega_{i,J'}^2}{\tilde{\omega}_i^2} \cos(\tilde{\omega}_i s), \tag{5.9}$$

which satisfies the symmetry properties:

$$K_{J,J'}(s) = K_{J',J}(s) = K_{J,J'}(-s) \tag{5.10}$$

If we complement (5.6) by the initial conditions, $X_J(0)$ and $dX_J/dt(0) = 0$, the noise term $\epsilon_J(t)$ is given by

$$\epsilon_J(t) \equiv \sum_i m_i \omega_{i,J}^2 \left\{ \tilde{q}_i(0) \cos(\tilde{\omega}_i t) + \frac{d\tilde{q}_i(0)}{dt} \frac{\sin(\tilde{\omega}_i t)}{\tilde{\omega}_i} \right\} \tag{5.11}$$

with

$$\tilde{q}_i(t) \equiv q_i(t) - \sum_{J=1}^{2} \frac{\omega_{i,J}^2}{\tilde{\omega}_i} [\ell_{i,J} + X_J(t)]. \tag{5.12}$$

The physical meaning of $\tilde{q}_i$ is clear from the following rewriting of the potential energy part of H_{bB} :

$$\sum_i \frac{m_i}{2} \sum_{J=1}^{2} \omega_{i,J}^2 (q_i - \ell_{i,J} - X_J)^2 = \sum_i \frac{m_i \tilde{\omega}_i^2}{2} \tilde{q}_i^2 + U_{BbB}(X_1, X_2). \tag{5.13}$$

This identity assures that, if the light particles are initially in canonical equilibrium at temperature T under a given set of $X_J(0)$ and $dX_J/dt(0) \equiv 0$, the variable $\tilde{q}_i(0)$ and $d\tilde{q}_i/dt(0)$ should

obey the equipartition laws:

$$m_i\tilde{\omega}_i^2\langle\tilde{q}_i(0)\tilde{q}_j(0)\rangle = \delta_{i,j}k_BT \quad (5.14)$$

$$m_i\langle\dot{\tilde{q}}_i(0)\dot{\tilde{q}}_j(0)\rangle = \delta_{i,j}k_BT \quad (5.15)$$

and

$$\langle\tilde{q}_i(0)\dot{\tilde{q}}_j(0)\rangle = 0 \quad (5.16)$$

where $\dot{\tilde{q}}_i = d\tilde{q}_i/dt$, and $\langle\cdot\rangle$ denotes the average over the initial canonical ensemble, $\delta_{i,j}$ is the Kronecker's delta and $k_{\rm B}$ is the Boltzmann constant.
It then leads to the fluctuation-dissipation relation of the second kind between the random forces and the friction kernel $K_{J,J'}(s)$:

$$\langle\epsilon_J(t)\epsilon_{J'}(t')\rangle = k_BT\,K_{J,J'}(t-t'). \quad (5.17)$$

In the Markovian limit where the decay of $K_{J,J'}(s)$ with s is sufficiently rapid, the friction coefficient $\gamma_{J,J'}$ is defined through the approximation, $K_{J,J'}(s) \simeq 2\gamma_{J,J'}\delta(s)$.

5.2 Analysis of the results

A first observation is that each light mass degrees of freedom (q_i, p_i) may represent one of the fluctuation modes of the heat bath that couple with the heavy mass particles. The static aspect of their coupling leads to an effective interaction potential U_{BbB} between the Brownian particles (see (5.13)), while their dynamic aspect leads to the cross-coupling , $K_{1,2}$ or $\gamma_{1,2}$ (see (5.9)). In both aspects, static and dynamic, the non-local terms arose from those modes i with $\omega_{i,1}^2\omega_{i,2}^2 > 0$.
Secondly, the presence of a bath-mediated potential U_{BbB} suggested us to pursue possible links between this quantity and the friction kernel $K_{1,2}$, either at a phenomenological level or by analytical model. In our work we propose the following general relation:

$$K_{1,2}(0) = -\frac{\partial}{\partial X_1}\frac{\partial}{\partial X_2}U_{BbB}(X_1, X_2) \quad (5.18)$$

where the right-hand side is evaluated at the equilibrium position of the Brownian particles $X_J = \langle X_J\rangle_{eq}$. The main implication of this relation is that the bath-mediated potential cannot be controlled independently from the frictional one. Our approach is to regard the heat bath as the weakly nonequilibrium system which is both perturbed and sensed by the mesoscopic Brownian particles. A general argument supporting this expression can be given using linear-response theory of non-equilibrium statistical mechanics as it is explained in more details in [111].
A part from the Hamiltonian system described in section 5.1, relation (5.18) holds exactly also for another analytically solvable model.
This is constructed by modifying the previous one by replacing the Hamiltonian evolution of each light mass particle (5.5) by the overdamped stochastic evolution governed by the Langevin equation:

$$0 = -\gamma_i\frac{dq_i}{dt} + \epsilon_i(t) - m_i\sum_{J=1}^{2}\omega_{i,J}^2(q_i - X_J(t) - \ell_{i,J}) \quad (5.19)$$

where γ_i is the friction constant with which the ith gas particle is coupled to an *outer-heat* bath of temperature T; $\epsilon_i(t)$ is a Gaussian white random force from the outer-bath obeying:

$$\langle\epsilon_i(t)\rangle = 0 \quad (5.20)$$

$$\langle\epsilon_i(t)\epsilon_{i'}(t')\,\rangle = 2k_BT\delta(t-t')\delta_{i,i'} \quad (5.21)$$

This outer-heat bath may represent those degrees of freedom of the whole-heat bath which are not directly coupled to the Brownian particles.

One can the integrate (5.19) and substitute the resulting $q_i(t)$ into the equation of motion for the Browinan particles:

$$M_J \frac{d^2 X_J(t)}{dt^2} = -\frac{\partial U_0}{\partial X_J} + \sum_i m_i \omega_{i,J}^2 (q_i - X_J - \ell_{i,J}) \tag{5.22}$$

obtaining again equations (5.6) and (5.17), the main result being the presence of the bath-mediated potential as we had before. In this overdamped model, $m_i \omega_{i,J}^2$ simply represents the spring constant between the ith light mass and the Jth Brownian particle.
The kernel and noise terms are different from the previous case:

$$K_{J,J'}(s) = \sum_i \frac{m_i \omega_{i,J}^2 \omega_{i,J'}^2}{\tilde{\omega}_i^2} e^{-(|s|/\tau_i)} \tag{5.23}$$

$$\epsilon_J(t) = \sum_i m_i \omega_{i,J}^2 \int_0^\infty \frac{e^{-(s/\tau_i)}}{\gamma_i} \epsilon_i(t-s)\, ds \tag{5.24}$$

where $\tau_i = \gamma_i/(m_i \tilde{\omega}_i^2)$. Nevertheless we have that the forms of $K_{1,2}(0)$ and $U_{BbB}(X_1, X_2)$ are the same as the previous model, those confirming our claim (5.18).
The validity of this claim implies that one cannot control the friction kernel without modifying the bath-mediated interaction between the two Brownian particles. For example suppose to modify all the $\omega_{i,J}$ by a multiplicative factor λ such that:

$$\omega_{i,J} \to \lambda \omega_{i,J} \tag{5.25}$$

this yields to change both kernel and bath-mediated potential by a factor of λ^2:

$$K_{J,J'}(s) \to \lambda^2 K_{J,J'}(s) \tag{5.26}$$

$$U_{BbB}(X) \to \lambda^2 U_{BbB}(X) \tag{5.27}$$

In particular this implies that the work W_K to change the off-diagonal friction kernel $K_{1,2}$ cannot be isolated from the work W_U to change the bath-mediated potential. From the point of view of stochastic energetics [116] this operational inseparability implies that we cannot access the work to change the friction coefficients. One could in fact identify [116] the work W_K as a function of the microscopic quantities of the above model. Therefore, unless we have access to the microscopic fluctuations of the heat bath, the quantity W_K cannot be accessed.
To summarize, we proposed a relation (5.18) between the bath-mediated effective potential and the friction memory kernel acting on two mesoscopic Brownian particles. We presented two solvable models that agree to this claim and allowed us to characterize these quantities as a function of the microscopic variables of the heat bath.
To support further these theoretical findings one should test this relation either experimentally or numerically. In particular an open question is the generalization of these results to other models [117, 118]; for example, in the reaction dynamics of protein molecules or of colloidal particles, where nonlocal fluctuations of the solvent may play important roles both kinetically and statically.

Chapter 6

Conclusions and future perspectives

The optimal routing of paths on a networks subject to non-local constraints is a challenge that can be addressed in various areas ranging from engineering to mathematics. The approach of statistical physics has been to propose the cavity method, originally developed in the context of spin glasses, as an efficient model to solve also this type of optimization problem. In this thesis we studied several variants of routing problems by applying and adapting the cavity method formalism to all the proposed different versions of routing. The message-passing equations describing the various systems have been proposed, along with their algorithmic implementations. In particular we addressed the challenge of imposing the non-local constraints for numerical implementation, and analyzed how varying a set of hard constraints affects the complexity of the problem.
Incorporating the lessons provided by the first works, namely the node-disjoint and the edge-disjoint path problems, we were able to move on to the final step in a crescendo of complexity by considering the Nash equilibrium version of the problem. Although we were able to lay down the theoretical set up, its actual performance it is still an open question at the present. Therefore as a future and on-going work it will be necessary to implement the algorithm and test the theory through numerical simulations.

The dynamic version of the cavity equations is a method that has been considered only recently. The success of the static version in solving various optimisation and inference problems suggests that the exploration of the dynamic version should be pursued, with the objective of unveiling hidden aspects associated to dynamic processes on networks that other standard algorithms, such as the Monte Carlo method, fail to capture. The Matrix Product State approximation we proposed to represent the dynamic cavity equation candidates as a promising tool to address this goal. This seminal work should therefore be treated as a starting point to explore the validity and performance of dynamic message-passing. Its implementation to study several types of dynamic processes on networks is a challenge and a task that we set for future works.

The topic of random walks on networks have been extensively studied in the last decades, its popularity being induced by the sprouting of communication and social networks. Despite the high volume of studies on this subjects, many questions are still open for answers. In this work we analyzed two of them, interpolating between spectral properties of the walk transition matrix, cavity method and large deviation theory. The main outcomes of this analysis are the characterization of the average number of distinct sites $S(n)$ visited during a random walk on a network and its behavior as a function of the graph topology. Then, starting from the theoretical set up we provided to study rare events associated to random walks on networks we were able to unveil two dynamic phase transitions.

All these findings open the stage for future explorations. In particular it would be interesting to generalize the formulation of $S(n)$ in the case of different types of random walk strategies, a characterization that has been addressed for random walks on regular topologies.
As for the rare events statistics, our preliminary work asks for a natural development: here we considered deterministic functions of the node degree but what would it happen if the functions where instead stochastic? In addition, these were defined on nodes, therefore it is a natural question to ask what would be the outcome if they were applied on edges instead.

Finally, one last remark about the non-equilibrium model presented in the last chapter of the thesis. In this work we were able to characterize the macroscopic behavior of two Brownian particles by considering the microscopic properties of the surrounding thermal bath. An interesting question to ask in this case is how would this discussion adapt in the presence of active motion. Addressing this issue would allow to shed some light on the interplay between the microscopic degrees of freedom and the macroscopic behavior of systems out-of-equilibrium. Therefore an extension of our model to the active case should be considered for future works.

1 The role of reinforcement.

In order to aid and speed-up convergence of the MP equations, we used a reinforcement technique [45, 46], in which a set of external local fields $h_{ij}^t(\mu) = E_{ij}^t(\mu) + E_{ji}^t(-\mu) - c_{ij}(\mu)$ act on the messages gradually biasing them to align with themselves. The reinforcement is introduced by promoting edge costs to become communication-dependent quantities defined as linear combinations of the cost at the previous time-step and the reinforcement local fields:

$$c_{ij}^{t+1}(\mu) = c_{ij}^t(\mu) + \gamma_t h_{ij}^t(\mu) \tag{1}$$

with $c_{ij}^0(\mu) = c_{ij}$. This cost will then be inserted into equation (2.20) to replace the term $c_{ij}(\mu)$. This has the effect to lead the messages to converge faster, gradually bootstrapping the system into a simpler one with large external fields. In practice we choose $\gamma_t = t\rho$ and one has to choose the growth rate of γ by tuning the reinforcement parameter ρ, that controls the trade-off between having a faster convergence and reaching a better solution.

2 Shuffling tensor indices.

In Chapter 3 we proposed an ansatz to write the dynamic cavity messages as a product of matrices (3.18). In order to have consistent expressions in both the sides of that equation we have to define matrices $A(\sigma^{s-1}|\sigma'^s)$ which have a dependence on σ^s reversed when compared with the one of the transition rate $W(\sigma^s|\{\sigma'^{s-1}\}_{\sigma'\in\partial\sigma})$. However one could exploit the properties of tensors and reshuffle consistently the indices in order to recover anstaz (3.18). Here we explain in more details the operations performed over the tensors in order to update the cavity messages in the MPS form. These have been developed by Thomas Barthel and allow for a natural algorithmic implementation.
The starting point is equation (3.28):

$$\mu_{ij}(\bar{\sigma}_i^{t+1}|\bar{\sigma}_j^t) = \left[\prod_{s=1}^{t+1} C_{ij}^{(s)}(\sigma_i^s|\sigma_j^{s-1})\right] C_{ij}^{(0)}(\sigma_i^0) \tag{2}$$

Notice the dependence on σ^s, at this point it is the same as for the transition rates.
Before giving all the operational details we describe the general procedure that will help us to follow the various steps. The entire routine is made of three sweeps, each one being a set of operations, either SVD or truncations, performed in one direction on the product appearing in (2).
The first is a *preparatory* sweep made from right to left, i.e. from $C^{(0)}$ to $C^{(t+1)}$. The aim of this is to impose orthonormality condition to the right basis of $|\psi\rangle$ as in the MPS formalism described in section 3.1, thus in this sweep we will preform SVD but no truncation. This operation will be denoted using the symbol $\overset{\text{SVD}}{=}$.
The result will be to transform the matrix C to a modified $\tilde{C}$ describing the same vector $|\psi\rangle$ but this time obeying the *right orthonormal condition*:

$$\sum_n \tilde{C}^n[\tilde{C}^n]^\dagger = \mathbb{1} \tag{3}$$

such that the reduced right basis for all the bipartitions of the system are orthonormal. In this sweep we do not truncate nor move any index, so the matrix dimension remains the same.
After this preparatory sweep two others will be performed, this time applying truncation along with SVD. This combined operation will be denoted with $\overset{\text{SVD}}{\approx}$.
The first sweep will be from left to right and the goal is to apply a fist shuffle to the time indices from:

$$(\sigma_i^{t+1}, \sigma_j^t)\,,\ (\sigma_i^t, \sigma_j^{t-1})\ \ldots\ (\sigma_i^1, \sigma_j^0)\,,\ (\sigma_i^0) \tag{4}$$

of the matrices $\tilde{C}$ to:

$$(\sigma_i^{t+1})\,,\,(\sigma_i^t,\sigma_j^t)\,,\,(\sigma_i^{t-1},\sigma_j^{t-1})\,\ldots\,(\sigma_i^0,\sigma_j^0) \tag{5}$$

by introducing a new set of matrices D such that (2) will be transformed to:

$$\mu_{ij}(\bar{\sigma}_i^{t+1}|\bar{\sigma}_j^t) = D_{ij}^{(t+1)}(\sigma_i^{t+1})\left[\prod_{s=0}^{t} D_{ij}^{(s)}(\sigma_i^s|\sigma_j^s)\right] \tag{6}$$

After this a final sweep will be performed from right to left. Here we apply again SVD and truncation, with the aim of yielding a second shuffling to the time indices from:

$$(\sigma_i^{t+1})\,,\,(\sigma_i^t,\sigma_j^t)\,,\,(\sigma_i^{t-1},\sigma_j^{t-1})\,\ldots\,(\sigma_i^0,\sigma_j^0) \tag{7}$$

of the matrices D to:

$$(\sigma_i^{t+1})\,,\,(\sigma_i^t)\,,\,(\sigma_i^{t-1},\sigma_j^t),\,\ldots\,(\sigma_i^0,\sigma_j^1)(\sigma_j^0) \tag{8}$$

which is the final form appearing in the matrices B of the canonical form (3.19).
Now we describe in more details each of the three sweeps.

Preparatory sweep: $C \to \tilde{C}$. This sweep is performed from right to left and SVD with no truncation are applied. Starting from the right boundary ($s = 0$):

$$C^{(0)}(\sigma_i^0) \overset{\text{SVD}}{=} U^{(0)}\Lambda^{(0)}\tilde{C}^{(0)}(\sigma_i^0) \tag{9}$$

where $\tilde{C}^{(0)}(\sigma_i^0)$ obeys orthonormal condition.
In the bulk, i.e. $0 < s < t+1$, we have:

$$C^{(s)}(\sigma_i^s,\sigma_j^{s-1}) \overset{\text{SVD}}{=} U^{(s)}\Lambda^{(s)}\tilde{C}^{(s)}(\sigma_i^s,\sigma_j^{s-1}) \tag{10}$$

where again $\tilde{C}^{(s)}(\sigma_i^s,\sigma_j^{s-1})$ is orthonormal.
Finally in the left boundary, i.e. $s = t+1$, we obtain:

$$C^{(t+1)}(\sigma_i^{t+1},\sigma_j^t)U^{(t)}\Lambda^{(t)} = \tilde{C}^{(t+1)}(\sigma_i^{t+1},\sigma_j^t) \tag{11}$$

but this time $\tilde{C}^{(t+1)}(\sigma_i^{t+1},\sigma_j^t)$ is not orthonormal. This will be fixed in the next sweep though.

Second sweep: $\tilde{C} \to D$. The sweep is performed from left to right, this time we apply truncation along with SVD. Starting from the left boundary ($s = t+1$):

$$\tilde{C}^{(t+1)}(\sigma_i^{t+1},\sigma_j^t) \overset{\text{SVD}}{\approx} D^{(t+1)}(\sigma_i^{t+1})\Lambda^{(t+1)}V^{(t+1)}(\sigma_j^t) \tag{12}$$

where now $D^{(t+1)}(\sigma_i^{t+1})$ obeys left orthonormal condition.
In the bulk, i.e. $0 < s < t+1$, moving to the right we have:

$$\Lambda^{(s+1)}V^{(s+1)}(\sigma_j^s)\tilde{C}^{(s)}(\sigma_i^s,\sigma_j^{s-1}) \overset{\text{SVD}}{\approx} D^{(s)}(\sigma_i^s,\sigma_j^s)\Lambda^{(s)}V^{(s)}(\sigma_j^{s-1}) \tag{13}$$

where again $D^{(s)}(\sigma_i^s,\sigma_j^s)$ is left orthonormal. Finally at the right boundary, i.e. $s = 0$, we obtain:

$$\Lambda^{(1)}V^{(1)}(\sigma_j^0)\tilde{C}^{(0)}(\sigma_i^0) = D^{(0)}(\sigma_i^0,\sigma_j^0) \tag{14}$$

where this last step is just a definition and no SVD has been performed. Note also that $D^{(0)}(\sigma_i^0,\sigma_j^0)$ is orthonormal. From now on both left and right basis will be orthonormal therefore we can truncate freely without worrying about altering the Frobenius norm and thus the quality of the approximation.

Third sweep: $D \to B$. The sweep is performed from right to left, we apply truncation along with SVD. Starting from the right boundary ($s = 0$):

$$D^{(0)}(\sigma_i^0,\sigma_j^0) \overset{\text{SVD}}{\approx} U^{(0)}(\sigma_i^0)\Lambda^{(0)}B^{(0)}(\sigma_j^0) \tag{15}$$

where $B^{(0)}(\sigma_j^0)$ obeys right orthonormal condition.
In the bulk, i.e. $0 < s < t+1$, moving to the left we have:

$$D^{(s)}(\sigma_i^s, \sigma_j^s) U^{(s-1)}(\sigma_i^{s-1}) \Lambda^{(s-1)} \overset{\text{SVD}}{\approx} U^{(s)}(\sigma_i^s) \Lambda^{(s)} B^s(\sigma_i^{s-1}, \sigma_j^s) \tag{16}$$

where again $B^s(\sigma_i^{s-1}, \sigma_j^s)$ is right orthonormal.
Finally at the left boundary, i.e. $s = t+1$, we obtain the last two matrices B as:

$$U^{(t)}(\sigma_i^t)\Lambda^{(t)} \overset{\text{SVD}}{\approx} U^{(t+1)}\Lambda^{(t+1)} B^{t+1}(\sigma_i^t) \tag{17}$$

$$D^{(t+1)}(\sigma_i^{t+1}) U^{(t+1)} \Lambda^{(t+1)} = B^{t+2}(\sigma_i^{t+1}) \tag{18}$$

where this last step is just a definition and no SVD has been performed.
The overall outcome of these three sweep will be the expression (3.19):

$$\mu_{ij}(\bar{\sigma}_i^{t+1} | \bar{\sigma}_j^t) = B_{ij}^{(t+2)}(\sigma_i^{t+1}) B_{ij}^{(t+1)}(\sigma_i^t) \left[\prod_{s=1}^{t} B_{ij}^{(s)}(\sigma_i^{s-1} | \sigma_j^s) \right] B_{ij}^{(0)}(\sigma_j^0) \tag{19}$$

which is consistent with the ansatz (3.18).

3 Random networks.

Throughout the thesis we have implemented many numerical simulations over various random network topologies. This section is thus devoted to description of the different topologies used in the main text so that the reader unacquainted could find here all the information needed to understand what type of graph we have considered in our analysis.

Regular graphs. The simpler type of random graph is the k-regular random graph. This is defined as the random graph where each node has fixed degree k. The topology is thus homogenous and this often allows to derive analytical results as we have done in section 4.2.3. In the statistical physics community this is often confused with the Bethe lattice of connectivity k. Although similar they are not equivalent because the latter is defined as an infinite tree where each node has the same degree k. The main difference between the two is than the presence of loops in the case of a regular graph. However this difference becomes more and more negligible as the system size increases, i.e. $V \to \infty$, thus one can interchange these two definition for large sizes.
The finite version of the Bethe lattice is called Cayley tree and this should not be confused with a regular graph instead. The Cayley tree is in fact rich of leaves of connectivity $k = 1$, also called the surface of the tree. The finite character thus makes this surface highly non-negligible and this fact has a huge impact on the phenomena happening on the graph. In particular this can be seen when studying dynamical processes such as the majority rule where a non-negligible number of leaves keeps flickering between two states; thus impacting the total magnetization of a great amount.

Erdős-Rényi graphs. An Erdős-Rényi (ER) graph is the most general prototype of a random graph and was first introduced in [33]. There are two equivalent, but different, definitions that one can arbitrarily choose. The one we used in the numerical simulations is the one that fixes the system size V and a real number $p \in [0, 1]$ such that: for each possible pair of nodes (i, j), with $i, j \in \mathcal{V}$, an edge is added between them with constant probability p. One can thus tune p in order to control the density of edges, the largest limit being $p = 1$ that leads to a fully connected graph with $V(V-1)/2$ edges.
Considering large system sizes V, the degree distribution tends to a Poissonian of the form:

$$P(k) \propto \frac{\langle k \rangle^k e^{-\langle k \rangle}}{k!} \tag{20}$$

where the average degree is linked to the parameter p through $pV = \langle k \rangle$.

Scale-free graphs. Contrarily to what has been said fro ER graphs, in the case of Scale-free (SF) graphs there is not a unique definition to build it. Instead it is defined as a graph where the degree distribution is a power-law of the form:

$$P(k) \propto k^{-\gamma} \tag{21}$$

where γ is a real parameter that one tunes in order to control the degree heterogeneity: the smaller γ the higher the heterogeneity among the realized degrees. Usually one chooses $\gamma \in (2, 3)$ because it has been shown [83] that several real networks fall into this interval. Their main feature is that they have *fat tails*, i.e. there is a non-negligible probability of having nodes with high degree and these are usually called hubs. Among the various existing methods to build them, in this work we used the preferential attachment routine proposed in [83]. The scheme works as follows: start with a graph of m_0 vertices and introduce sequentially $V - m_0$ new vertices by attaching each of them to m already existing nodes. The probability π to pick a certain node i as one of these m neighbors is proportional to its degree:

$$\pi(k_i) \sim k_i \tag{22}$$

thus high degree nodes will be more likely to be picked and hence they will increase their degree while the graph grows (a rule that is often informally called *rich gets richer*). This scheme leads to a power-law degree distribution with $\gamma = 2.9 \pm 0.1$ [83]; we empirically observe this value in our simulations. We also tried other generation methods for scale-free graphs that yield different values of γ, but as the results were qualitatively similar to those for preferential attachment we only show the latter as representative for our scale-free graph simulations.

RER graphs The last type of graph is an interpolation between a k-regular graph and an Erdős-Rényi one that we proposed in order to perturbe the homogenous topology of the regular graph with a noise that follows the ER rule, thus avoiding to achieve too large degree heterogeneity as in the SF graphs. The RER graph is built as follows: we start from a k_0-regular random graph which correspond to the basic structure of the graph. Then edges (that are not yet present) are added on top of this structure independently with probability p as in the ER model; if p is chosen such that $d = pV$ then the final average degree of such a graph is $\langle k \rangle = k_0 + d$ for large V. This graph ensemble thus interpolates between the regular and ER cases and is similar to the one analyzed in [119, 82] with the difference that here we start from a regular graph instead of a ring or a lattice.

Bibliography

[1] Marc Mézard and Giorgio Parisi. The bethe lattice spin glass revisited. *The European Physical Journal B-Condensed Matter and Complex Systems*, 20(2):217–233, 2001.

[2] Marc Mézard and Giorgio Parisi. The cavity method at zero temperature. *Journal of Statistical Physics*, 111(1-2):1–34, 2003.

[3] Mézard M, Parisi G, and Zecchina R. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, 2002.

[4] Alfredo Braunstein, Marc Mézard, and Riccardo Zecchina. Survey propagation: An algorithm for satisfiability. *Random Structures & Algorithms*, 27(2):201–226, 2005.

[5] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003.

[6] Mézard M and Montanari A. *Information, physics, and computation*. Oxford University Press, 2009.

[7] Pearl J. *Proceedings American Association of Artificial Intelligen ce National Conference on AI*, (Pittsburgh, PA, USA, 1982):133–136, 1982.

[8] Frank R Kschischang, Brendan J Frey, and H-A Loeliger. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519, 2001.

[9] Andrea Montanari and Tommaso Rizzo. How to compute loop corrections to the bethe approximation. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(10):P10011, 2005.

[10] Mézard M and Parisi G. The cavity method at zero temperature. *J. Stat. Phys.*, 111(1-2):1–34, 2003.

[11] Izaak Neri and Désiré Bollé. The cavity approach to parallel dynamics of ising spins on a graph. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(08):P08009, 2009.

[12] Erik Aurell and Hamed Mahmoudi. A message-passing scheme for non-equilibrium stationary states. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(04):P04014, 2011.

[13] Erik Aurell and Hamed Mahmoudi. Dynamic mean-field and cavity methods for diluted ising systems. *Physical Review E*, 85(3):031119, 2012.

[14] Hamed Mahmoudi and David Saad. Generalized mean field approximation for parallel dynamics of the ising model. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(7):P07001, 2014.

[15] Feller W. *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, 2008.

[16] Andrey Y Lokhov, Marc Mézard, Hiroki Ohta, and Lenka Zdeborová. Inferring the origin of an epidemic with a dynamic message-passing algorithm. *Physical Review E*, 90(1):012801, 2014.

[17] Fabrizio Altarelli, Alfredo Braunstein, Luca Dall'Asta, Alejandro Lage-Castellanos, and Riccardo Zecchina. Bayesian inference of epidemics on networks via belief propagation. *Physical review letters*, 112(11):118701, 2014.

[18] Roy J Glauber. Time-dependent statistics of the ising model. *Journal of mathematical physics*, 4(2):294–307, 1963.

[19] William A Little. The existence of persistent states in the brain. In *From High-Temperature Superconductivity to Microminiature Refrigeration*, pages 145–164. Springer, 1996.

[20] P Peretto. Collective properties of neural networks: a statistical physics approach. *Biological cybernetics*, 50(1):51–62, 1984.

[21] Daniel J Amit. *Modeling brain function: The world of attractor neural networks*. Cambridge University Press, 1992.

[22] Eric Goles and Gérad Y Vichniac. Lyapunov functions for parallel neural networks. In *Neural networks for computing*, volume 151, pages 165–181. AIP Publishing, 1986.

[23] Yashodhan Kanoria, Andrea Montanari, et al. Majority dynamics on trees and the dynamic cavity method. *The Annals of Applied Probability*, 21(5):1694–1748, 2011.

[24] Mézard M, Parisi G, and Virasoro M A. *Spin glass theory and beyond*, volume 9. World scientific Singapore, 1987.

[25] Paulo RA Campos, Viviane M de Oliveira, and FG Brady Moreira. Small-world effects in the majority-vote model. *Physical Review E*, 67(2):026104, 2003.

[26] Luiz FC Pereira and FG Brady Moreira. Majority-vote model on random graphs. *Physical Review E*, 71(1):016123, 2005.

[27] Claudio Castellano and Romualdo Pastor-Satorras. Zero temperature glauber dynamics on complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(05):P05001, 2006.

[28] James P Gleeson. Binary-state dynamics on complex networks: pair approximation and beyond. *Physical Review X*, 3(2):021004, 2013.

[29] R Abou-Chacra, DJ Thouless, and PW Anderson. A selfconsistent theory of localization. *Journal of Physics C: Solid State Physics*, 6(10):1734, 1973.

[30] R M Karp. *Reducibility among combinatorial problems*. Springer, 1972.

[31] Yeung C H, Saad D, and Wong KY M. From the physics of interacting polymers to optimizing routes on the london underground. *Proc. Nat. Ac. Sci.*, 110(34):13717–13722, 2013.

[32] Lovász L and Plummer D. *Matching Theory*. AMS Chelsea Publishing Series. American Mathematical Soc., 2009.

[33] Erdős P and Rényi A. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.

[34] Robertson N and Seymour P D. Graph minors. xiii. the disjoint paths problem. *J. of Comb Th, B*, 63(1):65–110, 1995.

[35] Aggarwal A, Kleinberg J, and Williamson D P. Node-disjoint paths on the mesh and a new trade-off in vlsi layout. In *Proc. of the twenty-eighth annual ACM symposium on Theory of computing*, pages 585–594. ACM, 1996.

[36] Chekuri C and Ene A. Poly-logarithmic approximation for maximum node disjoint paths with constant congestion. In *SODA*, pages 326–341, 2013.

[37] Caterina De Bacco, Silvio Franz, David Saad, and C H Yeung. Shortest node-disjoint paths on random graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(7):P07009, 2014.

[38] Albert R and Barabási A-L. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[39] Robertson N and Seymour P D. Graph minors. xiii. the disjoint paths problem. *Journal of combinatorial theory, Series B*, 63(1):65–110, 1995.

[40] Chatterjee B C, Sarma N, Sahu P P, et al. Review and performance analysis on routing and wavelength assignment approaches for optical networks. *IETE Technical Review*, 30(1):12, 2013.

[41] Fabrizio Altarelli, Alfredo Braunstein, Luca Dall'Asta, Caterina De Bacco, and Silvio Franz. The edge-disjoint path problem on random graphs by message-passing. *arXiv preprint arXiv:1503.00540*, 2015.

[42] Galil Z, Micali S, and Gabow H. An $o(ev\backslash logv)$ algorithm for finding a maximal weighted matching in general graphs. *SIAM Journal on Computing*, 15(1):120–130, 1986.

[43] Blesa M and Blum C. Ant colony optimization for the maximum edge-disjoint paths problem. In *App. Ev. Comp.*, pages 160–169. Springer, 2004.

[44] Pham Q D, Deville Y, and Van Hentenryck P. Ls (graph): a constraint-based local search for constraint optimization on trees and paths. *Constraints*, 17(4):357–408, 2012.

[45] Braunstein A and Zecchina R. Learning by message passing in networks of discrete synapses. *Physical review letters*, 96(3):030201, 2006.

[46] Altarelli F, Braunstein A, Realpe-Gomez J, and Zecchina R. Statistical mechanics of budget-constrained auctions. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(07):P07002, 2009.

[47] Yeung C H and Saad D. Competition for shortest paths on sparse graphs. *Physical Review Letters*, 108(20):208701, 2012.

[48] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[49] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[50] Fabrizio Altarelli, Alfredo Braunstein, Luca Dall'Asta, Caterina De Bacco, Silvio Franz, David Saad, and Riccardo Zecchina. The nash equilibrium in routing problems by message-passing. *Manuscript in preparation*, 2015.

[51] Robert W Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

[52] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge, 2007.

[53] Scott Kirkpatrick, MP Vecchi, et al. Optimization by simmulated annealing. *science*, 220(4598):671–680, 1983.

[54] Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The complexity of pure nash equilibria. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 604–612. ACM, 2004.

[55] David S Johnson, Christos H Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of computer and system sciences*, 37(1):79–100, 1988.

[56] Thomas Barthel, Caterina De Bacco, and Silvio Franz. The matrix-product representation of dynamic cavity equations. *Manuscript in preparation*, 2015.

[57] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011.

[58] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[59] Claude Cohen-Tannoudji, Bernard Diu, and Frank Laloe. *Quantum Mechanics (2 vol. set)*. Wiley-Interscience, October 2006.

[60] László Lovász. Random walks on graphs: A survey. *Combinatorics, Paul erdos is eighty*, 2(1):1–46, 1993.

[61] Caterina De Bacco, Satya N Majumdar, and Peter Sollich. The average number of distinct sites visited by a random walker on random graphs. *Journal of Physics A: Mathematical and Theoretical*, 48(20):205004, 2015.

[62] Caterina De Bacco, Alberto Guggiola, Kühn Reimer, and Pierre Paga. Rare events statistics of random walks on networks: localization and other dynamical phase transitions. *Manuscript in preparation*, 2015.

[63] Chung F RK. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[64] Dirac P A M. A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 35:416–418, 7 1939.

[65] Frobenius G. Über matrizen aus nicht negativen elementen. *Sitzungsber. Königl. Preuss. Akad. Wiss. Berlin*, page 456–477, 1912.

[66] Perron O. Zur theorie der matrices. *Mathematische Annalen*, 64(2):248–263, 1907.

[67] F. R. Gantmacher. *Applications of the Theory of Matrices*. Interscience, New York, 1959.

[68] Jasch F and Blumen A. Target problem on small-world networks. *Physical Review E*, 63(4):041108, 2001.

[69] Jasch F and Blumen A. Trapping of random walks on small-world networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 64(6 Pt 2):066104–066104, 2001.

[70] Beeler Jr JR. Distribution functions for the number of distinct sites visited in a random walk on cubic lattices: Relation to defect annealing. *Physical Review*, 134(5A):A1396, 1964.

[71] Klafter J and Blumen A. Models for dynamically controlled relaxation. *Chemical physics letters*, 119(5):377–382, 1985.

[72] Cattuto C, Barrat A, Baldassarri A, Schehr G, and Loreto V. Collective dynamics of social annotation. *Proceedings of the National Academy of Sciences*, 106(26):10511–10515, 2009.

[73] Yeung C H and Saad D. Networking—a statistical physics perspective. *Journal of Physics A: Mathematical and Theoretical*, 46(10):103001, 2013.

[74] Vineyard G H. The number of distinct sites visited in a random walk on a lattice. *Journal of Mathematical Physics*, 4(9):1191–1193, 1963.

[75] Montroll E W and Weiss G H. Random walks on lattices. ii. *Journal of Mathematical Physics*, 6(2):167–181, 1965.

[76] Dvoretzky A and Erdös P. Proceedings of the second berkeley symposium. 1951.

[77] Larralde H, Trunfio P, Havlin S, Stanley H E, and Weiss G H. Number of distinct sites visited by n random walkers. *Physical Review A*, 45(10):7128, 1992.

[78] Kundu A, Majumdar S N, and Schehr G. Exact distributions of the number of distinct and common sites visited by n independent random walkers. *Physical review letters*, 110(22):220602, 2013.

[79] Hughes Barry D and Sahimi M. Random walks on the bethe lattice. *Journal of Statistical Physics*, 29(4):781–794, 1982.

[80] Burioni R and Cassi D. Random walks on graphs: ideas, techniques and results. *Journal of Physics A: Mathematical and General*, 38(8):R45, 2005.

[81] Redner S. *A guide to first-passage processes*. Cambridge University Press, 2001.

[82] Watts D J and Strogatz S H. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[83] Barabási A-L and Albert R. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[84] Friedman J. On the second eigenvalue and random walks in random d-regular graphs. *Combinatorica*, 11(4):331–362, 1991.

[85] Broder A and Shamir E. On the second eigenvalue of random regular graphs. In *Foundations of Computer Science, 1987., 28th Annual Symposium on*, pages 286–294. IEEE, 1987.

[86] Farkas I J, Derényi I, A-L Barabási, and Vicsek T. Spectra of "real-world" graphs: Beyond the semicircle law. *Physical Review E*, 64(2):026704, 2001.

[87] Füredi Z and Komlós J. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.

[88] Chung F, Lu L, and Vu V. Spectra of random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 100(11):6313–6318, 2003.

[89] Rogers T, Castillo I P, Kühn R, and Takeda K. Cavity approach to the spectral density of sparse symmetric random matrices. *Physical Review E*, 78(3):031116, 2008.

[90] Edwards S F and Jones R C. The eigenvalue spectrum of a large symmetric random matrix. *Journal of Physics A: Mathematical and General*, 9(10):1595, 1976.

[91] Urry M J and Sollich P. Random walk kernels and learning curves for Gaussian process regression on random graphs. *The Journal of Machine Learning Research*, 14(1):1801–1835, 2013.

[92] Urry M J and Sollich P. Replica theory for learning curves for Gaussian processes on random graphs. *Journal of Physics A: Mathematical and Theoretical*, 45(42):425005, 2012.

[93] Sollich P, Tantari D, Annibale A, and Barra A. Extensive load in multitasking associative networks. *arXiv:1404.3654*, 2014.

[94] Kühn R. Spectra of sparse random matrices. *Journal of Physics A: Mathematical and Theoretical*, 41(29):295002, 2008.

[95] Alon N. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.

[96] Nilli A. On the second eigenvalue of a graph. *Discrete Mathematics*, 91(2):207–210, 1991.

[97] V. Kishore, M. S. Santhanam, and R. E. Amritkar. Extreme events and event size fluctuations in biased random walks on networks. *Phys. Rev. E*, 85:056120, May 2012.

[98] J. P. Garrahan, R. L. Jack, V. Lecomte, E. Pitard, K. van Duijvendijk, and F. van Wijland. First-order dynamical phase transition in models of glasses: an approach based on ensembles of histories. *J. Phys. A*, 42:075007, 2009.

[99] R. L. Jack and P. Sollich. Large deviations and ensembles of trajectories in stochastic models. *Progr. of Theor. Phys. Supplement*, 184:304–317, 2010.

[100] Amir Dembo and Ofer Zeitouni. *Large deviations techniques and applications*, volume 38. Springer Science & Business Media, 2009.

[101] Hugo Touchette. The large deviation approach to statistical mechanics. *Physics Reports*, 478(1):1–69, 2009.

[102] Hugo Touchette. Legendre-fenchel transforms in a nutshell. *URL http://www.physics.sun.ac.za/ htouchette/archive/lfth2.pdf*, 2005.

[103] R Tyrrell Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1970.

[104] Cornelius C. Lanczos. *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Governm. Press Office, 1950.

[105] Yoshiyuki Kabashima, Hisanao Takahashi, and Osamu Watanabe. Cavity approach to the first eigenvalue problem in a family of symmetric random sparse matrices. In *Journal of Physics: Conference Series*, volume 233, page 012001. IOP Publishing, 2010.

[106] Mark EJ Newman. Random graphs as models of networks. *arXiv preprint cond-mat/0202208*, 2002.

[107] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. *arXiv preprint arXiv:1408.2701*, 2014.

[108] Thomas Guhr, Axel Müller-Groeling, and Hans A Weidenmüller. Random-matrix theories in quantum physics: common concepts. *Physics Reports*, 299(4):189–425, 1998.

[109] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[110] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[111] Caterina De Bacco, Fulvio Baldovin, Enzo Orlandini, and Ken Sekimoto. Nonequilibrium statistical mechanics of the heat bath for two brownian particles. *Physical review letters*, 112(18):180605, 2014.

[112] Robert Zwanzig. Nonlinear generalized langevin equations. *Journal of Statistical Physics*, 9(3):215–220, 1973.

[113] Hazime Mori. Transport, collective motion, and brownian motion. *Progress of theoretical physics*, 33(3):423–455, 1965.

[114] Robert Zwanzig. Ensemble method in the theory of irreversibility. *The Journal of Chemical Physics*, 33(5):1338–1341, 1960.

[115] Robert Zwanzig. Memory effects in irreversible thermodynamics. *Physical Review*, 124(4):983, 1961.

[116] Ken Sekimoto. *Stochastic energetics*, volume 799. Springer, 2010.

[117] Rongxin Huang, Isaac Chavez, Katja M Taute, Branimir Lukić, Sylvia Jeney, Mark G Raizen, and Ernst-Ludwig Florin. Direct observation of the full transition from ballistic to diffusive brownian motion in a liquid. *Nature Physics*, 7(7):576–580, 2011.

[118] Lydéric Bocquet and Jarosław Piasecki. Microscopic derivation of non-markovian thermalization of a brownian particle. *Journal of statistical physics*, 87(5-6):1005–1035, 1997.

[119] Monasson R. Diffusion, localization and dispersion relations on "small-world" lattices. *The European Physical Journal B-Condensed Matter and Complex Systems*, 12(4):555–567, 1999.

Publications

2015 **A matrix product algorithm for stochastic dynamics on locally tree-like graphs**, *T. Barthel, C. De Bacco, S. Franz*, http://arxiv.org/abs/1508.03295.

2015 **Rare events statistics for random walks on networks**, *C. De Bacco, A. Guggiola, P. Paga, R. Khun*, http://arxiv.org/abs/1506.08436.

2015 **Non-equilibrium statistical mechanics of the heat bath for two Brownian particles— Internal degrees of freedom found where there shouldn't be**, *C. De Bacco, F. Baldovin, E. Orlandini, K. Sekimoto*, Proceedings ASTE Vol.B11 2015.3.

2015 **The edge-disjoint path problem on random graphs by message-passing**, *F. Altarelli, A. Braunstein, L. Dall'Asta, C. De Bacco and S. Franz*, http://arxiv.org/abs/1503.00540.

2014 **The average number of distinct sites visited by a random walker on random graphs**, *C. De Bacco, S. Majumdar and P. Sollich* , Journal of Physics A: 48.20 (2015): 205004.

2014 **Shortest node-disjoint paths on random graphs**, *C. De Bacco, S. Franz, D. Saad, C.H. Yeung*, J. Stat. Mech. (2014) P07009.

2014 **Non-equilibrium statistical mechanics of the heat bath for two Brownian particles**, *C. De Bacco, F. Baldovin, E. Orlandini, K. Sekimoto*, Phys. Rev. Lett. 112, 180605.

2012 **Microscopic characterization of active matter**, *Caterina De Bacco*, Master thesis, http://archivio.fisica.unipd.it.
Supervisors: F. Baldovin, E. Orlandini.

Shortest node-disjoint paths on random graphs

C De Bacco[1], S Franz[1], D Saad[2] and C H Yeung[2,3,4]

[1] LPTMS, Centre National de la Recherche Scientifique et Universite Paris-Sud 11, 91405 Orsay Cedex, France.

[2] The Nonlinearity and Complexity Research Group, Aston University, Birmingham B4 7ET, United Kingdom.

[3] Department of Physics, The Hong Kong University of Science and Technology, Hong Kong.

[4] Department of Science and Environmental Studies, The Hong Kong Institute of Education, Hong Kong.

E-mail: caterina.de-bacco@lptms.u-psud.fr

Abstract. A localized method to distribute paths on random graphs is devised, aimed at finding the shortest paths between given source/destination pairs while avoiding path overlaps at nodes. We propose a method based on message-passing techniques to process global information and distribute paths optimally. Statistical properties such as scaling with system size and number of paths, average path-length and the transition to the frustrated regime are analyzed. The performance of the suggested algorithm is evaluated through a comparison against a greedy algorithm.

1. Introduction

Among the various computationally-hard constraint satisfaction problems, routing and path optimization have attracted particular attention in recent years due to their non-localized nature and interdisciplinary relevance. The node-disjoint path (NDP) problem on graphs studied here, aims at finding a set of paths linking specified pairs of nodes (communications) such that no two paths share a node; the problem is classified among the NP-complete class [1] of hard combinatorial problems. This has not only been studied as a purely theoretical problem by mathematicians in the series of graph minors [2] under the name of subgraph homeomorphism problem, but also by practitioners due to its wide applicability to various fields. For instance, in communication systems where the network performance is often strictly related to capacity limits, traffic congestion and the rate of information flow; and in problems of virtual circuit routing where switches located at nodes may become bottlenecks. Moreover, due to their distributive nature NDP is more resilient to failure and represents one aspect of optimal routing where network robustness is the main objective.

One specific communication application where efficient and effective NDP algorithms are essential is in the area of optical networks where transmissions using the same wavelength cannot share the same edge or vertex, hence all communications of the same wavelength must be non-overlapping (disjoint). Consequently, such an algorithm impacts on the achievable network capacity and transmission rate. In this field of routing and wavelength assignment [3], the objective is to find a routing assignment that minimizes the number of wavelengths used. Different techniques that exploit disjoint paths heuristic algorithms have been proposed to tackle this problem; for instance, greedy algorithms [4, 5], approximations based on rounding integer linear programming formulations [6, 7], post-optimization methods [8], bin packing algorithms [9], various heuristic genetic algorithms such as ant colony optimization [10] and differential evolution [11].

Another important application of NDP is in the design of very large system integrated circuits (VLSI), where one searches for non-overlapping wired paths to connect different integrated hardware components, to avoid cross-path interference.

Similarly, in wireless ad-hoc communication networks [12, 13, 14], where each node can act as a router, path overlaps imply signal interference and low transmission quality, whereas longer paths imply poor signal to noise ratio due to multiple relays and higher transmission power; hence the need to consider both path length and transmission overlaps to be minimized is essential for routing problems. Solutions to the NDP problem also provide fault tolerant routes due to the optimal separation of communication paths all over the network, so that if a node (router) fails, as frequently happens in wireless networks due to the mobility of hosts, only few communications will be affected [15, 16]. This feature is particularly important when quality of service (QoS) is one of the main requirements in the set up of a communication network, along with the load-balancing feature of NDP that prevents network congestion by establishing non-overlapping routes. This is especially relevant to connection-oriented networks [17] that are strongly affected by node failures and congestion [18].

Practical algorithms for various applications often depend on the specific network topologies considered [19] and mostly focus on the optimization version of the problem, i.e. maximizing the number of paths routed [20]. The satisfiability version of the problem, i.e. whether all paths can be routed successfully without overlap, is not considered; theoretical studies often give bounds to the achievable approximation instead of providing a practical algorithm for individual instances and fail to calculate path lengths and possible overlaps at the same time as part of the optimization process observables. Given that paths are constrained to be contiguous and interaction between paths is non-localized, a local protocol is insufficient and global optimization is required. The computational complexity is determined by the fact that such a global optimization problem has to consider all variables simultaneously in order to minimize a cost function with non local interactions between variables.

Unlike other constrained satisfaction problems on networks, NDP has received little attention within the statistical physics community. In this paper we consider a random version of NDP on regular graphs (Reg), Erdős Rényi (ER) [21] and a dedicated type of random graph (RER) described in Section 4, with the aim of testing the efficacy of statistical physics-based methods derived in the context of spin glass theory [22] such as belief propagation or message-passing (MP) cavity method [23, 24] as viable alternatives to greedy algorithms; we also study statistical and scaling properties of quantities of interest as a function of network size and number of paths. We study sparse regular, ER and RER random graphs as they are the most interesting for the problem at hand, but the methodology can be easily extended to accommodate other sparsely connected architectures. Clearly, due to the hard constraint of node disjoint paths, typically no solutions would be found in graphs having a non negligible number of nodes with degree $k = 1, 2$. Moreover, graphs with a small number of high degree nodes (hubs) or with high modularity measure, such as scale-free or planar graphs, are not interesting for the node-disjoint routing problem since when a paths passes through one of these special nodes it leads directly to graph fragmentation, hence frustration. The situation would be very different for constraints on edges instead, but this variant of the problem is left for future work. Finally, the requirement for the graph to be sparse is suggested by restrictions on the validity of the cavity method which is based on fast decaying correlation functions, i.e. a negligible number of loops in the graph.

Numerical simulations indicate that MP outperforms greedy breadth-first search algorithms not only in finding better solution but also in reaching a higher frustration threshold. Moreover, we find scaling of the expected total length of the NDP as a function of the system size and graph connectivity that goes as $\frac{M \log V}{V \log^{\gamma}(k-1)}$ with exponent γ that depends on the type of graph, where V is the number of nodes and M the number of paths. We find good agreements between theory and simulation data for graphs of average degrees $k = 3, 5, 7$ and sizes $V = 1000, 2000, 4000, 5000, 10000$. Finally, we study statistical properties of physical quantities observed a posteriori, i.e. when a solution is found, such as path length distribution, degree distribution and maximum cluster size for the case of regular graphs.

The reminder of the paper is organized as follows: in Section 2 we will introduce the model used followed by the algorithmic solution in Section 3. Results obtained from numerical studies will be presented in Section 4 followed by conclusions and future research

directions in Section 5.

2. Model

Given an undirected graph (or network) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ characterized by $V = |\mathcal{V}|$ nodes and $E = |\mathcal{E}|$ edges we define a set of M communications C as paths on edges of the graph, each of which originates from a source node S and terminates in a receiver node R. We introduce a variable Λ_i^μ to characterize each node $i \in \mathcal{V}$:

$$\Lambda_i^\mu = \begin{cases} +1 & \text{if } i \text{ is a sender for communication } \mu \\ -1 & \text{if } i \text{ is a receiver for communication } \mu \\ 0 & \text{if } i \text{ is neither a sender nor a receiver for communication } \mu \end{cases} \tag{1}$$

The full node characterization is specified by a vector $\bar{\Lambda}_i := (\Lambda_i^1, \dots, \Lambda_i^M)$ of modulus $||\bar{\Lambda}_i|| := \sum_{\mu=1}^M |\Lambda_i^\mu| \in \{0, 1\}$, where $|\Lambda_i^\mu|$ denotes the absolute value of Λ_i^μ; $||\bar{\Lambda}_i||$ is 0 if i is neither a sender nor a receiver for any communication, termed a transit node, and 1 if i is either a sender or a receiver of some communication. In this way each node can send or receive at most one communication.

For a given set of M sender-receiver pairs (S^μ, R^μ) with $\mu = 1, \dots, M$ we address the problem of finding a set of communications that optimize a cost function which penalizes path length and prevents communications overlap (traffic). The state of the network can be specified by introducing a variable I_{ij}^μ for each edge $(ij) \in \mathcal{E}$ and for each communication μ, which specifies whether communication μ passes through edge (ij) and in which direction:

$$I_{ij}^\mu = \begin{cases} +1 & \text{if } \mu \text{ passes through } (ij) \text{ from } i \text{ to } j \\ -1 & \text{if } \mu \text{ passes through } (ij) \text{ from } j \text{ to } i \\ 0 & \text{if } \mu \text{ does not pass through } (ij) \end{cases} \tag{2}$$

Notice that in this formalism $I_{ij}^\mu = -I_{ji}^\mu$. We term these variables currents and define for each edge (ij) a vector $\bar{I}_{ij} := (I_{ij}^1, \dots, I_{ij}^M)$ that collects information on all currents involved in that edge. Currents are subject to Kirchhoff law:

$$\sum_{j \in \partial i} I_{ij}^\mu - \Lambda_i^\mu = 0 \qquad \forall \mu = 1, \dots, M \, . \tag{3}$$

For a given path optimization problem we seek the communication configuration C^* that minimizes a cost function $c(\{\bar{I}_{ij}\})$, which penalizes path length and traffic congestion:

$$c(\{\bar{I}_{ij}\}) = \sum_{(ij) \in \mathcal{E}} f(||\bar{I}_{ij}||) \tag{4}$$

where $f(||\bar{I}_{ij}||)$ is a monotonically increasing function of $||\bar{I}_{ij}|| := \sum_{\mu=1}^M |I_{ij}^\mu|$, that penalizes both congestion and path length; where $|I_{ij}^\mu|$ denotes the absolute value of I_{ij}^μ.

We would like now to search for approximate solutions to this problem by message-passing equations [23]. To derive a distributed algorithm it is useful first to consider tree-like graphs $\mathcal{T}$, for which one can derive exact recursive equations, and later on use these equations as an approximation for arbitrary graphs $\mathcal{G}$.

If $\mathcal{T}$ is a tree, the removal of any edge $(ij) \in \mathcal{E}$ divides $\mathcal{T}$ in two disjoint subtrees T_i and T_j (see

figure 1). We can define $\hat{E}_{ij}(\bar{I})$ as the optimized cost on T_i when the current $\bar{I}$ flows through on the edge (ij); in this way we can write the message sent from node i to his neighbor j when the current $\bar{I}_{ij}$ flows through the edge (ij) as $E_{ij}(\bar{I}_{ij}) := \hat{E}_{ij}(\bar{I}_{ij}) + f(\|\bar{I}_{ij}\|)$.

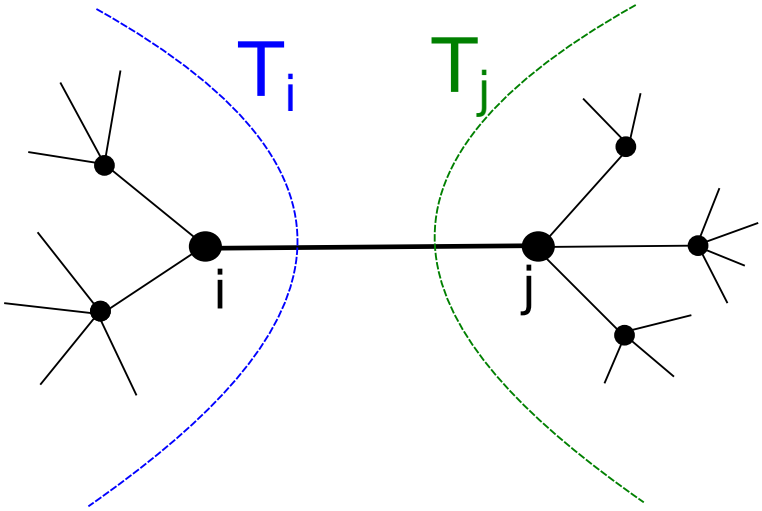


Figure 1: Subtree division. T_i is the subtree rooted in i when the edge (ij) is removed. Conversely, T_j is the subtree rooted in j after the same edge removal.

The messages $E_{ij}(\bar{I}_{ij})$ admit the *min-sum* [23] recursion relation:

$$E_{ij}(\bar{I}_{ij}) = \min_{\bar{I}_{ki}|constraint} \left\{ \sum_{k\in\partial i\setminus j} E_{ki}(\bar{I}_{ki}) \right\} + f(\|\bar{I}_{ij}\|) \,, \tag{5}$$

where the symbol ∂i stands for the set of neighbors of node i and the *constraint* is the Kirchoff law (3).
In the following we will use the recursion equation (5) on arbitrary random graphs to approximate the constrained minimum of $c(\{\bar{I}_{ij}\})$, the cost defined in equation (4). Namely:

$$c^* := \min_{\bar{I}} \frac{1}{E} \sum_{(ij)\in\mathcal{E}} \left\{ E_{ij}(\bar{I}) + E_{ji}(-\bar{I}) - f(\|\bar{I}\|) \right\} \tag{6}$$

where the last subtracted term is introduced to avoid double counting the cost of edge (ij).
Unfortunately, the computational complexity of this algorithm is exponential in the number of communications M. In fact, messages $E_{ij}(\bar{I}_{ij})$ can a priori take 3^M values corresponding to all possible currents passing through a single edge (ij). Therefore, we cannot generally treat even moderately large values of M [25]. The problem can be simplified if we introduce the hard constraint that paths cannot overlap on nodes (and thus neither on edges). This has the important consequence of reducing the configuration space from 3^M to $2M + 1$ and the computational complexity becomes linear in M. This restricted version of the path optimization problem is called the node-disjoint path problem (NDP), as we already mentioned in the introduction, and is the problem we address here. Notice that since we

impose the node-disjoint constraint for the communications, then one communication at most flows through the edges, so that $\|\bar{I}_{ij}\| = \sum_{\mu=1}^{M} |I_{ij}^{\mu}| \in \{0, 1\}$. This corresponds to taking:

$$f(\|\bar{I}\|) = \begin{cases} \infty & \text{if} \quad \|\bar{I}\| \geq 2 \\ 1 & \text{if} \quad \|\bar{I}\| = 1 \\ 0 & \text{if} \quad \|\bar{I}\| = 0 \end{cases} \tag{7}$$

so that the cost function (4) represents indeed the total path length.
In order to solve equation (5) iteratively we define a protocol for taking into account only the allowed configurations at each edge given the current value $\bar{I}$ passing through it and $\bar{\Lambda}_i$ at vertex i.

If $|\bar{\Lambda}_i| = 0$ then:

$$E_{il}(\bar{I}_{il} = \bar{0}) = \min \left\{ \sum_{j \in \partial i \setminus l} E_{ji}(\bar{I}_{ji} = \bar{0}), \right. \tag{8}$$

$$\left. \min_{j_1, j_2 \in \partial i \setminus l; \mu \in M} \left[E_{j_1 i}(I_{j_1 i}^{\mu} = +1) + E_{j_2 i}(I_{j_2 i}^{\mu} = -1) + \sum_{k \in \partial i \setminus l, j_1, j_2} E_{ki}(\bar{I}_{ji} = \bar{0}) \right] \right\}$$

$$E_{il}(I_{il}^{\mu} = \pm 1) = \min_{j \in \partial i \setminus l} \left\{ E_{ji}(I_{ji}^{\mu} = \pm 1) + \sum_{k \in \partial i \setminus l, j} E_{ki}(\bar{I}_{ki} = \bar{0}) \right\} + 1 \tag{9}$$

If $\Lambda_i^{\mu} = \pm 1$ then:

$$E_{il}(\bar{I}_{il} = \bar{0}) = \min_{j \in \partial i \setminus l} \left\{ E_{ji}(I_{ji}^{\mu} = \mp 1) + \sum_{k \in \partial i \setminus l, j} E_{ki}(\bar{I}_{ki} = \bar{0}) \right\} \tag{10}$$

$$E_{ji}(I_{ji}^{\nu} = \pm 1) = +\infty \quad (\nu \neq \mu) \tag{11}$$

$$E_{ji}(I_{ji}^{\mu} = \mp 1) = +\infty \tag{12}$$

$$E_{ji}(I_{ji}^{\mu} = \pm 1) = \sum_{j \in \partial i \setminus l} E_{ji}(\bar{0}) + 1 \tag{13}$$

The constant +1 that appears equations (9) and (13) are the costs assigned for a unit of current passing through the considered edge (i.e. $f(1) = 1$). This cost is the one required for the shortest paths but can be generalized to other arbitrary types of costs.

Equation (8) represents the case where i is a transit node and no current passes through edge (ij), then the allowed configurations are that either no currents pass through the remaining neighboring edges (first term inside curly brackets) or one current enters and then exits i through a pair of neighboring edges, all others edges being unused (second term inside brackets). In figure 2 you can see a diagram representing the different allowed configurations for a transit node. Equation (9) represents the case where i is a transit node and the communication μ passes through edge (ij); in this case the only allowed configuration is that where the same communication μ enters/exits from one of the other neighboring edges, all others being unused. Similar considerations are used to formulate the equations (10-13) for senders and receivers.

The procedures of applying the algorithm can be summarized as follows:

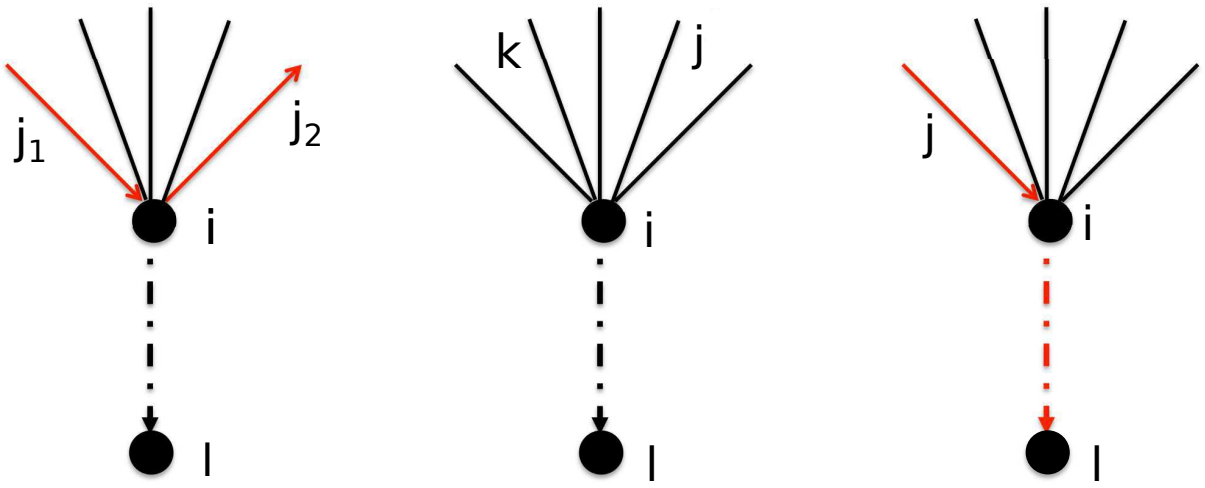


Figure 2: Transit node cavity diagram. Left and center represent the two terms inside the min brackets in equation (8). Right represents equation (9) .

- Initialize messages at random.
- Pick in random order all $i \in \mathcal{V}$ and update messages using (8), (9) and (10-13) until convergence is reached (i.e., message changes are below a given threshold).
- Use the converged messages to calculate physical observables.

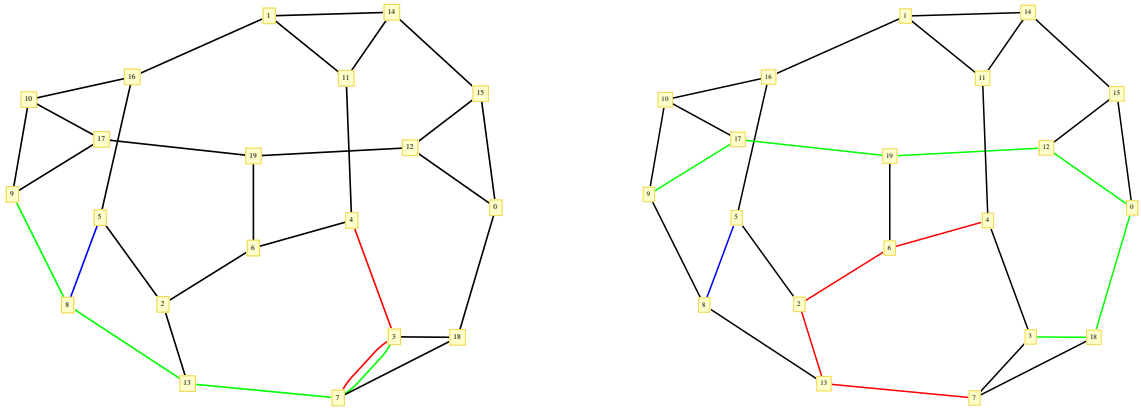


Figure 3: Edge-disjoint routes. For a single instance of a graph of size $V = 20$ and $M = 3$ communications we have on the left the shortest paths and on the right the optimal non-overlapping solutions. We can see that the green path has to be re-routed to avoid both blue and red communications. Also the red one cannot take its shortest path because the sender of the green communication is on that path.

3. Obtaining a solution

Once the iterative equations (8), (9) and (10-13) have converged, the resulting messages can be used to calculate the solution. We define the energy per link [24]:

$$E_{ij}^{Link}(\bar{I}) := \left\{ E_{ij}(\bar{I}) + E_{ji}(-\bar{I}) - ||\bar{I}|| \right\} \tag{14}$$

where the last term on the right $||\bar{I}||$ is subtracted to avoid double counting as it appears in both of the previous two terms. To find a solution we calculate:

$$E_{ij}^{*Link} := \min_{\bar{I}} E_{ij}^{Link}(\bar{I}) \tag{15}$$

for each link in the graph and store the current values that minimize the energy per link for each edge:

$$\bar{I}_{ij}^{*} := arg\, min_{\bar{I}}\, E_{ij}^{Link}(\bar{I}) \,. \tag{16}$$

Eventually, we sum over all $(ij) \in \mathcal{E}$ to find the different paths and total length:

$$L_{tot} := \sum_{(ij)\in\mathcal{E}} ||\bar{I}_{ij}^{*}|| \,. \tag{17}$$

In the cavity formalism [24] this is equivalent to calculating the quantity:

$$E_i := \min_{\bar{I}_{ki}|constraint} \sum_{k\in\partial i} E_{ki}(\bar{I}_{ki}) \,, \tag{18}$$

which represents the energy per node. Finally, the total energy (or path length) is the combination of the two, which in the case of a $k-$regular graph establishes the formal relation:

$$E_t = \sum_{i\in\mathcal{V}} E_i - \frac{k}{2} \sum_{(ij)\in\mathcal{E}} E_{ij}^{Link} \,. \tag{19}$$

Notice that the calculation of E_{ij}^{Link*} is carried out link by link as if the energies per link were statistically independent. It is not intuitively clear that doing this will result in the optimal paths which do not overlap and are also fully connected from the source to the receiver. This is a consequence of having used messages which implicitly contain global information on the constraints and path lengths, so that the energies per link are indeed globally interdependent albeit in a non-obvious manner.

To fully characterize the solutions statistically we calculate also other observables as explained below. Finally, we calculate the paths and corresponding lengths in a sequential order using a greedy breadth first-search local algorithm (BFS) to compare the results obtained against our MP-based algorithm.

3.1. Algorithmic complexity

The node-disjoint constraint is very restrictive and algorithmically helpful in comparison to other routing models where overlaps are allowed but minimized [25, 26]. This hard constraint is indeed paramount in reducing considerably the algorithm's computational complexity. If we allow for overlaps we need to span a configuration space of the order of 3^M at each cavity iteration, leading to a complexity of $O(N\,(3^M)^{k-1})$, where the exponent $k-1$ explores

the different flux combinations for each of the $k-1$ independent neighboring sites of the considered message; there are order of N such messages. In order to tackle this issue proper approximations have to be introduced as in [25, 26] where they use techniques from polymer physics [25] or convexity properties of the cost function [26]. On the contrary when the overlap is prohibited we reduce the configuration space from 3^M to $(2M+1)$, as this is the number of allowed configurations (the term $2M$ is derived from the number of possible currents $I^\mu = \pm 1$ and the additional $+1$ is due to the configuration of all 0), hence there is no need for approximations because the entire configuration space can be efficiently calculated by the cavity equation. Actually, the use of cavity MP implicitly requires one important approximation as it assumes that when node $i \in \mathcal{V}$ is removed, all its neighbors are statistically independent. This is equivalent of having fast decaying correlation function between these neighboring nodes. This hypothesis is verified in trees and in locally tree-like sparse graphs.

For the same reason is important to distinguish between edge and node overlaps. In this work we chose to consider constraints on nodes motivated by the reduced complexity as explained before; in case of edge constraints one has to consider a much bigger configuration space where all configurations with different communications entering and exiting the same transit node must be considered in the optimization routine. For this reason approximations should be introduced as in the case of the models which minimize overlap. The edge-disjoint variant of the problem will be left for future work.

We performed single instance simulations to find optimal microscopic solutions; to obtain macroscopic averages one would usually use population dynamics, one of the most commonly used numerical tools in statistical mechanics literatures [27, 23] for studying similar models. Population dynamics is considered when the thermodynamic limit $V \to \infty$ is taken and the system size is not fixed a priori as in the single instance algorithm. In our model the use of population dynamics does not make much sense since the parameter M enters explicitly in the expressions of the messages because it represents the domain of the fluxes, which is of size $2M+1$. But when we fix M at the same time we are fixing a system size V, because we extract random pairs (S, R) with density M/V. Hence, it is impossible to decouple the messages domain from the system size, preventing us to properly employ the thermodynamic limit through population dynamics. There is also another problem, that such a macroscopic oriented approach would introduce averages over all possible configurations (S, R), including both frustrated and unfrustrated configurations with much higher energies. Thus the macroscopic averages are highly biased by the fewer frustrated configurations and more complex algorithm should be designed to discard such cases. For these two reasons we did not consider in the following the population dynamic counterpart of the algorithm but focused only on averages over single instances.

3.2. Greedy algorithm

To test the performance of the algorithm we compared the results obtained with those given by a greedy algorithm (or its variant) that is often used in literatures to solve the NDP problem in different contexts [12, 13, 4, 5]. The greedy protocol considers only local information around

the sources and then builds up a solution step by step recursively, hence reducing considerably the complexity but at the same time completely ignoring other communication positions in the network.

A typical greedy algorithm works in the following way: start by choosing an arbitrary pair (S, R), find the shortest path linking the two nodes and then remove nodes belonging to this path from the available network nodes. Choose a second pair and repeat the procedures until either all the M paths from sources to destinations have been established or no solution can be found due to frustration. Clearly, the performance of this algorithm is strictly dependent on the order in which we choose the pairs. For instance, in the extreme case the first pair selected is the one with the longest shortest path among all the M communications; this implies that we have effectively a more restricted graph and choice of paths, leading for a long second path and even more restricted choice of paths later on.

4. The results

We performed numerical simulations on three types of random graphs. Standard regular random graphs (Reg): each node has fixed degree k; Erdős Rényi random graphs (ER) [21]: edges are drawn at random between each pair of nodes with probability $p = \langle k \rangle / V$; a decorated random graph (RER): starting from a regular random graph of degree k_1 (which is the minimum degree of this graphs), we then randomly add new edges as in the ER model until the final average connectivity is $\langle k \rangle = k > k_1$. Notice that the degree distribution in this case can be obtained by writing $k = k_1 + d$ where d is Poisson distributed with $\langle d \rangle = k - k_1$. The parameters used were average degrees $\langle k \rangle = 3, 5, 7$ and system sizes of $V = 1000, 2000, 4000, 5000, 10000$. We calculated averages over $[50 - 500]$ realizations for both the MP and the greedy algorithms; we used a smaller number of realizations for cases of higher complexity (as for $V = 10^4$ and $k = 7$). Nevertheless, results in all cases are stable and with small error bars with respect to the symbols used. We omitted the error bars from the figures for clarity.

We found a system size scaling that is a cubic function of the variable $x := \frac{M \log V}{V \log^{\gamma}(k-1)}$. A qualitative explanation of the scaling is as follows. The average path length in random graphs goes as $\langle l \rangle \sim \log V / \log \langle k \rangle$ (see [28] for an extensive review of graphs properties) and in our case we have M paths to consider.We can refine the dependence on k using instead $\langle l \rangle \sim \log V / \log (k-1)$. Now, suppose all communications take their shortest path, the quantity $x = M \log V / \log^{\gamma}(k-1)$ would be a good estimate of graph occupancy for the NDP, where the exponent γ has been introduced as a free parameter to account for the approximation in the expression for $\langle l \rangle$ as a function of k for different types of graphs. Furthermore, if we divide by the number of available nodes V we can define the occupancy ratio as $\frac{M \log V}{V \log^{\gamma}(k-1)} = x$. Therefore in this simple case we would expect L_{tot}/V increasing linearly in x. If overlaps are prohibited, for a sufficiently high value of M the communications are increasingly forced to take longer routes, leading to a faster than linear increase in the scaling variable x. From numerical simulations we found for the NDP a cubic increase $\frac{L_{tot}}{V} = ax + cx^3$ in the scaling variable $x = \frac{M \log V}{V \log^{\gamma}(k-1)}$. For small M this function agrees well with the linear shortest path

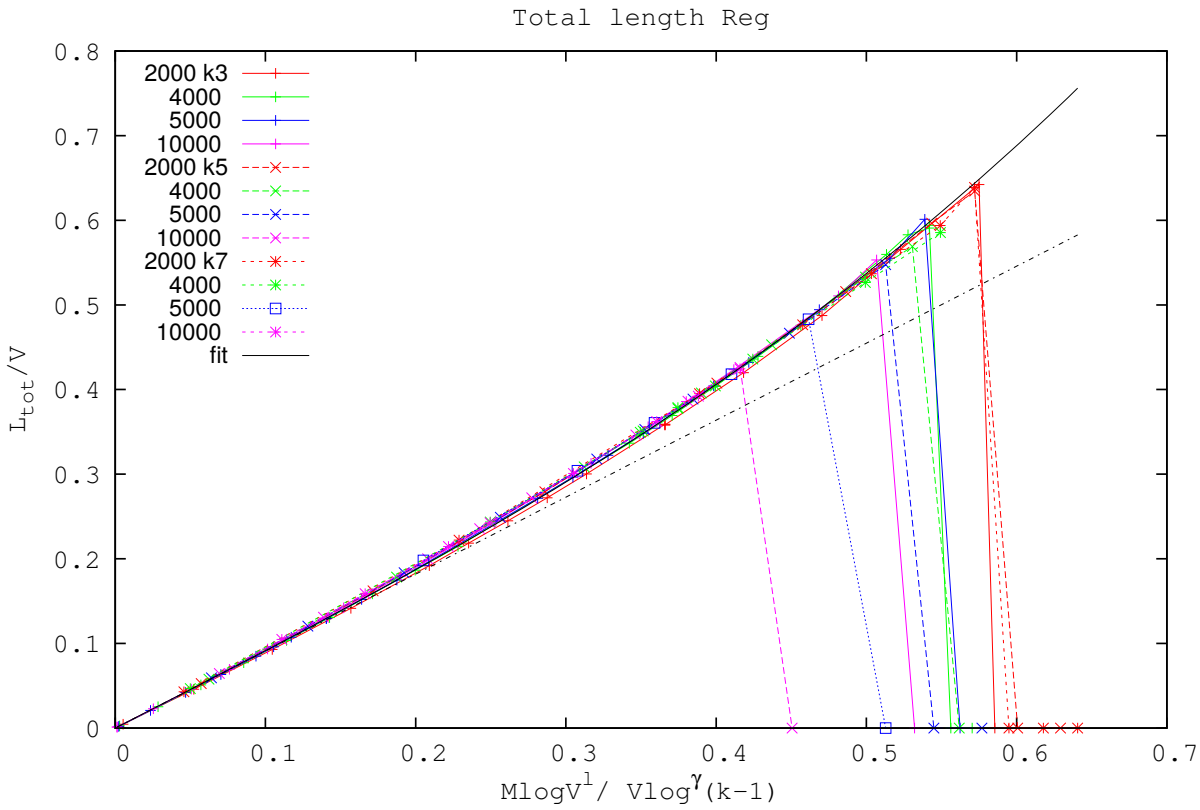


Figure 4: Expected total normalized path length - regular graphs (Reg). We obtained results for regular graphs of fixed degree $k = 3, 5, 7$ and for system size $V = 2000, 4000, 5000, 10000$. We found a global scaling rule with respect to the variable $x := M \log V^l / V \log^\gamma(k-1)$, with $l = 1.00$ and $\gamma = 0.87$. In black (solid line) we draw the cubic fit $ax + cx^3$, with $a = 0.910 \pm 0.001$ and $c = 0.660 \pm 0.009$. The dotted-dashed line represent the shortest path (not considering overlap) trivial solution. Error are not reported because smaller or comparable to point sizes.

behavior but for values of $x > 0.2$ the steeper increase of L_{tot} becomes predominant. Figures 4 and 5 show a good data collapse of the normalized expected total length per node L_{tot}/V as a function of the scaling variable x for different graph connectivities for Reg and ER graphs respectively. We notice a first regime where the curves follow the linear behavior of the dashed line representing the shortest paths. The term "sparse regime" is used since paths are sufficiently far apart, M is small, and then no re-routing is needed as each communication will simply take its shortest path. For $x > 0.2$ the curves show the cubic steeper behavior that represents the increase in path lengths to avoid overlaps. The term "dense phase" reflects the increase in path density; M is sufficiently high so that shortest-path choices induce conflicting demands and communications are rerouted, taking longer paths to avoid overlaps.

Finally, for large M we can identify different frustration points, represented by vertical lines in figure 4, that connect the largest M for which solutions have been found with the points where frustration is reached and the length is set to 0 by convention. We see that the frustration points do not collapse and that the bigger the graph size V and the higher the

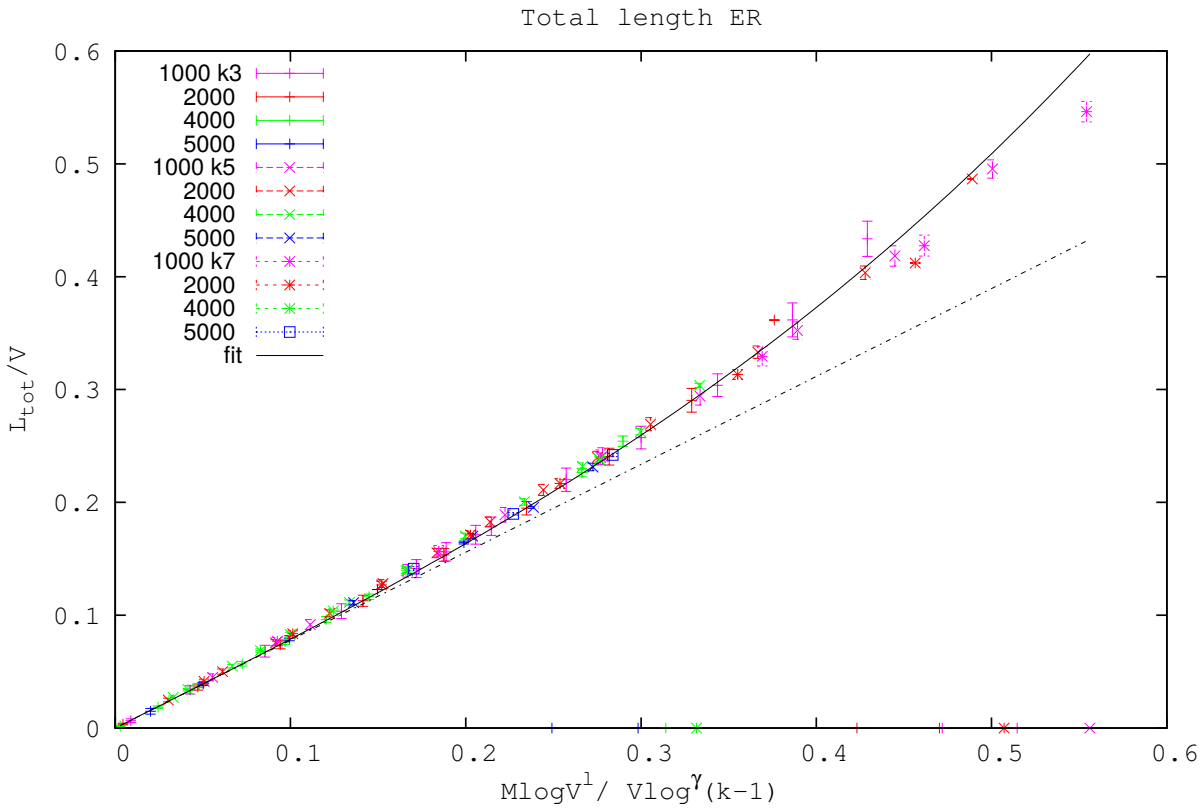


Figure 5: Expected total normalized path length - Erdős Rényi (ER). We obtained results for ER graphs of expected degree $\langle k \rangle = 3, 5, 7$ and for system size $V = 1000, 2000, 4000, 5000$. We found a global scaling rule with respect to the variable $x := M \log V^l / V \log^\gamma (k-1)$, with $l = 1.00$ and $\gamma = 0.69$. In black (solid line) we draw the cubic fit $ax + cx^3$, with $a = 0.77 \pm 0.01$ and $c = 0.96 \pm 0.03$. The dotted-dashed line represent the shortest path (not considering overlap) trivial solution.

connectivity k, the earlier frustration sets in (as a functions of x). Arguably, this is due to algorithmic convergence rather than theoretical arguments. In fact the higher V and k are, the higher the corresponding algorithmic complexity, and hence the larger the number of iterations required to reach convergence. Due to the prohibitive computation cost we ran a smaller number of instances for higher values of V and k, and without increasing the preset maximum convergence time. We suspect that convergence can be reached in these cases albeit in a much longer times, and hence a solution could be found as well in theory, but has not been found due to the computational limits imposed. Hence we can not provide a precise measure for the frustration transitions nor make further statements regarding their collapses for different systems sizes and connectivities.

In figure 6 we can see the scaling behavior for Reg, ER and RER of given average connectivity and different system sizes; we fixed $\gamma = 0$ arbitrarily to highlight the dependence on V. We can notice how different types of graph, although having different average lengths, follow the same cubic scaling in x. The steeper slope of the ER graphs shows the smaller number of paths choices in this type of graphs that forces the path to rewire in increasingly

more convoluted patterns and hence also reach frustration earlier.

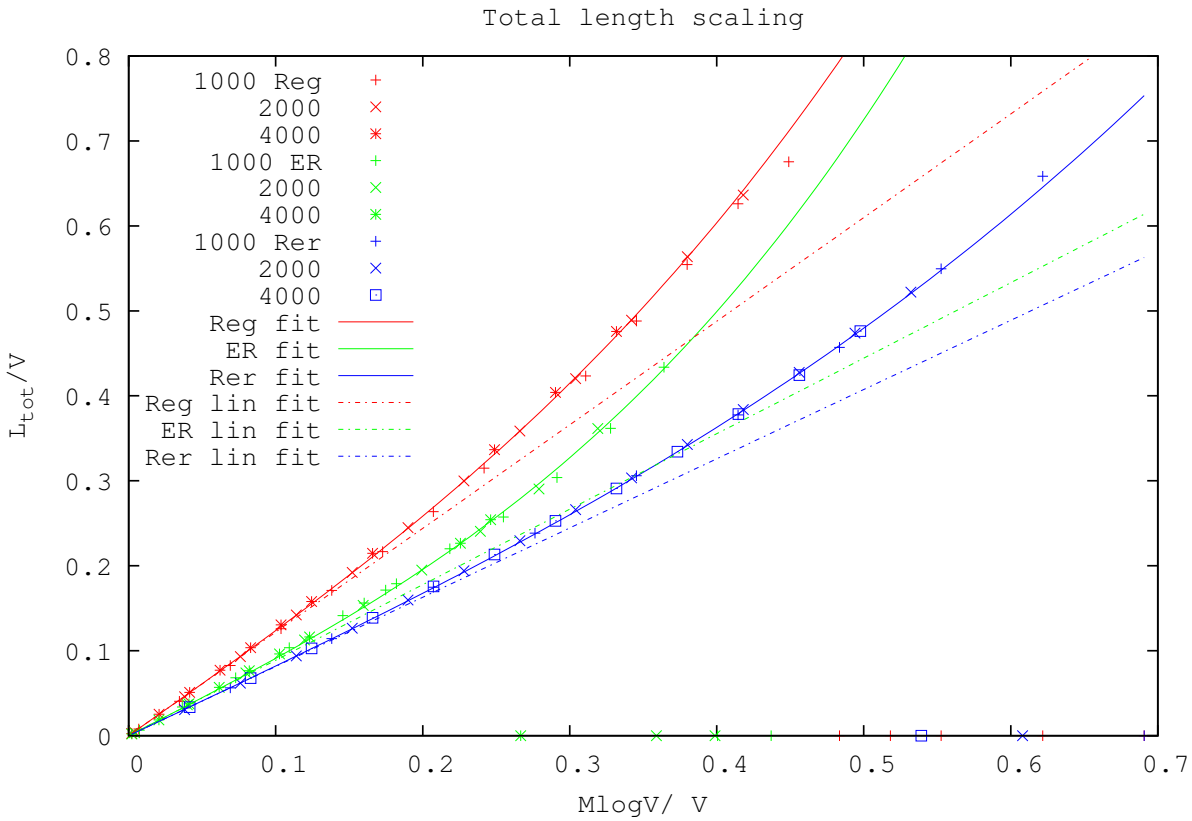


Figure 6: Scaling in total length. We plotted the scaling of the total length per node as a function of $M \log V/V$ for Reg, ER and RER graphs of different system sizes; $\langle k \rangle = 3$ for Reg and ER and $\langle k \rangle = 4$ for RER. We can see different slopes in the cubic fits of the various curves, for instance ER graphs achieve shorter lengths but with higher cubic slope, meaning that their length increases faster with traffic due to the smaller number of path choices.

We found that our MP algorithm outperforms the greedy BFS both in finding a better solution (smaller L_{tot}) and in reaching higher values of the frustration transitions. In figures 7, 8 and 9 we plotted the expected normalized total length for both greedy and MP algorithms, for the three graph types, fixed connectivity and different system sizes. We focused on the case $\langle k \rangle = 3$ for Reg and ER and $\langle k \rangle = 4$ for RER because of its lower complexity compared to higher k; nevertheless, simulations for different k values indeed agree with the suggested scaling and exhibit the same behavior. Initially, in the x range where solutions exist the greedy algorithm gives the same total length as the global algorithm up to a certain value of M (and x). The explanation is that in this interval the graph is sparse, communications *typically* do not interact and shortest paths can be selected. This also shows that for a small number of paths the global procedure reduces to act similarly to the greedy algorithm does, e.g. when rerouting is required it involves only two paths, the optimal solution will adopt the shortest path for one and will reroute the second. When M increases, we see that the global optimization algorithm outperforms the greedy approach in both finding the optimal solution

and in achieving a higher frustration threshold. In the regime where the global algorithm gives a better solution (i.e. shorter total length) we see that it is more efficient to globally reroute paths rather than taking the shortest path of selected paths and adapt the other paths. This means that the optimal solution is not a simple superposition of the first n-shortest path of the M communications, but is a more complex solution.

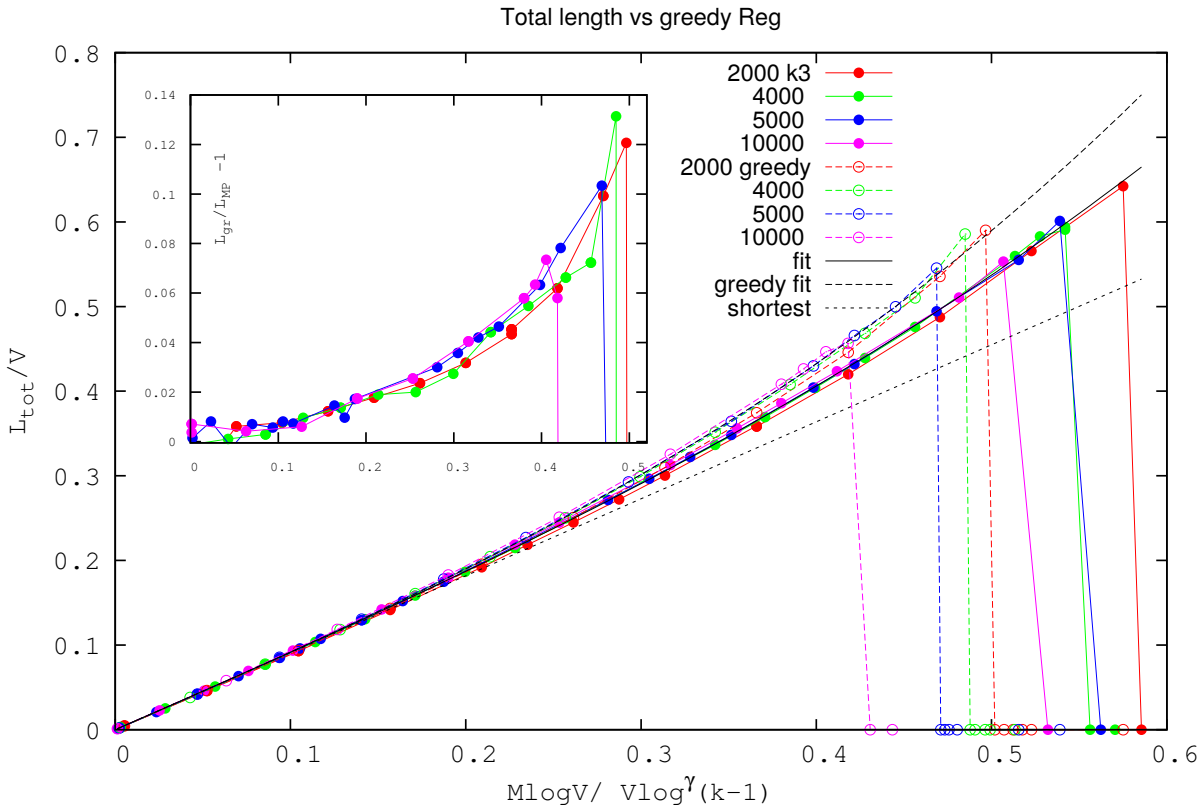


Figure 7: Expected total normalized path length - greedy vs global optimization algorithms - Reg graphs. We compared results obtained by a greedy BFS algorithm and our global optimization for system size $V = 2000, 4000, 5000, 10000, \gamma = 0.87$ and degree $k = 3$. We can identify the sparse interval where the algorithms give similar results because communications are far apart and take the shortest paths. As the number of communications grow we observe an intermediate regime where global optimization performs better than BFS; and finally the dense regime where the greedy BFS algorithm fails to find a solution whereas the global optimization algorithm succeeds up to a critical M value. Cubic fits are also plotted (solid black line for global optimization, dashed line for the greedy BFS algorithm) whereas the dotted line represents the shortest path (not considering overlap) trivial solution, i.e. the sum of the M shortest path lengths, which is linear in x. Vertical lines show the frustration points where no solution is found and the total path length is set to zero. Inset: Ratio $L_{greedy}/L_{MP} - 1$ is plotted as a function of x. Notice the worse performance of the greedy algorithm.

Figure 10 shows the failure ratio defined as the number of unsuccessful instances (for which a solution is not found) over the total number of realizations as a function of the scaling variable x. We notice that the greedy algorithm reaches the frustration point (as a function of x) earlier than the corresponding global MP algorithm, regardless the system sizes or graph

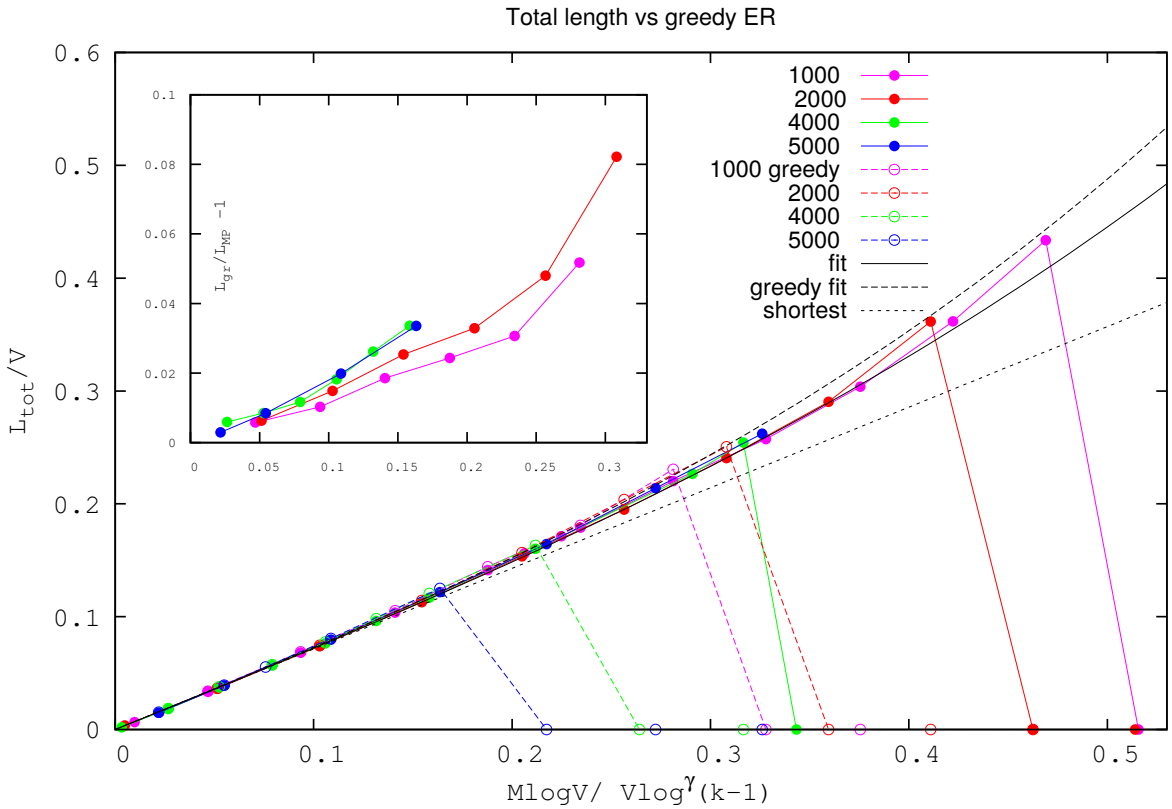


Figure 8: Expected total normalized path length - greedy vs global optimization algorithms ER. System sizes $V = 1000, 2000, 4000, 5000$, $\gamma = 0.69$ and degree $\langle k \rangle = 3$. Inset: the ratio $L_{greedy}/L_{MP} - 1$ is plotted as a function of x. Notice the worse performance of the greedy.

type.

This shows that, if a solution exists, a global management of the entire set of communications is required in order to find an optimal solution. Whereas if each communication acts selfishly, seeking the corresponding shortest path, unsolvable overlaps between communications emerge at lower x values. Both algorithms show an increased failure rate as the system size increases, presumably due to the unscaled limit on the number of iterations allowed and possibly inherent finite-size effects.

4.1. A posteriori statistics: maximum cluster size and degree distribution (regular graph case).

To better understand the optimization process and characterize the solutions obtained we carried out a statistical analysis of the solution a posteriori. Given the clearer statistical interpretation of the results obtained (due to the limited number of possible connectivity values and their evolution, and the higher frustration threshold for a given connectivity), we chose to study regular graphs. In this case one can gain more insights into the type of routes formed and the reduced effective graphs that emerge for any number of communications. By a posteriori we mean that once a solution was found, by an MP or greedy algorithm, we removed from the

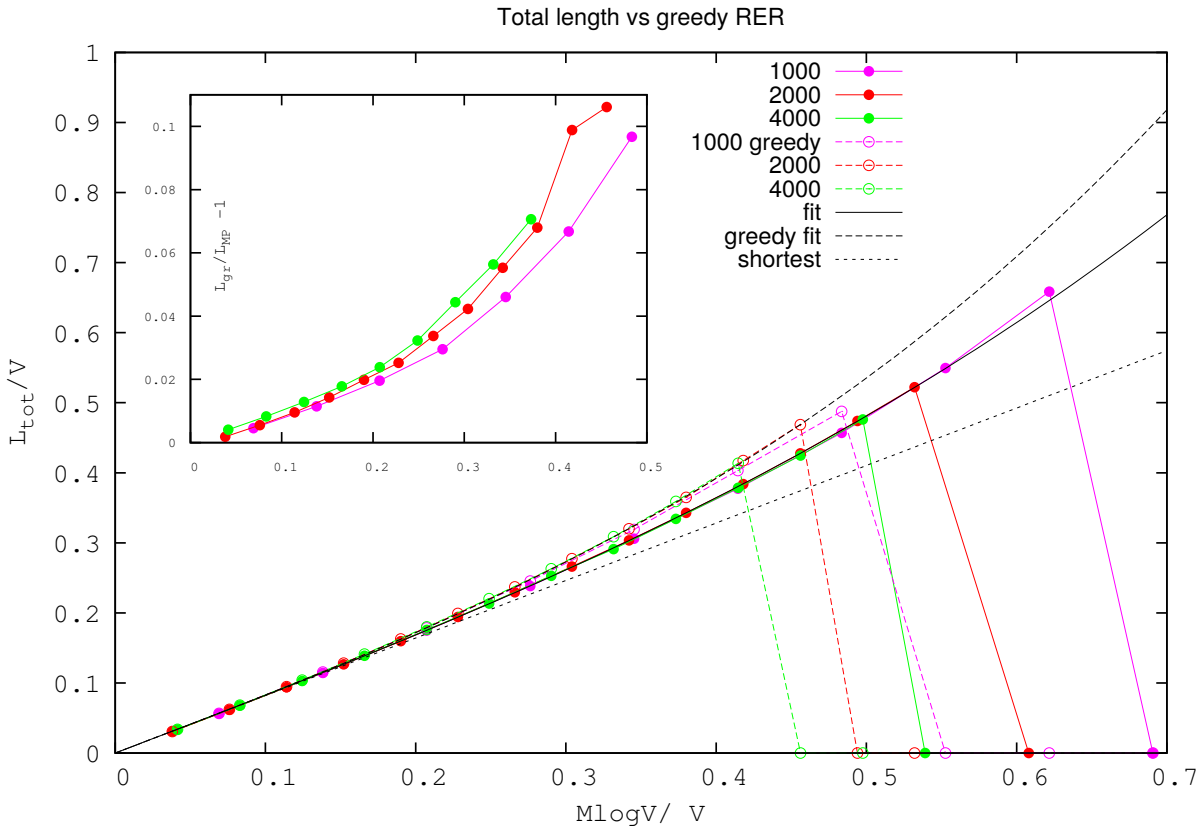


Figure 9: Expected total normalized path length - greedy vs global optimization algorithms RER. System sizes $V = 1000, 2000, 4000$ and degree $\langle k \rangle = 4$. Inset: the ratio $L_{greedy}/L_{MP} - 1$ is plotted as a function of x. Notice the worse performance of the greedy.

graph $\mathcal{G}$ all nodes and edges taking part in the paths and then calculated statistical properties of the remaining graph $\mathcal{G}'$. In particular, we calculated the maximum cluster size and the degree distribution.

The existence of a solution to a given set of communications is strictly related to the connectivity of the graph. Each time a solution for a subset of communications is found, edges and nodes involved in the solution paths are effectively removed; and properties of the reduced graph provide information on its ability to accommodate more source-destination pairs and the efficiency of the obtained solution in making use of the topology. Figure 11 shows the max cluster sizes ratio of $\mathcal{G}'$ as a function of the scaling variable x. This quantity is defined as the ratio between the number of nodes in the maximum connected cluster over the number of nodes in the same graph $\mathcal{G}'$, the graph obtained after edge and node removal of the obtained solution paths. For both the greedy and the global MP algorithm we see an abrupt step change at some x value, between a graph that has a giant connected component and a situation where no solution exists, such that we set the ratio to zero by convention. Moreover, this drop is more abrupt and occurs for smaller x values in the case of the greedy algorithm. This means that the greedy procedure does not distribute paths evenly on the graph and creates small disconnected clusters; the greedy algorithm is therefore more sensitive to

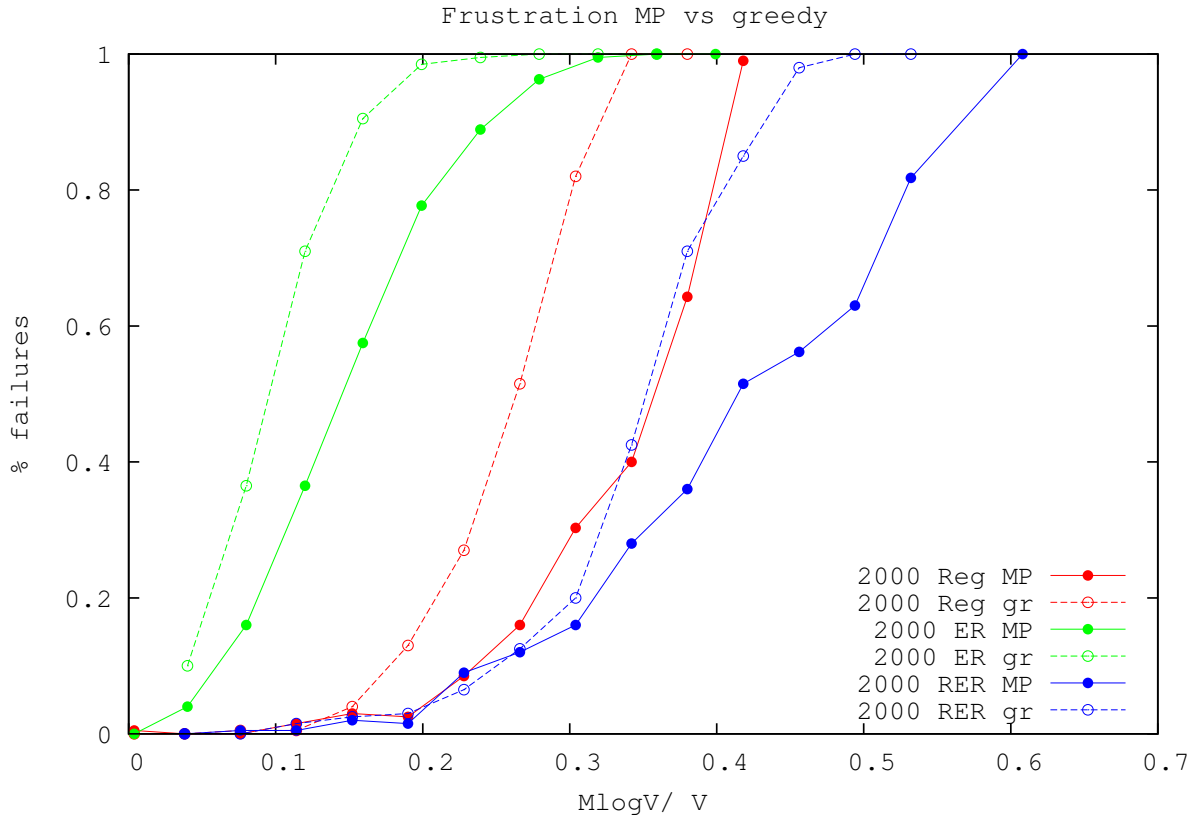


Figure 10: Failure rate for greedy and MP algorithms. We plotted for $V = 2000$ and for Reg and ER of degree $\langle k \rangle = 3$ and RER of $\langle k \rangle = 4$, the failure rate as a function of $\frac{M \log V}{V}$. We can notice that the greedy algorithm fails to find solution earlier than MP for all types of graph considered. ER reaches frustration sooner due to less path choices, whereas RER has higher frustration threshold because of the higher connectivity. The MP data shown are results of averages calculated over a smaller number of instances than for the greedy algorithm, hence the lines connecting them are less smooth.

small changes in connectivity compared to the global MP algorithm, for which the drop is more gradual at first and occurs at higher x values. This reconfirms the previous results that the greedy behavior is fragile and sensitive to the position of the communication pairs and the order in which they are selected.

We evaluate the a posteriori degree distribution $P(k)$ by calculating the connectivities $k_i \, \forall i \in \mathcal{G}'$ for the different k values, and from these derive the average degree $\langle k \rangle$ as a function of the scaling variable x. Results shown in Figure 12 for different system size and $k = 3$ show consistent trends; starting from a 3−regular graph we end up, close to the frustration transition point and after the node and edge removal, with about 20% of the nodes with $k = 3$, whereas $\sim$ 40% have degree $k = 2$ and $\sim$ 30% have $k = 1$. The decay of $\langle k \rangle / k$ is also plotted for the same process. Also here we see a good data collapse (the different curves can only be distinguished close to frustration).

From graph theory [28] we know that when $\langle k \rangle \sim 1$ the graph is likely disconnected, at least to two giant components; the numerical results show that frustration is reached when

$\langle k\rangle/k$ has value $\sim 50-60\%$, which corresponds to an average degree $\langle k\rangle \sim 1.5-1.8$, still higher than the connectivity threshold 1. This can be explained by the fact that tighter constraints on edge availability are imposed in the case of the NDP problem, resulting in frustration even before the graph disconnects (i.e., disconnection is sufficient but not necessary for frustration). Indeed in our model it is insufficient to have just a good number of available links, but they should also constitute clusters of connected links to accommodate new communication paths. Hence the average connectivity value observed at the frustration point of $\langle k\rangle > 1$.

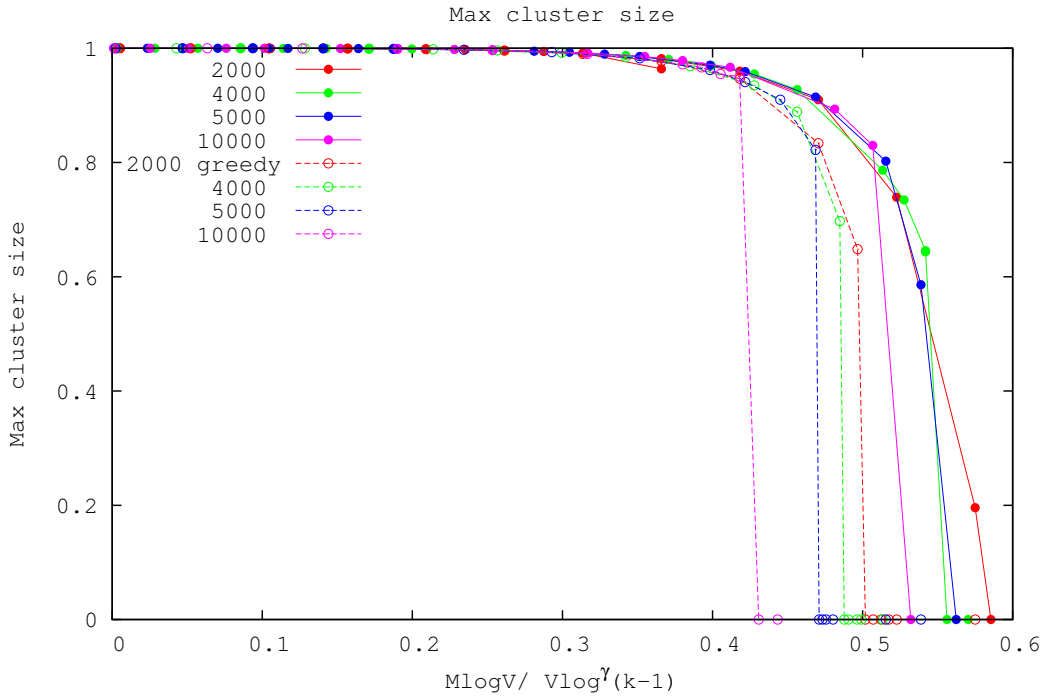


Figure 11: Maximum cluster size greedy vs MP algorithms. The maximum cluster size, i.e. the ratio of giant component size normalized with respect to V, is plotted as a function of the scaling variable x. We see that in both cases frustration is reached for giant component values above 60%; the lower values obtained for smaller graphs result from very few biased successful instances where source-destination pairs (S, R) are selected from different clusters.

4.2. Path length and stretch distribution.

Another interesting quantity to consider is the path length distribution close to the critical threshold, and its comparison with the shortest path distribution. Using the rescaled variable $\frac{L}{\log V/\log^{\gamma}(k-1)}$, where L is the length per communication, we present in figure 13 the distribution obtained for different system sizes. We see a good data collapse for graphs of different system sizes and connectivities to a Gaussian-like distribution with left fat tails, as confirmed by the log-plot on the right panel. This can be explained by the fact that the shorter of the M shortest paths are less likely to be rerouted. A graph with a high number of communications exhibits a path length distribution with higher length averages (with respect to the shortest

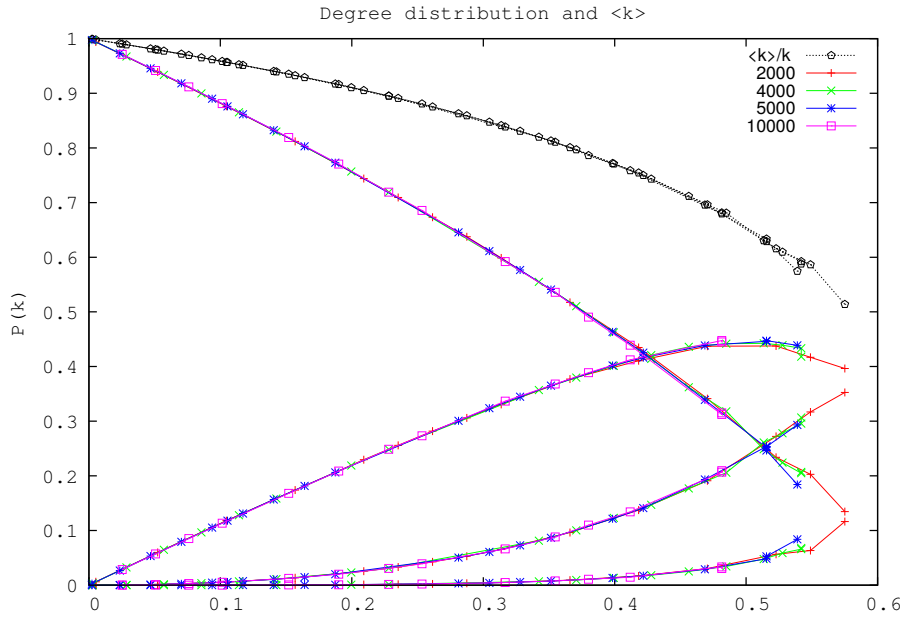


Figure 12: Degree distribution and $\langle k \rangle$. We plotted $P(k)$ for $k = 0, \ldots, 3$ as a function of $x = M \log V / V \log^{\gamma}(k-1)$ for a 3−regular graph and system size $V = 2000, 4000, 5000, 10000$. From top to bottom: the curve at the top is $P(3)$ while the one at the bottom represents $P(0)$. The black line represents $\langle k \rangle / 3$ for the different system sizes.

path distribution) as well as higher variances because solution path lengths are more broadly spread. We notice that the left tails are similar for all connectivity values whereas the right tails are broader for lower connectivities close to the frustration point. This can be explained by the fact that short paths are less likely to be rerouted and occur in roughly the same proportion in graphs of different connectivities; hence the similarity in the fat left tails. Regarding the right tails - many paths are rerouted through longer routes by the MP-algorithm, but graphs with higher degree allow for more communications with shorter routes due to the higher routing flexibility they offer.

Figure 14 shows the stretch, defined as the difference between the shortest path and the path length obtained through MP optimization, for M close to frustration ($\frac{M \log V}{V \log^{\gamma}(k-1)} \sim 0.5$) for different system sizes. We can see that for graphs of degree $k = 3$ only $34 - 38\,\%$ of the communications follow the shortest path, all other communications are routed through longer paths. A higher fraction of shortest-path communications is found for higher connectivity graphs, presumably due to the higher routing flexibility they offer. Looking at the tails we can see that there is a non-negligible fraction of paths that stretch considerably compared to the average shortest path length.

5. Conclusion

We studied the shortest node-disjoint path problem on regular, ER and RER random graphs using message-passing cavity equations. We found that the suggested MP algorithm

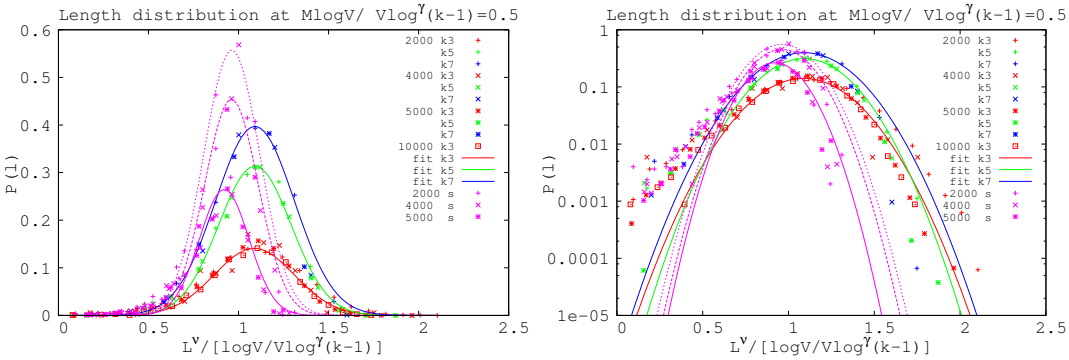


Figure 13: Path length distribution. Left - Path lengths L^ν in the dense interval $M \log V/\log^\gamma(k-1) \sim 0.5$ are normalized and plotted for $k = 3, 5, 7$ graphs against the corresponding shortest paths (violet curves). We see a more broadly spread distributions with higher averages for all graphs with respect to the shortest paths, signaling the path rerouting due to the MP optimization. Right - path lengths are plotted in log scale to highlight the left fat tails where the shortest paths are similar irrespective of connectivity, whereas other paths are rerouted by the MP algorithm and are considerably longer, almost by a factor of two (right tails).

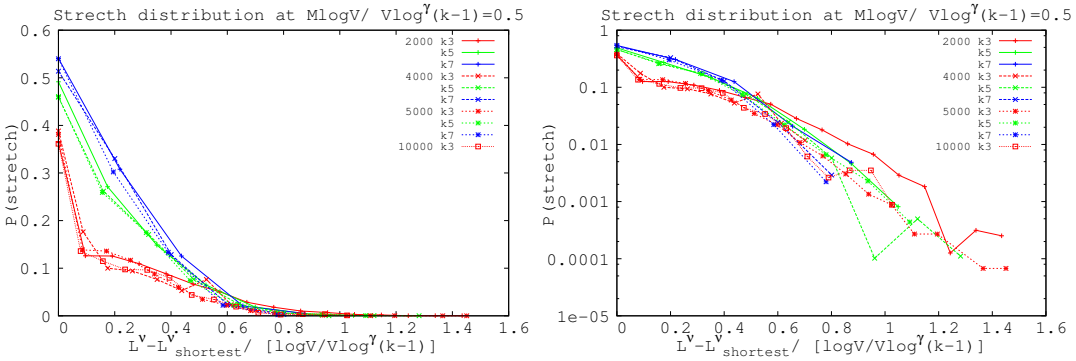


Figure 14: Stretch distribution. Left - The difference $L^\nu - L^\nu_{shortest}$ is normalized and plotted for each communication in the dense regime for degrees $k = 3, 5, 7$. For $k = 3$ we have 40% of communications correspond to the shortest paths while others take routes which are long compared to the average path length $\log V/\log^\gamma(k-1)$. In the case of higher connectivities $k = 5, 7$ there is a higher fraction of paths with the same length as the shortest path due to the routing flexibility offered. Right - The same data is plotted on a log scale to highlight the tails behavior.

outperforms the greedy breadth-first search approach both in finding better solutions (shorter total path length) and in finding solutions for higher values of M. This shows that a global strategy is needed to optimally route paths which do not overlap at nodes but also have minimal path lengths. We found a scaling rule for the total length that goes as a cubic function of the occupancy ratio $\frac{M \log V}{V \log^\gamma(k-1)}$, with γ varying with the graph topology. This behavior resembles the shortest path length for small M but increases faster than linearly for a higher number of paths.

We also studied statistical properties of physical observables a posteriori in the case of

regular graphs. We found good data collapses for regular graphs of different system sizes and connectivities for quantities such as maximum cluster size, degree distribution and, length and stretch distributions.

We believe this approach is theoretically interesting due to its relevance to hard combinatorial complexity problems but also offers a new direction for solving important practical routing problem in communication, in particular in optical and wireless ad-hoc networks and VLSI design. This study offers the first step for realizing the potential in this new direction.

Acknowledgement

This work is supported by the Marie Curie Training Network NETADIS (FP7, grant 290038), the EU FET FP7 project STAMINA (FP7-265496) and the Royal Society Exchange Grant IE110151. This work is partially supported by Research Grants Council of Hong Kong (605010 and 604512).

References.

[1] R M Karp. *Reducibility among combinatorial problems*. Springer, 1972.

[2] Robertson N and Seymour P D. Graph minors. xiii. the disjoint paths problem. *J. of Comb Th, B*, 63(1):65–110, 1995.

[3] Chatterjee B C, Sarma N, Sahu P P, et al. Review and performance analysis on routing and wavelength assignment approaches for optical networks. *IETE Technical Review*, 30(1):12, 2013.

[4] Chen C and Banerjee S. A new model for optimal routing and wavelength assignment in wavelength division multiplexed optical networks. In *INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proc. IEEE*, volume 1, pages 164–171. IEEE, 1996.

[5] Manohar P, Manjunath D, and Shevgaonkar RK. Routing and wavelength assignment in optical networks from edge disjoint path algorithms. *Comm. Lett., IEEE*, 6(5):211–213, 2002.

[6] Banerjee D and Mukherjee B. A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. *Sel. Ar. in Comm., IEEE*, 14(5):903–908, 1996.

[7] Kolliopoulos S and Stein C. Approximating disjoint-path problems using greedy algorithms and packing integer programs. In Bixby R E, Boyd E A, and Ros-Mercado R Z, editors, *Integer Programming and Combinatorial Optimization*, volume 1412 of *Lecture Notes in Computer Science*, pages 153–168. Springer Berlin Heidelberg, 1998.

[8] Belgacem L, Charon I, and Hudry O. A post-optimization method for the routing and wavelength assignment problem applied to scheduled lightpath demands. *Eur. J. Op. Res.*, 232(2):298–306, 2014.

[9] Skorin-Kapov N. Routing and wavelength assignment in optical networks using bin packing based algorithms. *Eur. J. Op. Res.*, 177(2):1167–1179, 2007.

[10] Blesa M and Blum C. Ant colony optimization for the maximum edge-disjoint paths problem. In *App. Ev. Comp.*, pages 160–169. Springer, 2004.

[11] Storn R and Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. glob. opt.*, 11(4):341–359, 1997.

[12] Sumpter Z, Burson L, Tang B, and Chen X. Maximizing number of satisfiable routing requests in static ad hoc networks. *IEEE GLOBECOM 2013 Conference Proceedings*, 2013.

[13] Srinivas A and Modiano E. Minimum energy disjoint path routing in wireless ad-hoc networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 122–133. ACM, 2003.

[14] Akkaya K and Younis M. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3):325–349, 2005.

[15] Li X and Cuthbert L. Stable node-disjoint multipath routing with low overhead in mobile ad hoc networks. In *Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004.(MASCOTS 2004). Proceedings. The IEEE Computer Society's 12th Annual International Symposium on*, pages 184–191. IEEE, 2004.

[16] Jain S and Das S R. Exploiting path diversity in the link layer in wireless ad hoc networks. *Ad Hoc Networks*, 6(5):805–825, 2008.

[17] Tanenbaum A S. *Computer Networks 4th Edition*. Prentice Hall, 2003.

[18] X et al Masip-Bruin. Research challenges in qos routing. *Computer communications*, 29(5):563–581, 2006.

[19] Aggarwal A, Kleinberg J, and Williamson D P. Node-disjoint paths on the mesh and a new trade-off in vlsi layout. In *Proc. of the twenty-eighth annual ACM symposium on Theory of computing*, pages 585–594. ACM, 1996.

[20] Chekuri C and Ene A. Poly-logarithmic approximation for maximum node disjoint paths with constant congestion. In *SODA*, pages 326–341, 2013.

[21] Erdős P and Rényi A. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.

[22] Mézard M, Parisi G, and Virasoro M A. *Spin glass theory and beyond*, volume 9. World scientific Singapore, 1987.

[23] Mézard M and Montanari A. *Information, physics, and computation*. Oxford University Press, 2009.

[24] Mézard M and Parisi G. The cavity method at zero temperature. *J. Stat. Phys.*, 111(1-2):1–34, 2003.

[25] Yeung C H, Saad D, and Wong KY M. From the physics of interacting polymers to optimizing routes on the london underground. *Proc. Nat. Ac. Sci.*, 110(34):13717–13722, 2013.

[26] Yeung C H and Saad D. Competition for shortest paths on sparse graphs. *Phys. Rev. Lett.* , 108(20):208701, 2012.

[27] Mézard M and Parisi G. The bethe lattice spin glass revisited. *Eur. J. Phys. B*, 20(2):217–233, 2001.

[28] Albert R and Barabási A-L. Statistical mechanics of complex networks. *Rev. Mod. Phys.* , 74(1):47, 2002.

The edge-disjoint path problem on random graphs by message-passing

Fabrizio Altarelli,[1,2,*] Alfredo Braunstein,[1,3,2] Luca Dall'Asta,[1,2] Caterina De Bacco,[4,†] and Silvio Franz[4]

[1] *DISAT and Center for Computational Sciences, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy*
[2] *Collegio Carlo Alberto, Via Real Collegio 30, 10024 Moncalieri, Italy*
[3] *Human Genetics Foundation, Via Nizza 52, 10126 Torino, Italy*
[4] *LPTMS, Centre National de la Recherche Scientifique et Universite Paris-Sud 11, 91405 Orsay Cedex, France.*

We present a message-passing algorithm to solve the edge disjoint path problem (EDP) on graphs incorporating under a unique framework both traffic optimization and path length minimization. The *min-sum* equations for this problem present an exponential computational cost in the number of paths. To overcome this obstacle we propose an efficient implementation by mapping the equations onto a weighted combinatorial matching problem over an auxiliary graph. We perform extensive numerical simulations on random graphs of various types to test the performance both in terms of path length minimization and maximization of the number of accommodated paths. In addition, we test the performance on benchmark instances on various graphs by comparison with state-of-the-art algorithms and results found in the literature. Our message-passing algorithm always outperforms the others in terms of the number of accommodated paths when considering non trivial instances (otherwise it gives the same trivial results). Remarkably, the largest improvement in performance with respect to the other methods employed is found in the case of benchmarks with meshes, where the validity hypothesis behind message-passing is expected to worsen. In these cases, even though the exact message-passing equations do not converge, by introducing a reinforcement parameter to force convergence towards a sub optimal solution, we were able to always outperform the other algorithms with a peak of 27% performance improvement in terms of accommodated paths. On random graphs, we numerically observe two separated regimes: one in which all paths can be accomodated and one in which this is not possible. We also investigate the behaviour of both the number of paths to be accomodated and their minimum total length.

* Presently at Capital Fund Management, 23 Rue de l'Université, 75116 Paris, France
† caterina.de-bacco@lptms.u-psud.fr

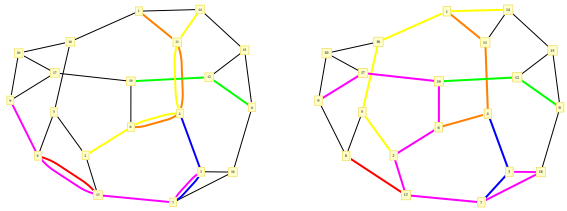


FIG. 1: An instance of the EDP problem over a 3-regular random graph of $V = 20$ and $M = 6$: examples of solutions of the unconstrained (left) and optimal (right) EDP problem are displayed. In the latter, the purple communication is redirected along a longer path to avoid edge-overlap. The yellow one has two shortest paths of equal length (degeneracy) in the unconstrained case, but once EDP is enforced this degeneracy is broken and only one of the two is optimal (right).

I. INTRODUCTION

The optimization of routing and connection requests is one of the main problems faced in traffic engineering and communication networks [1]. The need to deliver Quality of service (QoS) [2, 3] performances, when transmitting data over a network subject to overload and failures, requires both efficient traffic management and resource optimization.

Some aspects of these problems can be formalized using the edge-disjoint path (EDP) problem. This is a constrained optimization problem that is defined as follows. For a given network and a set of communication requests among pairs of users, the EDP consists in finding the maximum number of communications that can be accommodated at the same time, under the constraint that different paths cannot overlap on edges. Moreover, the additional requirement of minimization of the total path length can be considered. Apart from a purely theoretical interest [4], the EDP finds a wide range of applications: in very-large-scale-integration (VLSI) design, in admission control and virtual circuit routing and in all-optical networks. In VLSI design it is required to route wires on a circuit avoiding overlaps, along with minimizing the length of the wires [5, 6]. In admission control and virtual circuit routing [7–9] one needs to reserve in advance a given path for each communication request so that once the communication is established no interruption will occur. This has applications in real-time database servers, large-scale video servers [10–12], streaming data and bandwidth reservation in communication networks [13–16] and in parallel supercomputers. All these applications require high quality data transmission and full bandwidth exploitation. Routing via edge disjoint paths allows for an efficient bandwidth allocation among users because overlap avoidance means full bandwidth exploitation by each single user. An area that has attracted particular attention in the last decade is communication transmission in all-optical networks. Along an optical fiber different communications cannot be assigned the same wavelength to transmit data. Moreover, a unique wavelength must be assigned on all the edges contributing to the path assigned for a given communication. Routing communications under the above two requirements define the problem of routing and wavelength assignments (RWA) in this type of networks [17]. These two constraints suggest that a strategy that iteratively builds edge disjoint paths solutions could allow for a more efficient bandwidth management, namely by using an overall smaller number of wavelengths. This leaves available the remaining ones (according to the edge capacity) to be used either by new users entering the network or by allowing current users to exploit higher bandwidth. This strategy has indeed been applied using greedy [18] and genetic algorithms [19] with performances comparable to other methods based on integer linear programming, graph coloring or bin packing.

The EDP is classified among Karp's NP-hard combinatorial problems [20, 21]. Defining the approximation ratio of a given algorithm as the ratio between the result obtained in term of cost/profit by the algorithm and the optimal one (or viceversa depending on what order gives the maximum ratio), the EDP problem is hard to approximate in the worst case; it has been proved that even an approximation with ratio $O(m^{\frac{1}{2}-\varepsilon})$ is NP-Hard. The best known approximation ratio for the number of accomodated paths is $O(\min\{n^{2/3}, \sqrt{m}\})$ [22, 23] where n and m are the number of nodes and edges in the graph, respectively. Negative results on worst-case inapproximability did not stop progress on heuristc approaches. The problem has been studied intensely with a variety of classical techniques: heuristic greedy algorithms [13, 14, 18, 24], elaborated strategies using bin packing[25], integer/linear programming relaxations [26–29], post-optimization [30], Montecarlo local search [31], genetic algorithms [32–34], particle swarm optimization [35] and ant colony optimization [36], among them.

In this paper we propose a distributed algorithm to solve the EDP problem based on message-passing (MP) techniques (or cavity method) [37]. This method has been extensively employed to address problems in spin glass theory [38–40], combinatorial optimization [41] and more recently in routing problems on networks [42–45]. The evaluation of the equations at the core of the MP technique requires, for each vertex i in the underlying graph, to solve a local combinatorial optimization problem, performing a minimum over a set which is exponentially large in the number of neighbors of i. We propose an efficient method to perform this calculation, by mapping it into a minimum-weight matching problem on a complete auxiliary graph with vertices in the set ∂i of neighbors of i, that can be solved by classical algorithms [46]. With this construction, each iteration of the MP equations can be computed in a time which is polynomial in the number of graph edges (and linear in average for sparse random graphs).

The MP algorithm is tested on computer-generated instances of different classes of random graphs to study the scaling properties with the system size and to compare the performances against a greedy algorithm. We also considered the EDP problem on some benchmark instances found in the literature, for which we could compare the message-passing results with those obtained using other types of algorithms: greedy, ant colony optimization [36] and Montecarlo local search [31].
The paper is organized as follows. In Section II we define the EDP optimization problem, for which we present the message-passing equations in Section III, together with the mapping on a matching problem that simplifies their actual implementation. Section IV reports the results of simulations on random graphs and the scaling of the relevant quantities with the system size, while the comparison between the performances of the message-passing algorithm and other methods is discussed in Section V. Conclusions are given in Section VI.

II. THE EDGE-DISJOINT PATHS PROBLEM

Given a network and a set of M communications requests between pairs of senders and receivers, the standard EDP consists in finding the maximum number of accommodated paths which are mutually edge disjoint. In the applications described in the Introduction, the length of the communication paths is a quantity that, directly or indirectly, affects the overall transmission performances, in terms of transmission delays, infrastructure cost and network robustness. We take into account this aspect by considering the Minimum Weight Edge Disjoint Paths (MWEDP) problem, a generalization of the EDP problem that combines in a unique framework both path length optimization and edge disjointness. An instance of the MWEDP problem is defined by a graph $G(\mathcal{V},\mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ is the set of edges, by an assignment of edge weights w, that we assume to be non-negative real numbers and by a set of M communication requests $\{(S^\mu, R^\mu)\}_{\mu=1,\dots,M}$ between ordered pairs of nodes. We denote by π_μ, a path, i.e. a set of consecutive edges $e \in \mathcal{E}$, that connects a sender S^μ with the corresponding receiver R^μ. The optimization problem consists in finding M pairwise edge-disjoint paths π_μ while minimizing the total edge weight $\sum_\mu w(\pi_\mu)$, where $w(\pi_\mu) = \sum_{e\in\pi_\mu} w(e)$.

The classical EDP problem could be trivially recovered by assigning zero weight to all edges in $G(\mathcal{V},\mathcal{E})$ and a positive cost to each communication that is not accommodated. Alternatively, any solution of the MWEDP problem can be reinterpreted as a solution of the classical EDP problem by slightly modifying the original instance of the graph $G(\mathcal{V},\mathcal{E})$ by introducing an extra edge between each pair (S^μ, R^μ) with sufficiently large cost, such that the algorithm could still always find a solution possibly using these expensive extra edges. By construction, the cost of each of these M extra edges should be larger than the maximum possible weight a single path can take. Then the solution of the classical EDP problem is obtained from any solution of the MWEDP problem by discarding the paths passing through the extra edges. In the present paper, we keep information about path length minimization by assigning unit weights (i.e. $w_{ij} = 1, \forall (ij) \in \mathcal{E}$) to the original edges of the graph $G(\mathcal{V},\mathcal{E})$ and a fixed cost $|\mathcal{E}| + 1$ to the extra edge added between each pair (S^μ, R^μ).

We introduce M-dimensional variables $\bar{I}_{ij} = (I^1_{ij}, \dots, I^M_{ij})$ with entries $I^\mu_{ij} \in \{\pm 1, 0\}$ representing the communication passing along an edge:

$$I^\mu_{ij} = \begin{cases} 1, & \text{if communication } \mu \text{ passes from } i \text{ to } j, \\ -1, & \text{if communication } \mu \text{ passes from } j \text{ to } i, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

We call these vectors *currents* as they must satisfy current conservation at each node i (Kirchhoff law):

$$\sum_{j\in\partial i} I^\mu_{ij} + \Lambda^\mu_i = 0, \quad \forall \mu = 1, \dots, M, \tag{2}$$

where we defined for each node i and each communication μ a variable Λ^μ_i such that

$$\Lambda^\mu_i = \begin{cases} 1 & \text{if } i = S^\mu, \\ -1 & \text{if } i = R^\mu, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The constraint of edge-disjointness specifies that for each edge (ij), at most one of I^μ_{ij} is non-zero, therefore each vector $\bar{I}_{ij}$ can be parametrized by a variable taking $2M + 1$ different values. Notice that the set of variables $\{\bar{I}_{ij}\}_{(ij)\in E}$ completely specifies the state of the network. In this multi-flow formalism, the MWEDP problem is a combinatorial optimization problem in which the global cost function $C(\{\bar{I}_{ij}\}) = \sum_{(ij)\in E} w_{ij} f(\|\bar{I}_{ij}\|)$ depends additively on the total net current $\|\bar{I}_{ij}\| = \sum_\mu |I^\mu_{ij}|$ along the edges, and the edge-disjointness is ensured by defining

$$f(\|\bar{I}_{ij}\|) = \begin{cases} 0, & \text{if } \|\bar{I}_{ij}\| = 0, \\ 1, & \text{if } \|\bar{I}_{ij}\| = 1, \\ +\infty, & \text{if } \|\bar{I}_{ij}\| > 1, \end{cases} \tag{4}$$

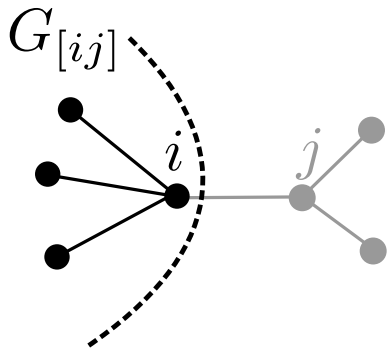


FIG. 2: The modified cavity graph $G_{[ij]}$.

where $|I^{\mu}_{ij}| = 0, 1$ denotes the absolute value of I^{μ}_{ij}. Thus configurations with more than one communication passing along an edge have infinite cost and, in the case of unit weights, the total cost $C(\{\bar{I}_{ij}\})$, if finite, represents exactly the total path length, i.e. the number of edges traversed by paths.

III. THE MESSAGE-PASSING ALGORITHM

On a tree, the optimization problem defined in Sec II can be solved exactly by iteration using the following message-passing algorithm. Let us assume that G is a tree and consider the subtree $G_{[ij]}$ defined by the connected component of i in $G \setminus (ij)$ (see Figure 2). We define $E_{ij}(\bar{I}_{ij})$ to be the minimum cost $C(\{\bar{I}_{ij}\})$ among current configurations that satisfy Kirchoff's laws on all vertices of $G_{[ij]}$ given that we fix an input (or output) extra current $\bar{I}_{ij}$ entering (or exiting) node i. Because of the absence of cycles, it is possible to write a recursive equation for E_{ij} as a sum of cost contributions coming from neighbors of i in the subtree, plus the single cost contribution due to the current $\bar{I}_{ij}$ passing along edge (ij). We call these quantities $E_{ij}(\bar{I}_{ij})$ messages and they verify the *min-sum* recursion relation [39]:

$$E_{ij}(\bar{I}_{ij}) = \min_{\{\bar{I}_{ki}\}|constraint} \left\{ \sum_{k\in\partial i\setminus j} E_{ki}(\bar{I}_{ki}) \right\} + f(||\bar{I}_{ij}||) \tag{5}$$

where *constraint* is the Kirchhoff law at node i and ∂i denotes the neighborhood of i. This relation is exact for trees and can be considered as approximately correct for locally tree-like graphs, such as sparse random ones [37, 39], where correlations between neighbors of a given node decay exponentially. One can develop further this recursion to obtain a set of three types of message-passing equations, one for each type of node, i.e. for each value of Λ^{μ}_{i}. A fixed point of these equations can be found by iteration from arbitrary initial values for the messages until convergence. Then, one can collect at each edge the incoming and the outgoing converged messages to find the optimal configuration $\{\bar{I}^{*}_{ij}\}_{(ij)\in E}$ such that:

$$\bar{I}^{*}_{ij} = arg\, min_{\bar{I}} \left\{ E_{ij}(\bar{I}) + E_{ji}(-\bar{I}) - f(||\bar{I}||) \right\} \tag{6}$$

where the last term is subtracted to avoid double counting of the cost of the single edge (ij).

A. The mapping into a weighted matching problem.

The min-sum algorithm as in (5) presents a computational bottleneck coming from the fact that for each output current $\bar{I}_{ij}$ there is a large number of possible neighborhood's configurations $\{\bar{I}_{ki}\}_{k\in\partial i\setminus j}$ that are consistent both with the edge-disjoint constraints and with Kirchhoff's law. In the calculation of the minimum in (5) one needs in fact to consider all possible combinations of paths entering and exiting node i; the number of such combinations grows exponentially with the degree of node i. Nevertheless, the calculation can be performed efficiently by reducing it to a maximum weight matching problem [46] on an auxiliary weighted complete graph G'_i. The nodes of G'_i are the neighbors $k \in \partial i$ and the (symmetric) weights matrix Q will be defined as

$$Q_{kl} = -\min_{1\le|\nu|\le M} \{E_{ki}(\nu) + E_{li}(-\nu)\} + E_{ki}(0) + E_{li}(0) \tag{7}$$

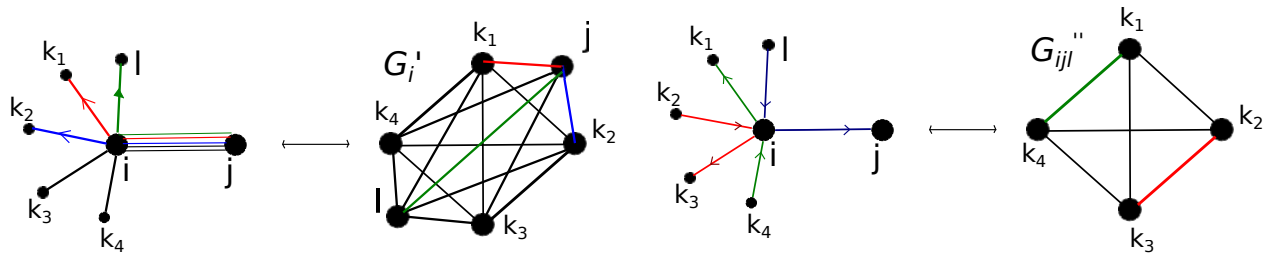


FIG. 3: Mapping into a weighted matching problem. Left: intermediate step where G'_i is built. On the leftmost part we show an example of several communications passing along (ij) and exiting along the remaining neighbors $k \in \partial i \setminus j$. Right: the final step where G''_{ijl} is built; the best configuration around node i when the blue current passes through (ij) is given by the minimum weighted matching on the complete auxiliary graph G''_{ijl}. Edges red and green represent the best matching, i.e. the configuration where two other communications enter/exit neighbors of $i \setminus j$.

where $E_{kl}(\nu) = E_{kl}(\bar{I}_{kl})$ with $I^{\mu}_{kl} \equiv \delta_{\mu,\nu}$ for $\nu > 0$, $I^{\mu}_{kl} \equiv -\delta_{\mu,\nu}$ for $\nu < 0$ and $I^{\mu}_{kl} \equiv 0$ for $\nu = 0$. Notice that this notation maps the M-dimensional vectors $\bar{I}_{ij}$ to the $2M+1$ possible current configurations ν allowed by the edge-disjointness constraint along a given edge. The computation of matrix Q, that requires $O(Mk^2)$ operations, should be performed only once at the beginning of the update routine for node $i \in G$.

Consider now a neighbor $j \in \partial i$ and a given μ passing through edge (ij), we want to update $E_{ij}(\bar{I}_{ij})$. Assuming to know the other vertex $l \in \partial i \setminus j$ where the current μ entering (resp. exiting) node i can exit (resp. enter), then the least costly configuration in the remaining neighborhood is given by

$$q^{\min}_{jl} = -M_{jl} + \sum_{k \in \partial i \setminus \{j,l\}} E_{ki}(0) \tag{8}$$

where M_{jl} is the maximum weight of a matching on a complete graph G''_{ijl} with $k-2$ nodes, built from G'_i by removing nodes j and l (and all their incident edges). Recall that a matching is a subset of edges of G''_{ijl} that do not share any vertex[46]. This is indeed equivalent to assigning to some of the remaining pairs of neighboring nodes currents $\nu \in [-M, \dots, M]$ that enters through one of them and exits through the other, such that the overall cost of the configuration is minimum. The key point is that the matching condition, i.e. the fact that edges in the solution set cannot have a vertex in common, in our problem translates in the condition of forbidding edge overlaps. Hence, thanks to this auxiliary mapping, we are able to reduce the computation of the update rule for the MP equations of the edge disjoint path problem to the solution of a standard (polynomial) combinatorial optimization problem, i.e. maximum weight matching. In Figure 3 we give a diagrammatic representation of the mapping. Note that in the maximum matching problem, edges in the input graph with negative weight can be simply removed. Notice that the neighboring current ν can also be a priori equal to μ in this algorithm, because the configurations where μ appears in more than one pair of edges will be eliminated in the minimization calculation as they have higher cost in our formulation. The minimum weight is thus independent of μ, i.e. of which message we are updating, a fact that allows reducing the complexity of the algorithm by a factor M.

Finally one needs to minimize over l given the matrix $q^{\min}$:

$$E^{t+1}_{ij}(\mu) = \min_{l \in \partial i \setminus j} \left\{ E^{t}_{li}(\mu) + q^{\min}_{jl} \right\} + c_{ij}(\mu) \tag{9}$$

where $c_{ij}(\mu)$ is the cost of edge (ij), that in our case is 0 if $\mu = 0$ and 1 otherwise. We can notice that in order to evaluate each term inside the brackets we need to perform a matching optimization on each of the $(k-2)$-node complete graphs G''_{ijl} built $\forall\, l \in \partial i \setminus j$. Each of these matching routine has complexity $O(k^3 \log k)$ [47] and there are $O(k^2)$ possible combinations of j and l. Reminding that we first need to evaluate the weight matrix Q, the overall complexity of this algorithm will be:

$$O(k^5 \log k + Mk^2)$$

which is polynomial in the variables k and M. Once we have performed this whole procedure, we get all the information we need to calculate the $2M+1$ update messages $E^{t+1}_{ij}(\mu)$, for each $j \in \partial i$, adding a term $O(kM)$ to the final complexity (which is nonetheless negligible compared to the previous two).

The case of $\mu = 0$, in which no current passes through edge (ij) regardless of what happens on the other edges, is addressed by calculating a matching on the $(k-1)$-node complete graph composed of all nodes $l \in \partial i \setminus j$. If i is either a sender or a receiver, i.e. $\Lambda^{\mu}_{i} \in \{\pm 1\}$ for a given $\mu \in [1, \dots, M]$, the same computation can be performed provided that an auxiliary node, indexed by the communication label μ is added to the original graph G and connected to node i, such that its exiting messages will be fixed once at the beginning in the following way and never updated: $E_{\mu i}(\nu) = -\infty$ if $0 < \nu = \mu$ (sender) or $0 < -\nu = \mu$ (receiver), and $E_{\mu i}(\nu) = +\infty$ otherwise.

B. The role of reinforcement.

In order to aid and speed-up convergence of the MP equations, we used a reinforcement technique [48, 49], in which a set of external local fields $h^t_{ij}(\mu) = E^t_{ij}(\mu) + E^t_{ji}(-\mu) - c_{ij}(\mu)$ act on the messages gradually biasing them to align with themselves. The reinforcement is introduced by promoting edge costs to become communication-dependent quantities defined as linear combinations of the cost at the previous time-step and the reinforcement local fields:

$$c^{t+1}_{ij}(\mu) = c^t_{ij}(\mu) + \gamma_t h^t_{ij}(\mu) \tag{10}$$

with $c^0_{ij}(\mu) = c_{ij}$. This cost will then be inserted into equation (9) to replace the term $c_{ij}(\mu)$. This has the effect to lead the messages to converge faster, gradually bootstrapping the system into a simpler one with large external fields. In practice we choose $\gamma_t = t\rho$ and one has to choose the growth rate of γ by tuning the reinforcement parameter ρ, that controls the trade-off between having a faster convergence and reaching a better solution. We tested ρ on instances on three types of graphs to finally choose to fix it to $\rho = 0.002$ in the rest of the simulations. In Figure 4 we could notice that this value achieves comparable results (inset) in terms of M_{acc}/M to lower ρ in less time.
In Figure 5(left) we report the number of converged instances (over 100 realizations) for standard MP (without reinforcement) on four types of random graphs (as described in the next section) and fixed size $V = 1000$ and average degree $\langle k\rangle = 3$. The convergence failure of the standard MP increases considerably with M/V until it reaches a peak value, then it decreases. On the contrary, when reinforcement is used, convergence is always achieved in less than 100 steps (right panel).

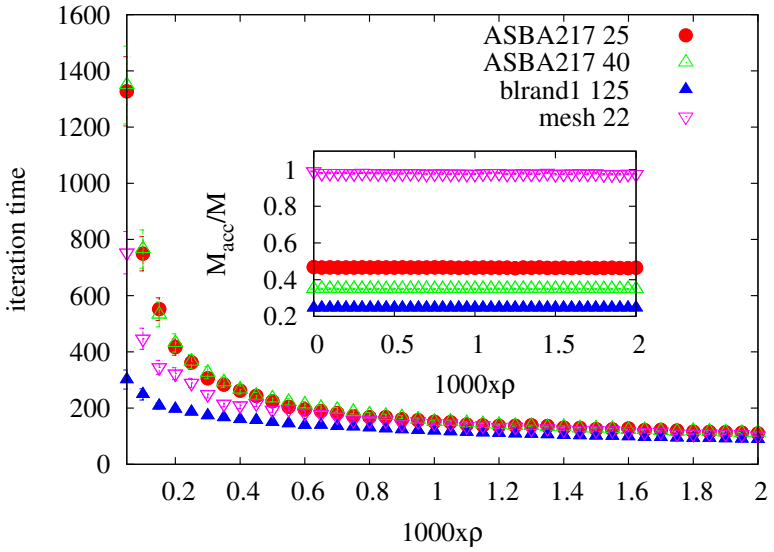


FIG. 4: Reinforcement performance. Number of iterations to reach convergence as a function of reinforcement parameter ρ on BRITE graphs AS-BA217 (V=100) with $M = 25, 40$, blrand1 (V=500) with $M = 125$ and mesh 15x15 with $M = 22$. Inset: the number of accomodated paths M_{acc} is substantially unchanged in the range of parameter values under study.

IV. RESULTS ON RANDOM GRAPHS

First, we tested the MP algorithm on various types of random graphs, with fixed size $V = |\mathcal{V}| = 1000$ and average degree $\langle k\rangle = 3, 5, 7$: regular random graphs (Reg), Erdős-Rényi random graphs (ER) [50], random graphs with power-law distribution (SF) [51] and a set of graphs (RER) obtained adding edges independently with probability p starting from a k_0-regular random graph (for large V, the final average degree of such graphs is $\langle k\rangle = k_0 + d$, with $d = pV$). We compared the performance with a multi-start greedy algorithm (MSG) [36]. This heuristic algorithm calculates paths by iteratively choosing a (random) communication μ, finding the corresponding shortest path and removing the edges belonging to the path from the graph. The process is repeated until either there are no paths left to be routed or no communications can be accommodated anymore in the graph. The multi-start version repeats the same procedure a given number of times and keeps the best solution in terms of M_{acc}, the number of accommodated paths. A bounded-length version [52] of MSG has been used to develop an iterative algorithm to solve the RWA using EDP in [18]: its performance was comparable to the one obtained using a linear programming

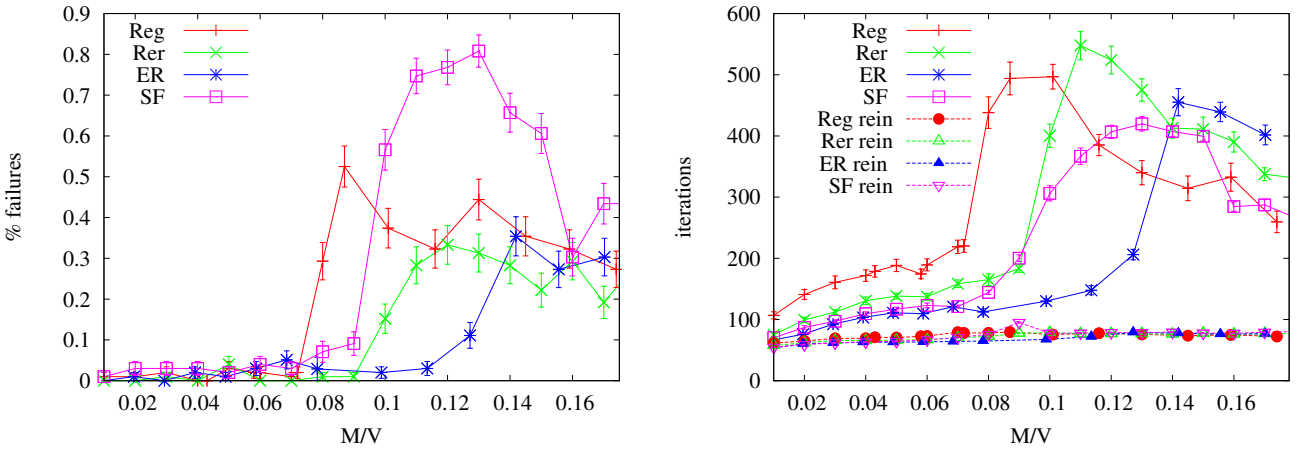


FIG. 5: Left: Fraction of instances in which convergence standard MP fails (reinforced MP always converged in our experiments). Right: number of iterations for convergence for standard MP and reinforced MP ($\rho = 0.002$) in case of random graphs of $V = 1000$ and $\langle k \rangle = 3$ as a function of M/V. Notice how the reinforcement term, besides ensuring convergence, greatly improves the convergence time.

solver on graphs of small sizes ($V \leq 40$) but with faster execution times. This makes it suited to be tested on larger graphs. A disadvantage of the greedy method is that it relies heavily on the order in which communications are accommodated (it disregards the information about sender-receiver pairs other than the ones already accommodated). The difference in the performances of the message-passing and greedy algorithm could then be used to assess the relevance of local information usage in such optimization problem. We tested both the standard multi-start and the bounded-length version but we found equal results with the first being slightly faster, in our tests, in terms of execution times. Thus we decided to use the standard MSG in our simulations. First we compared the results in terms of number of accommodated paths M_{acc} by calculating the ratio M_{acc}/M. In Figure 6 we show the behavior of M_{acc}/M for each type of random graph and $V = 10^3$, $\langle k \rangle = 3$ using MP, reinforced MP and MSG. Both MP versions perform better than MSG, with the standard MP giving better results. The corresponding results for $\langle k \rangle = 5$ are similar (not reported) but the value $M_{acc}/M < 1$ is reached at higher values of M/V and standard MP and MP with reinforcement give almost always the same solutions. The case $\langle k \rangle = 7$ is not reported because, given the high number of edges, the solutions are often trivial (i.e. $M_{acc}/M = 1$), a part from the case of SF graphs where we have instead $M_{acc}/M < 1$ due to the presence of many small degree nodes. We also studied the total path length as a function of M/V for the solutions, obtained with the different algorithms. We consider the ratio between the total path lengths obtained with greedy and MP for solutions in which the number M_{acc} of accommodated path is the same. In Figure 7 we can see that MP always outperforms the MSG algorithm for all types of graph under study. The results for the SF graph with $\langle k \rangle = 7$ are quite different from the other graphs: both for MP and MSG the ratio departs from 1 at rather small values of M/V, possibly because the maximum number of accommodated paths is limited by the existence of many small degree nodes that act as bottlenecks, preventing the use of many alternative edge-disjoint routes. The scaling behavior of the fraction $1 - M_{acc}/M$ of unaccommodated communications and the average total path length L/V of accommodated paths with the system size in the solutions obtained using the MP algorithm is shown in Figure 8 for regular random graphs and ER random graphs. These quantities are plot as functions of the scaling variable $x = \frac{M \log V}{V}$. Note that when paths do not interact, x is a measure the total path length per site, as the average path lenght is proportional to $\log V$. In the top panels, two regimes are visible: for small x, all communications can be accommodated, whereas at some value x^* the curves for different values of V depart from zero. This behavior can be interpreted as a SAT/UNSAT transition, in analogy with the terminology of constraint-satisfaction problems [41]. The collapse of the curves L/V for different values of V is very good in the region in which all paths can be accommodated. On the contrary, in the UNSAT region, the curves for different sizes do not collapse anymore, though the relative difference between them seems to decrease by increasing the system size, and the curves for the largest graphs analyzed ($V = 8000, 10000$) are almost superimposed. We argue that x is the correct scaling variable in the limit of infinitely large graphs, and the observed mismatch could be due to finite-size effects. The change of slope in the roughly linear behavior of the average total length L/V is motivated by the fact that in the SAT region, all communications can be accommodated at the cost of taking longer paths with respect to those actually accommodated in the UNSAT region.

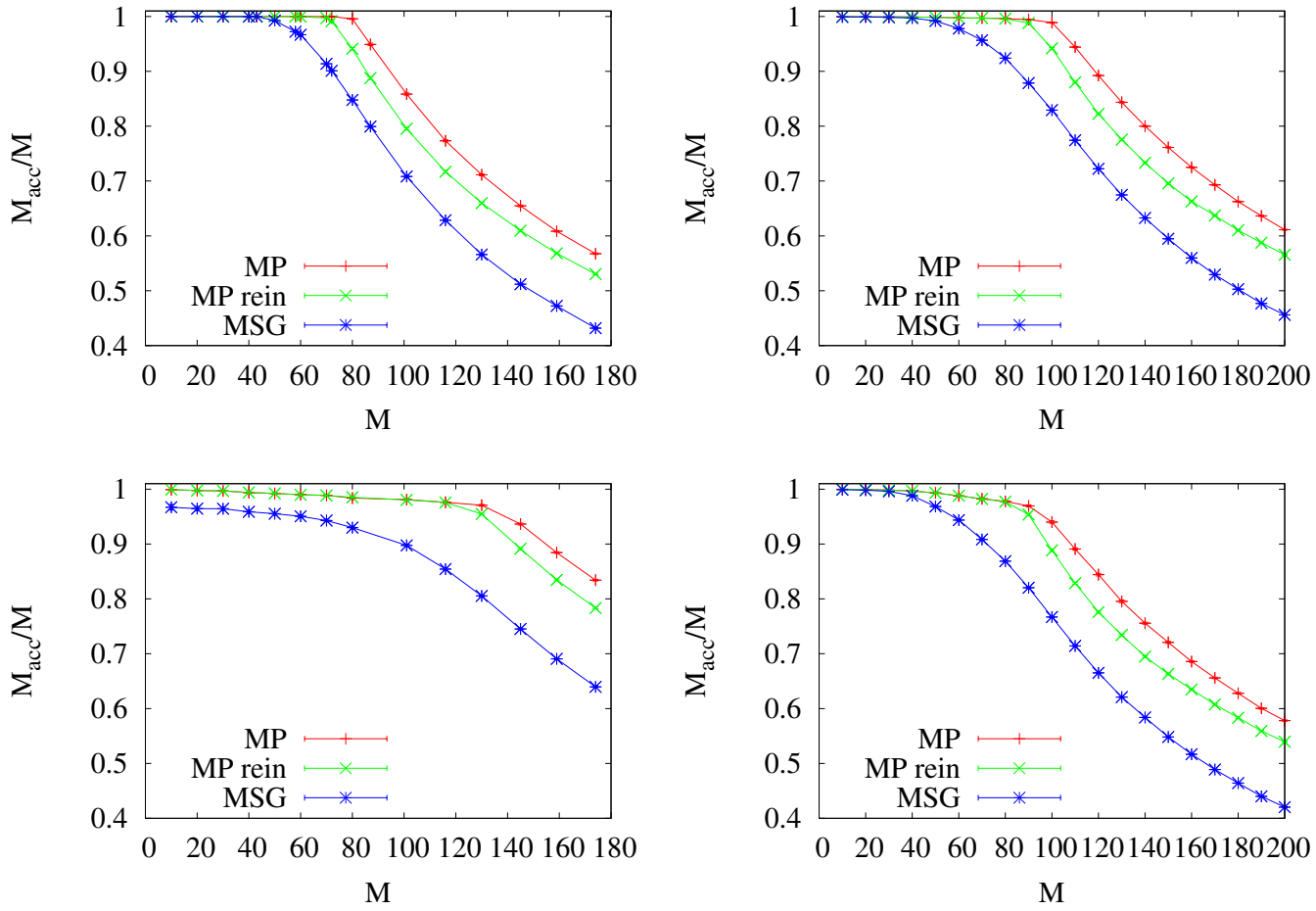


FIG. 6: MP vs greedy performance. We plot the performance in terms of M_{acc}/M for (from top to bottom) regular, RER, ER and SF graphs of fixed size $V = 10^3$ and average degree $\langle k \rangle = 3$. Error bars are smaller than the size of the symbols.

V. COMPARISON WITH OTHER METHODS

A comparison between the performances of the MP algorithm and those of alternative algorithms proposed in the literature [31, 36] is reported in Table I. As benchmark instances we used: two internet-like topologies generated using the BRITE graph generator [53] with parameters set as in [36]; mesh graphs of sizes 15x15 and 25x25, Steiner and planar graphs as reported in [31]. For each of these graphs we used the same set of sender-receiver pairs of size $M = 0.10V, 0.25V, 0.40V$ used in [31]. For each of these instances we ran the MP, MP with reinforcement and MSG algorithms 20 times and collected the average, minimum and maximum number of accommodated paths M_{acc} along with the average computational time in seconds. All results are reported in Table I.

A. Other optimization methods.

A part from the multi-start greedy, we used as comparison two more structured algorithms. The first one is an Ant Colony Optimization metaheuristic [36]. This method builds an EDP solution incrementally from partial solutions provided by a set of M ants. Each ant generates a path for a given communication making probabilistic decisions during the construction steps. These are made by processing local information modeled as *pheromone* information provided by other ants. The advantage of this method is to divide the EDP in subproblems and to use local information. The drawback is that it relies on several parameters that need to be carefully tuned in order to have a sensitive solution. Moreover the computational time increases considerably with the system size. The second algorithm is a Montecarlo-based Local Search [31], that uses as main Montecarlo step a path rewiring based on rooted spanning trees. Unfortunately the running time grows rapidly with the system size, making

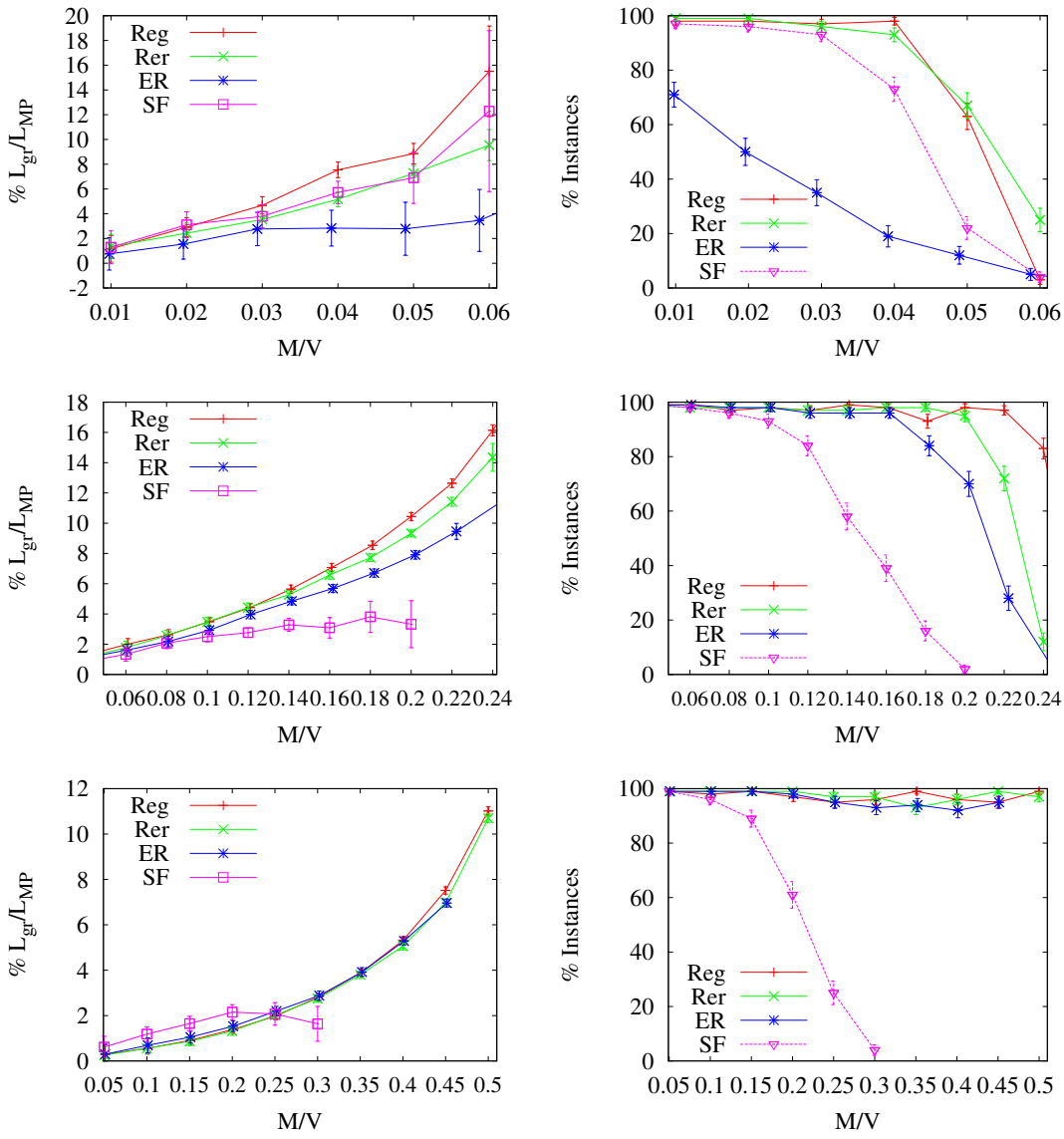


FIG. 7: Length performance. We plot (left) the relative performance of MSG over MP in terms of total length of the solution paths: $y = 100(L_g/L_{MP} - 1)$. Here L_g and L_{MP} denote the total path lengths calculated with MSG and MP respectively. We use Reg, RER, ER and SF graphs of fixed size $V = 10^3$ and average degree $\langle k \rangle = 3, 5, 7$ (from top to bottom). On the right we report the number of instances where the two algorithms find the same solution in term of M_{acc}/M over 100 realizations.

it computationally expensive when used on large graphs. Results are reported in Table I. Finally, we performed simulation using the multi-start greedy heuristic described above.

B. Results.

In Table I we report the performance comparison in terms of M_{acc} between the two versions of MP (with and without reinforcement) and the other 3 types of algorithms. The message-passing always performs equal or better than the other methods. Surprisingly the best performances are given for meshes and planar graphs, where we would expect the failure of MP due to the existence of short loops. What we find instead is that, even though the standard MP converges in few of these instances on meshes, the version with reinforcement always finds a solution that is always better than the other algorithms. The larger performance gap is seen on larger set of commodities and bigger graphs. Performance improvement reaches 27% with respect to LS, the best one between the other algorithms tested. The same considerations can be made in the case of planar graphs. We

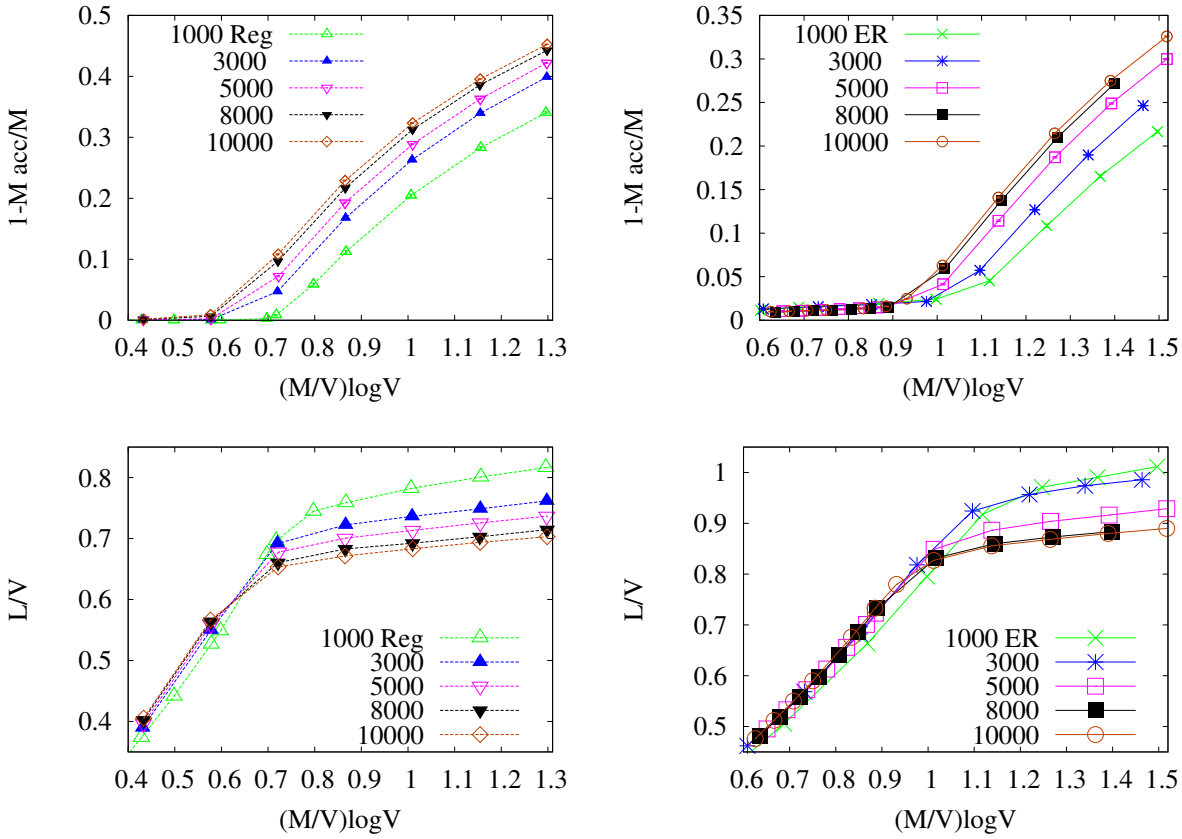


FIG. 8: Finite-size effects. We plot $1 - M_{acc}/M$ (top) and the total length per node L/V (bottom) for Reg (left) snd ER (right) graphs as a function of the scaling variable $\frac{M \log V}{V}$. We can notice the finite-size effects decreasing with system size leading to the curves corresponding to the biggest graphs $V = 8000, 10000$ to almost superimpose. Note that in the SAT phase the total length grows linearly in $\log V$ for all system sizes as expected but in the UNSAT phase the graphs split. Error bars are smaller than point size.

claim that this gap would increase with system size, but unfortunately the size of benchmark graphs remains limited to $V \leq 500$. Moreover these alternative algorithms do not consider path length optimization, thus we cannot compare the performance with respect to this variable. The ACO has been recently tested on several types of graphs (still with $V \leq 500$) against a Genetic Algorithm (GA) in [54]. It performed better than GA in the case of BRITE graphs 1-6 and 14% worse in the case of 10x10 and 15x15 mesh graphs. The MP algorithm always outperforms ACO and in the case of 15x15 mesh the gap reaches 23.5%. Unfortunately neither the GA has been tested on larger graphs nor gives results in terms of path length of the solutions.

VI. CONCLUSIONS

The EDP problem is a combinatorial optimization problem that finds applications in several traffic engineering problems, from VLSI design to routing and access control management in communication networks. In this work we proposed a min-sum message-passing algorithm to find the maximum number of communications M_{acc} that can be accommodated in a network subject to edge-disjoint constraints and minimizing total path length at the same time. We devised an efficient method to implement these equations by exploiting a mapping into a minimum weight matching problem on an auxiliary graph. The standard MP algorithm and the version with reinforcement consistently outperform alternative algorithms found in the literature on different types of benchmark graphs in terms of the fraction M_{acc}/M of accommodated communications. We found two different behaviors: on some "easy" instances, all algorithm accommodate all requests, providing the same results and suggesting that these could be the optimal ones; there are non-trivial instances in which $M_{acc}/M < 1$, but the message-passing algorithm always outperforms the other algorithms in terms of the number of accommodated paths. In particular we obtained better results in the case of meshes and planar graphs, even though these topologies are not locally tree-like as required by the cavity method. In these cases, we could always ensure convergence of the MP equations by exploiting a reinforcement technique. The quality

of solutions improves with decreasing the reinforcement parameter, such that we could always find better solutions than those obtained using the other algorithms under study. Unfortunately, for the heuristic algorithms employed on the benchmarks we could not access other relevant metrics such as the average total path length, as it was not considered before in the literature [31, 36]. Nonetheless we could directly compute such quantity for a multi-start greedy heuristic in several graphs, finding that MP always gives a lower average path length for solutions with the same fraction of accommodated communications.

In conclusion, combining the good performance results, in terms of traffic and path length, with the polynomial time implementation, the use of the MP algorithm opens new perspectives in the solution of relevant routing problems over communication networks such as the RWA in optical networks. In particular, it would be interesting to apply the MP algorithm in the iterative construction of RWA solutions over communication networks with finite link capacity, as it has been done for other types of EDP algorithms.

ACKNOWLEDGEMENT

We thanks P. Quang Dung for kindly providing us the detailed benchmark instances. This work was supported by the Marie Curie Training Network NETADIS (FP7, grant 290038). L.D. also acknowledges the Italian FIRB Project No. RBFR10QUW4.

REFERENCES

[1] Tanenbaum A S. *Computer Networks 4th Edition*. Prentice Hall, 2003.
[2] Gerald R Ash. *Traffic Engineering and QoS Optimization of Integrated Voice & Data Networks*. Morgan Kaufmann, 2006.
[3] X et al Masip-Bruin. Research challenges in qos routing. *Computer communications*, 29(5):563–581, 2006.
[4] Neil Robertson and Paul D Seymour. Graph minors. xiii. the disjoint paths problem. *Journal of combinatorial theory, Series B*, 63(1):65–110, 1995.
[5] Sabih H Gerez. *Algorithms for VLSI design automation*, volume 8. Wiley Chichester, England, 1999.
[6] Wun-Tat Chan, Francis YL Chin, and Hing-Fung Ting. Escaping a grid by edge-disjoint paths. *Algorithmica*, 36(4):343–359, 2003.
[7] Baruch Awerbuch, Rainer Gawlick, Tom Leighton, and Yuval Rabani. On-line admission control and circuit routing for high performance computing and communication. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 412–423. IEEE, 1994.
[8] Mario Gerla and Jack Tzu-Chieh Tsai. Multicluster, mobile, multimedia radio network. *Wireless networks*, 1(3):255–265, 1995.
[9] Akkaya K and Younis M. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3):325–349, 2005.
[10] Kang G Shin and Parameswaran Ramanathan. Real-time computing: A new discipline of computer science and engineering. *Proceedings of the IEEE*, 82(1):6–24, 1994.
[11] Abinash Mahapatra, Kumar Anand, and Dharma P Agrawal. Qos and energy aware routing for real-time traffic in wireless sensor networks. *Computer Communications*, 29(4):437–445, 2006.
[12] Hussein F Salama, Douglas S Reeves, and Yannis Viniotis. Evaluation of multicast routing algorithms for real-time communication on high-speed networks. *Selected Areas in Communications, IEEE Journal on*, 15(3):332–345, 1997.
[13] Sumpter Z, Burson L, Tang B, and Chen X. Maximizing number of satisfiable routing requests in static ad hoc networks. *IEEE GLOBECOM 2013 Conference Proceedings*, 2013.
[14] Srinivas A and Modiano E. Minimum energy disjoint path routing in wireless ad-hoc networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 122–133. ACM, 2003.
[15] Jain S and Das S R. Exploiting path diversity in the link layer in wireless ad hoc networks. *Ad Hoc Networks*, 6(5):805–825, 2008.
[16] X Li and L Cuthbert. Node-disjointness-based multipath routing for mobile ad hoc networks. In *Proceedings of the 1st ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, PE-WASUN '04, pages 23–29. ACM, 2004.
[17] Chatterjee B C, Sarma N, Sahu P P, et al. Review and performance analysis on routing and wavelength assignment approaches for optical networks. *IETE Technical Review*, 30(1):12, 2013.
[18] Manohar P, Manjunath D, and Shevgaonkar RK. Routing and wavelength assignment in optical networks from edge disjoint path algorithms. *Comm. Lett., IEEE*, 6(5):211–213, 2002.
[19] Chia-Chun Hsu, Hsun-Jung Cho, and Shu-Cherng Fang. Routing and wavelength assignment in optical networks from maximum edge-disjoint paths. In *Genetic and Evolutionary Computing*, pages 95–103. Springer, 2014.
[20] R M Karp. *Reducibility among combinatorial problems*. Springer, 1972.
[21] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Series of books in the mathematical sciences. W. H. Freeman, 1979.
[22] Chandra Chekuri and Sanjeev Khanna. Edge disjoint paths revisited. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 628–637. Society for Industrial and Applied Mathematics, 2003.

[23] Thomas Erlebach. Approximation algorithms for edge-disjoint paths and unsplittable flow. In *Efficient Approximation and Online Algorithms*, pages 97–134. Springer, 2006.

[24] Chen C and Banerjee S. A new model for optimal routing and wavelength assignment in wavelength division multiplexed optical networks. In *INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proc. IEEE*, volume 1, pages 164–171. IEEE, 1996.

[25] Skorin-Kapov N. Routing and wavelength assignment in optical networks using bin packing based algorithms. *Eur. J. Op. Res.*, 177(2):1167–1179, 2007.

[26] Banerjee D and Mukherjee B. A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. *Sel. Ar. in Comm., IEEE*, 14(5):903–908, 1996.

[27] Stavros G Kolliopoulos and Clifford Stein. Approximating disjoint-path problems using packing integer programs. *Mathematical Programming*, 99(1):63–87, 2004.

[28] Asuman E Ozdaglar and Dimitri P Bertsekas. Routing and wavelength assignment in optical networks. *IEEE/ACM Transactions on Networking (TON)*, 11(2):259–272, 2003.

[29] Alok Baveja and Aravind Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. *Mathematics of Operations Research*, 25(2):255–280, 2000.

[30] Belgacem L, Charon I, and Hudry O. A post-optimization method for the routing and wavelength assignment problem applied to scheduled lightpath demands. *Eur. J. Op. Res.*, 232(2):298–306, 2014.

[31] Quang Dung Pham, Yves Deville, and Pascal Van Hentenryck. Ls (graph): a constraint-based local search for constraint optimization on trees and paths. *Constraints*, 17(4):357–408, 2012.

[32] Thiago F Noronha, Mauricio GC Resende, and Celso C Ribeiro. A biased random-key genetic algorithm for routing and wavelength assignment. *Journal of Global Optimization*, 50(3):503–518, 2011.

[33] Mitsuo Gen, Runwei Cheng, and Lin Lin. *Network models and optimization: Multiobjective genetic algorithm approach*. Springer, 2008.

[34] Storn R and Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. glob. opt.*, 11(4):341–359, 1997.

[35] Ali Hassan, Chris Phillips, and Jonathan Pitts. Dynamic routing and wavelength assignment using hybrid particle swarm optimization for wdm networks. In *EPSRC PostGraduate Network Symposium (PGNet)*, 2007.

[36] Blesa M and Blum C. Ant colony optimization for the maximum edge-disjoint paths problem. In *App. Ev. Comp.*, pages 160–169. Springer, 2004.

[37] Mézard M and Parisi G. The cavity method at zero temperature. *J. Stat. Phys.*, 111(1-2):1–34, 2003.

[38] Mézard M, Parisi G, and Virasoro M A. *Spin glass theory and beyond*, volume 9. World scientific Singapore, 1987.

[39] Mézard M and Montanari A. *Information, physics, and computation*. Oxford University Press, 2009.

[40] Mézard M and Parisi G. The bethe lattice spin glass revisited. *Eur. J. Phys B*, 20(2):217–233, 2001.

[41] Marc Mézard, Giorgio Parisi, and Riccardo Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, 2002.

[42] Yeung C H and Saad D. Competition for shortest paths on sparse graphs. *Phys. Rev. Lett.*, 108(20):208701, 2012.

[43] Yeung C H, Saad D, and Wong KY M. From the physics of interacting polymers to optimizing routes on the london underground. *Proc. Nat. Ac. Sci.*, 110(34):13717–13722, 2013.

[44] M Bayati, C Borgs, Alfredo Braunstein, J Chayes, Abolfazl Ramezanpour, and Riccardo Zecchina. Statistical mechanics of steiner trees. *Physical review letters*, 101(3):037208, 2008.

[45] C. De Bacco, S. Franz, D. Saad, and C. H. Yeung. Shortest node-disjoint paths on random graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(7):P07009, July 2014.

[46] L. Lovász and D. Plummer. *Matching Theory*. AMS Chelsea Publishing Series. American Mathematical Soc., 2009.

[47] Zvi Galil, Silvio Micali, and Harold Gabow. An $o(ev\backslash logv)$ algorithm for finding a maximal weighted matching in general graphs. *SIAM Journal on Computing*, 15(1):120–130, 1986.

[48] Alfredo Braunstein and Riccardo Zecchina. Learning by message passing in networks of discrete synapses. *Physical review letters*, 96(3):030201, 2006.

[49] Fabrizio Altarelli, Alfredo Braunstein, John Realpe-Gomez, and Riccardo Zecchina. Statistical mechanics of budget-constrained auctions. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(07):P07002, 2009.

[50] Erdős P and Rényi A. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–ñ61, 1960.

[51] Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118, 2001.

[52] Jon Michael Kleinberg. *Approximation algorithms for disjoint paths problems*. PhD thesis, Citeseer, 1996.

[53] Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers. Brite: An approach to universal topology generation. In *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on*, pages 346–353. IEEE, 2001. Available at `http://www.cs.bu.edu/brite/`.

[54] Chia-Chun Hsu and Hsun-Jung Cho. A genetic algorithm for the maximum edge-disjoint paths problem. *Neurocomputing*, 2014.

Appendix A: Convergence criterion

Given a decision variable d^t to be calculated at each iteration update step t, an integer variable n and a time step T_{max} we have convergence if, for n consecutive iteration steps, d^t does not change, and we fix a maximum iteration time T_{max} to update MP

equations. Formally this writes:

$$\exists\, t_0 \in [1, T_{max} - n] \quad s.t. \quad d^{t_0+i} = d^{t_0} \quad \forall i = 1, \ldots, n \tag{A1}$$

In our simulations we defined the decision variable as the total difference of the optimal currents (calculated edge by edge) between two consecutive iteration steps :

$$d^t = \sum_{(ij)\in E} [1 - \delta_{\mu_{ij}^t, \mu_{ij}^{t-1}}] \tag{A2}$$

where $\mu_{ij}^t = |\min_{\mu=-M,\ldots,M} \left\{E_{ij}(\mu) + E_{ji}(-\mu) - c_{ij}(\mu)\right\}|$ and convergence is reached when $d^t = 0$ for n consecutive time steps.

Appendix B: Benchmark results

Instance				MP			MP rein = 0.002			MSG (greedy)			ACO			LS			MP gain vs.		
Name	$\|V\|$	$\|E\|$	$\langle k \rangle$	$\langle M \rangle$	M_{min}	M_{max}	$\langle M \rangle$	M_{min}	M_{max}	$\langle M \rangle$	M_{min}	M_{max}	$\langle M \rangle$	M_{min}	M_{max}	$\langle M \rangle$	M_{min}	M_{max}	MSG	ACO	LS
blrand10_1	500	1020	4.08	16.00	16	**16**	16.00	16	**16**	13.65	13	15	14.80	14	**16**	16.00	16	**16**	6.67	0.00	0.00
blrand25_1	500	1020	4.08	32.00	32	**32**	32.00	32	**32**	27.75	26	30	31.85	31	**32**	32.00	32	**32**	6.67	0.00	0.00
blrand40_1	500	1020	4.08	38.00	38	**38**	38.00	38	**38**	33.10	32	35	37.85	37	**38**	37.90	37	**38**	8.57	0.00	0.00
blrand10_2	500	1020	4.08	26.00	26	**26**	25.65	25	**26**	23.85	23	25	25.25	25	**26**	26.00	26	**26**	4.00	0.00	0.00
blrand25_2	500	1020	4.08	35.00	35	**35**	35.00	35	**35**	30.75	29	33	34.75	34	**35**	34.95	34	**35**	6.06	0.00	0.00
blrand40_2	500	1020	4.08	37.00	37	**37**	37.00	37	**37**	32.45	31	34	36.95	36	**37**	36.95	36	**37**	8.82	0.00	0.00
blsdeg10_1	500	1020	4.08	17.00	17	**17**	16.89	15	**17**	14.65	14	16	15.95	15	16	17.00	17	**17**	6.25	6.25	0.00
blsdeg25_1	500	1020	4.08	36.00	36	**36**	36.00	36	**36**	31.55	30	33	35.80	35	**36**	36.00	36	**36**	9.09	0.00	0.00
blsdeg40_1	500	1020	4.08	34.00	34	**34**	34.00	34	**34**	29.00	28	31	33.65	33	**34**	34.00	34	**34**	9.68	0.00	0.00
blsdeg10_2	500	1020	4.08	20.00	20	**20**	19.85	19	**20**	16.90	16	18	19.20	19	**20**	20.00	20	**20**	11.11	0.00	0.00
blsdeg25_2	500	1020	4.08	34.00	34	**34**	34.00	34	**34**	28.45	27	30	32.95	32	**34**	33.90	33	**34**	13.33	0.00	0.00
blsdeg40_2	500	1020	4.08	37.00	37	**37**	37.00	37	**37**	31.75	30	33	36.50	35	**37**	37.00	37	**37**	12.12	0.00	0.00
mesh15_10_1	225	420	3.73	22.00	22	**22**	22.00	22	**22**	20.60	20	**22**	19.65	19	21	21.55	21	**22**	0.00	4.76	0.00
mesh15_25_1	225	420	3.73	36.00	36	**36**	35.10	35	**36**	28.30	27	30	27.70	26	29	32.00	31	33	20.00	24.14	9.09
mesh15_40_1	225	420	3.73	43.00	43	**43**	42.50	42	**43**	30.10	28	32	35.30	32	38	38.80	37	40	34.38	13.16	7.50
mesh15_10_2	225	420	3.73	-	-	-	19.89	19	**20**	19.75	19	**20**	17.50	17	19	19.45	19	**20**	0.00	5.26	0.00
mesh15_25_2	225	420	3.73	35.00	35	**35**	34.70	33	**35**	29.25	29	30	29.20	28	31	33.05	32	34	16.67	12.90	2.94
mesh15_40_2	225	420	3.73	42.00	42	**42**	41.35	41	**42**	29.80	29	32	34.00	33	36	37.60	36	39	31.25	16.67	7.69
mesh25_10_1	625	1200	3.84	-	-	-	47.25	46	**48**	40.70	40	42	32.85	29	36	41.00	39	43	14.29	33.33	11.63
mesh25_25_1	625	1200	3.84	-	-	-	68.30	67	**69**	48.40	47	51	45.00	42	49	55.55	54	59	35.29	40.82	16.95
mesh25_40_1	625	1200	3.84	-	-	-	88.74	88	**90**	54.35	53	58	57.70	53	61	69.30	67	72	55.17	47.54	25.00
mesh25_10_2	625	1200	3.84	-	-	-	44.33	43	**46**	40.05	38	42	30.10	28	33	37.90	36	40	9.52	39.39	15.00
mesh25_25_2	625	1200	3.84	-	-	-	67.22	65	**70**	48.90	47	52	45.60	44	48	54.70	52	59	34.62	45.83	18.64
mesh25_40_2	625	1200	3.84	-	-	-	88.55	87	**90**	54.05	51	57	57.75	54	61	68.85	66	71	57.89	47.54	26.76
steinb4_10	50	100	4.00	5.00	5	**5**	5.00	5	**5**	5.00	5	**5**	5.00	5	**5**	5.00	5	**5**	0.00	0.00	0.00
steinb4_25	50	100	4.00	12.00	12	**12**	12.00	12	**12**	12.00	12	**12**	12.00	12	**12**	12.00	12	**12**	0.00	0.00	0.00
steinb4_40	50	100	4.00	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	19.90	19	**20**	0.00	0.00	0.00
steinb10_10	75	150	4.00	7.00	7	**7**	7.00	7	**7**	7.00	7	**7**	7.00	7	**7**	7.00	7	**7**	0.00	0.00	0.00
steinb10_25	75	150	4.00	18.00	18	**18**	18.00	18	**18**	18.00	18	**18**	17.85	17	**18**	18.00	18	**18**	0.00	0.00	0.00
steinb10_40	75	150	4.00	28.00	28	28	27.65	27	**29**	25.10	24	27	24.35	23	26	27.30	27	28	7.41	11.54	3.57
steinb16_10	100	200	4.00	10.00	10	**10**	10.00	10	**10**	10.00	10	**10**	10.00	10	**10**	10.00	10	**10**	0.00	0.00	0.00
steinb16_25	100	200	4.00	25.00	25	**25**	25.00	25	**25**	25.00	25	**25**	24.35	24	**25**	25.00	25	**25**	0.00	0.00	0.00
steinb16_40	100	200	4.00	36.12	36	**37**	36.00	36	36	33.20	32	34	32.45	32	34	35.95	35	**37**	8.82	8.82	0.00
steinc6_10	500	1000	4.00	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	49.10	47	**50**	50.00	50	**50**	0.00	0.00	0.00
steinc6_25	500	1000	4.00	125.00	125	**125**	122.55	121	124	107.50	106	110	89.90	85	94	104.95	102	108	13.64	32.98	15.74
stienc6_40	500	1000	4.00	145.84	144	**147**	140.40	139	142	114.10	112	117	109.80	106	117	121.40	119	125	25.64	25.64	17.60
steincc11_10	500	2500	10.00	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	0.00	0.00	0.00
steinc11_25	500	2500	10.00	125.00	125	**125**	125.00	125	**125**	125.00	125	**125**	123.30	122	**125**	125.00	125	**125**	0.00	0.00	0.00
steinc11_40	500	2500	10.00	200.00	200	**200**	200.00	200	**200**	200.00	200	**200**	194.25	190	198	200.00	200	**200**	0.00	1.01	0.00
steinc16_10	500	12500	50.00	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	0.00	0.00	0.00
steinc16_25	500	12500	50.00	-	-	-	125	125	**125**	125.00	125	**125**	125.00	125	**125**	125.00	125	**125**	0.00	0.00	0.00
steinc16_40	500	12500	50.00	-	-	-	200	200	**200**	200.00	200	**200**	200.00	200	**200**	200.00	200	**200**	0.00	0.00	0.00
plan50_10	50	135	5.40	5.00	5	**5**	5.00	5	**5**	5.00	5	**5**	5.00	5	**5**	5.00	5	**5**	0.00	0.00	0.00
plan50_25	50	135	5.40	12.00	12	**12**	12.00	12	**12**	12.00	12	**12**	12.00	12	**12**	12.00	12	**12**	0.00	0.00	0.00
plan50_40	50	135	5.40	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	19.90	19	**20**	0.00	0.00	0.00
plan100_10	100	285	5.70	10.00	10	**10**	10.00	10	**10**	10.00	10	**10**	10.00	10	**10**	10.00	10	**10**	0.00	0.00	0.00
plan100_25	100	285	5.70	25.00	25	**25**	25.00	25	**25**	25.00	25	**25**	25.00	25	**25**	25.00	25	**25**	0.00	0.00	0.00
plan100_40	100	285	5.70	37.00	37	37	37.05	37	**38**	35.80	35	37	34.00	33	36	36.00	35	37	2.70	5.56	2.70
plan200_10	200	583	5.83	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	20.00	20	**20**	0.00	0.00	0.00
pan200_25	200	583	5.83	-	-	-	48.95	48	**50**	46.50	46	48	41.80	39	43	45.95	45	48	4.17	16.28	4.17
plan200_40	200	583	5.83	-	-	-	60.65	58	**62**	52.95	52	56	49.35	47	51	55.70	54	58	10.71	21.57	6.90
plan500_10	500	1477	5.91	50.00	50	**50**	50.00	50	**50**	50.00	50	**50**	44.95	42	47	50.00	50	**50**	0.00	6.38	0.00
plan500_25	500	1477	5.91	-	-	-	92.29	90	**94**	78.15	76	80	60.95	57	65	78.20	77	80	17.50	44.62	17.50
plan500_40	500	1477	5.91	-	-	-	122.31	119	**124**	92.60	90	95	82.85	78	86	100.15	97	102	30.53	44.19	21.57

TABLE I: Message-passing and multi-start greedy performances. Columns 1-4 give the characteristics of the benchmark. For each algorithm, columns 1-3 represent the average, the minimum and the max number of accommodated paths over 20 runs of a given set of commodity instance respectively. ACO and LS performances are reported in [31, 36]. Performance comparison between MP and the other algorithms is given in the three last columns, representing the performance ratio $100 \cdot (M_{acc}^{BP}/M_{acc}^{alg} - 1)$ where *alg* indicates the algorithm used (MSG, ACO and LS respectively). We use as M_{acc}^{MP} the best one between MP with and without reinforcement.

The average number of distinct sites visited by a random walker on random graphs

Caterina De Bacco[1], Satya N. Majumdar[1] and Peter Sollich[2]

[1] LPTMS, Centre National de la Recherche Scientifique et Université Paris-Sud 11, 91405 Orsay Cedex, France.

[2] King's College London, Department of Mathematics, Strand, London WC2R 2LS, U.K.

E-mail: caterina.de-bacco@lptms.u-psud.fr

Abstract. We study the linear large-n behavior of the average number of distinct sites $S(n)$ visited by a random walker after n steps on a large random graph. An expression for the graph topology–dependent prefactor B in $S(n) = Bn$ is proposed. We use generating function techniques to relate this prefactor to the graph adjacency matrix and then devise message-passing equations to calculate its value. Numerical simulations are performed to evaluate the agreement between the message passing predictions and random walk simulations on random graphs. Scaling with system size and average graph connectivity are also analysed.

1. Introduction.

The average number of distinct sites $S(n)$ visited by a random walker of n steps moving on a graph provides important information about the geometry of the coverage of vertices on the graph. The problem of characterizing this quantity $S(n)$ as a function of time n finds interdisciplinary applications such as in target decay [1] and trapping problems [2] in chemical reactions, in the problem of annealing of point defects in crystals [3], in relaxation problems in disordered systems [4] or in problems of dynamics on the internet [5, 6]. Further studies have characterized the same quantity when multiple walkers are moving together [7, 8].

The problem has been widely studied (in the limit $n \gg 1$) in the case of d-dimensional lattices [9, 10, 11] where a number of independent studies all show that for $d > 3$ the average number of distinct visited sites grows linearly in time as $S(n) = n/W(d)$ with a prefactor $1/W(d)$ dependent on the dimension; whereas in $d = 1, 2$ this growth is slower, with $S(n) = \sqrt{8n/\pi}$ and $S(n) = \pi n/\ln n$, respectively. In the case of Bethe lattices of connectivity k the behaviour is linear again [12], with a prefactor dependent on the lattice connectivity $S(n) = [(k-2)/(k-1)]n$. This problem has been tackled also in the cases of graphs different from lattices or random graphs by using the spectral dimension $\tilde{d}$. Under certain assumptions $S_i(t) \sim t^{\min\{1,\tilde{d}\}}$ for $t \to \infty$ and $\tilde{d} \neq 2$. The quantity $\tilde{d}$ has been calculated for complex types of graphs such as decimable fractals, bundled structures, fractal trees and d-simplex. See [13, 14] for an overview. Nonetheless the determination of the prefactor remains an open questions for these complex types of graphs.

The situation where the underlying topology is a random network has only recently been studied; in particular it has been found that for Scale-Free graphs (SF) [15, 16] (in the time regime $n \gg 1$) one recovers the linear behaviour $S(n) \sim n$ seen in both Bethe lattices and d-dimensional lattices for $d \geq 3$. However, there is very limited information on the prefactor B describing this linear behavior $S(n) = Bn$ on random networks. Indeed all the studies referred to above are based on a scaling ansatz and on the analysis of numerical simulations; neither provides a theoretical framework that fully characterizes the prefactor B to the same extent as has been achieved for lattices. The difficulty in setting up a theoretical model to characterize this prefactor is due to the asymmetry between forward and backward steps during the walk; this asymmetry is induced by the random nature of the graph structure, where nodes have a number of neighbours (degree) that is a random quantity extracted from a probability distribution. In this work we combine a general generating function approach, valid also for lattices, with the cavity formalism [17, 18] that has proved to be useful in a wide range of other problems in statistical physics [19]. We derive an approximate expression for the topology dependent prefactor B that is valid in the thermodynamic limit of large graphs, and for $n \gg 1$. We develop message-passing equations to calculate its value and perform numerical simulations on different graph topologies. Finally we describe the behaviour of $S(n)$ in three different time regimes through scaling considerations. We propose this

framework as an alternative tool to the standard ones used in the case of lattices.
The paper is organized as follows: in section 2 we introduce the general model and the notation used to describe a random walk on random networks. Section 3 sets out the generating function approach to the problem. In section 4 we then adapt it to the particular case of random networks. Our main results are derived using message-passing techniques in section 5, leading to an explicit relation between the topology dependent prefactor and the cavity marginals. In section 6 we present and discuss the results of numerical simulations, including the scaling for finite graphs. We conclude in section 7 with a brief summary and outlook.

2. Random walks on graphs.

Given a random graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with $V = |\mathcal{V}|$ nodes and $E = |\mathcal{E}|$ edges, we denote the neighbourhood of a node $i \in \mathcal{V}$ by ∂i, and its degree, i.e. the number of neighbours, by $k_i = |\partial i|$. An overall characterization of the graph topology is then provided by the distribution of the degrees k_i, which we write as $P(k)$.
Introducing matrix notation we define the graph adjacency matrix A as the matrix with entries

$$a_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The nonzero entries of A then indicate which pairs of nodes are connected by an edge. We do not consider self-loops, thus $a_{ii} = 0$. Throughout we will assume that the graph is singly connected. Should the original random graph have disconnected pieces, we discard all except for the largest connected component.
A random walk on a graph is a path $\gamma = \{v_0, v_1, \ldots, v_n\}$ made up of successive random steps between adjacent nodes v_i on the graph, starting from a given node $v_0 \in \mathcal{V}$. Steps are performed according to a transition probability from a node i to an adjacent node j given by:

$$w_{ij} = \frac{a_{ij}}{k_i} \tag{2}$$

All adjacent neighbours of i then have equal probability of being reached in a step starting from i. In matrix notation we define the transition matrix W as the matrix with entries w_{ij}. Defining also D as the diagonal matrix with entries $\delta_{ij}k_i$, we have the relation:

$$W = D^{-1}A \tag{3}$$

We denote the probability of reaching node j in n steps starting from node i as $G_{ij}(n)$. With these definitions, given an n-step random walk $\gamma = \{v_0, v_1, \ldots, v_n\}$, the probability of reaching node v_n starting from node v_0 along this path is the product:

$$\prod_{i=0,\ldots,n-1} \frac{1}{k_i} = \frac{1}{k_0}\frac{1}{k_1} \times \ldots \times \frac{1}{k_{n-1}} \tag{4}$$

In general, in order to compute $G_{ij}(n)$ one has to consider all possible random walks connecting i to j in n steps. Using the transition matrix W we can write this probability as:

$$G_{ij}(n) = [W^n]_{ij} = [(D^{-1}A)^n]_{ij} \qquad . \tag{5}$$

3. Average number of distinct sites: general results.

We are interested in finding the average number of distinct sites $S_i(n)$ visited by a random walker taking n steps on a graph starting at node i.
In this section we derive general results that are valid for any graph topology, including in particular the case of d-dimensional lattices. We use the formalism of generating functions, a tool that has been used to calculate $S_i(n)$ on lattices [12, 10] as well as other quantities of interest in the study of random walks on networks [13, 20, 1, 21]. We denote by $F_{ij}(n)$ the probability of reaching site j for the first time after n steps for a random walk starting at site i; note that for the case $i = j$ we define "reaching" as "returning to" so that $F_{ii}(0) = 0$. We also define $H_{ij}(n)$ as the probability that site j has been visited at least once in n steps by a random walker starting at site i, and let $q_j(n)$ be the probability that a walker starting at site j does not return to it within n time steps.

With these definitions the average number of distinct sites visited by time n (i.e. after n steps), starting at node i, can be written as:

$$S_i(n) = \sum_{j \in \mathcal{V}} H_{ij}(n) \tag{6}$$

Now if a node j has been visited at least once in a walk of n steps starting at node i, we can call the time of the final visit of the walk $m \leq n$ and by definition the walk then never returns to j in the remaining $n - m$ steps.

Thus we can write the convolution:

$$H_{ij}(n) = \sum_{m=0}^{n} G_{ij}(m) q_j(n-m) \tag{7}$$

The generating function (or z-transform) of a quantity $f(n)$ is defined as $\hat{f}(z) = \sum_{n=0}^{\infty} z^n f(n)$, with $z \in [0, 1)$, and has the property that the z-transform of a convolution is the product of the z-transforms. The z-transform of (7) is then

$$\hat{H}_{ij}(z) = \hat{G}_{ij}(z)\hat{q}_j(z) \tag{8}$$

We now want to write everything in terms of $\hat{G}_{ij}(z)$ and so need to find a relation linking $\hat{q}_j(z)$ to $\hat{G}_{ij}(z)$, which we do via the first passage time probability $F_{jj}(n)$. The probability of returning to node j for the first time after exactly n steps can be written as:

$$q_j(n-1) - q_j(n) = F_{jj}(n) \tag{9}$$

Taking the z-transform of this expression and noting that $q_j(0) = 1$, $\hat{q}_j(z) = \sum_{n=0}^{\infty} z^n q_j(n)$ and $\hat{F}_{jj}(z) = \sum_{n=1}^{\infty} z^n F_{jj}(n)$ we have:

$$z\sum_{n=1}^{\infty} q_j(n-1)z^{n-1} - \sum_{n=1}^{\infty} q_j(n)z^n \qquad = \sum_{n=1}^{\infty} F_{jj}(n)z^n \tag{10}$$

$$z\hat{q}_j(z) - [\hat{q}(z) - 1] = 1 - (1-z)\hat{q}_j(z) = \hat{F}_{jj}(z) \tag{11}$$

Hence:

$$\hat{q}_j(z) = \frac{1 - \hat{F}_{jj}(z)}{1-z} \tag{12}$$

We now relate the generator $G_{jj}(n)$ to the first passage time probability $F_{jj}(n)$. The probability of arriving at node j in n steps starting at the same node j, can be seen as the sum of the probabilities grouped according to how often j is visited overall: we can reach j for the first time after n steps; or a first time at $n_1 < n$ and a second time after another $n - n_1$ steps; or a first time at $n_1 < n$, a second time after another $n_2 - n_1$ steps and a third time after a final $n - n_2$ steps, and so on. Mathematically this can be written as:

$$G_{jj}(n) = F_{jj}(n) + \sum_{n_1=0}^{n} F_{jj}(n_1)F_{jj}(n-n_1) + \sum_{n_2=0}^{n}\sum_{n_1=0}^{n_2} F_{jj}(n_1)F_{jj}(n_2-n_1)F_{jj}(n-n_2) + \dots \tag{13}$$

To make the convolution structure clearer, we have included the extreme values (e.g. $n_1 = 0$ and $n_1 = n$ in the first sum) here even though – because $F_{jj}(0) = 0$ – they do not contribute. Taking the z-transform of both sides one sees that

$$\hat{G}_{jj}(z) = 1 + \hat{F}_{jj}(z) + \hat{F}^2_{jj}(z) + \dots = \frac{1}{1 - \hat{F}_{jj}(z)} \tag{14}$$

Substituting this result into (8) using (12) we obtain:

$$\hat{H}_{ij}(z) = \hat{G}_{ij}(z)\frac{1 - \hat{F}_{jj}(z)}{1-z} = \frac{1}{(1-z)}\frac{\hat{G}_{ij}(z)}{\hat{G}_{jj}(z)} \tag{15}$$

This can now be inserted into (6) to give finally the z-transform of the average number of distinct sites visited starting from site i:

$$\hat{S}_i(z) = \frac{1}{1-z}\sum_{j\in\mathcal{V}}\left[\frac{\hat{G}_{ij}(z)}{\hat{G}_{jj}(z)}\right] \tag{16}$$

One sees that the underlying quantity of central interest for our problem is $\hat{G}_{ij}(z)$. The result of equation (16) is valid in general, i.e. regardless of the graph topology. We note that to understand the large n-behaviour of $S_i(n)$ we need to consider $\hat{S}_i(z)$ near $z = 1$. Specifically, if as expected for $V \to \infty$ we have $S_i(n) = Bn$ for large n, then the z-transform will diverge for $z \to 1$ as $\hat{S}_i(z) = B/(1-z)^2$. To calculate B we thus need to understand the behaviour of $\hat{G}_{ij}(z)$ for $z \to 1$.

4. Average number of distinct sites: random graph results.

In this section we will derive an expression for $G(n)$, the matrix with entries $G_{ij}(n)$, where the dependence on the graph size for large graphs is explicit. Here we will for the first time have to restrict the type of graph: as explained below, we require that the eigenvalue spectrum of A has a nonzero gap.

As we saw in section 2, in the case of random graphs we have $G(n) = W^n = (D^{-1}A)^n$ and hence $\hat{G}(z) = (\mathbb{1} - zD^{-1}A)^{-1}$, which relates the propagator G to the graph topology via the adjacency matrix A.

To transform to a symmetric matrix whose properties are simpler to understand, we rewrite this as

$$\hat{G}(z) = D^{-1/2}\hat{R}(z)D^{+1/2} \tag{17}$$

in terms of the matrix

$$\hat{R}(z) = (\mathbb{1} - zD^{-1/2}AD^{-1/2})^{-1} \tag{18}$$

This matrix is now clearly symmetric, and we can diagonalize it as

$$\hat{R} = P\mathbf{\Lambda}P^T \tag{19}$$

where the matrix P has as columns the eigenvectors of $\hat{R}$ and $\mathbf{\Lambda}$ is a matrix containing the eigenvalues of $\hat{R}$ on the diagonal.
In terms of the normalized adjacency matrix $M = D^{-1/2}AD^{-1/2}$ [22], one has

$$\hat{R}(z) = (\mathbb{1} - zM)^{-1} \tag{20}$$

In the following we use Dirac bra-ket notation [23] to denote the eigenvectors $|u_k\rangle$ of M. If $|u_k\rangle$ is one such eigenvector and λ_k the corresponding eigenvalue, then

$$M\,|u_k\rangle = \lambda_k\,|u_k\rangle \tag{21}$$

and it follows that

$$\hat{R}(z)\,|u_k\rangle = (1 - z\lambda_k)^{-1}\,|u_k\rangle \tag{22}$$

In words, $\hat{R}(z)$ has the same eigenvectors $|u_k\rangle$ as M but with corresponding eigenvalues $1/(1 - z\lambda_k)$.
From spectral graph theory [22] we know that the z-independent matrix M has eigenvalues all lying in the range $[-1, 1]$.

By direct substitution into the eigenvalue equation for M one sees that the vector with entries $u_{1,i} = c\sqrt{k_i}$ is an eigenvector with eigenvalue $\lambda_1 = 1$. The constant c is found from the normalization condition $\langle u_1|u_1\rangle = \sum_{i=1}^{V} u_{1,i}^2 = 1$ as $c^{-1} = \sqrt{V\,\langle k\rangle}$ where $\langle k\rangle = \sum_{j\in V} k_j/V$ is the average degree of the graph. If the graph is singly connected then there are no other eigenvectors with eigenvalue 1, so we can order the eigenvalues as

$$1 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_V \geq -1 \tag{23}$$

(The fact that the eigenvalues lie between -1 and 1 can also be seen from the Perron-Frobenius theorem [24, 25], given that the entries of $|u_1\rangle$ are all positive and $\lambda_1 = 1$.)

Splitting off the contribution from λ_1, we can now write the eigenvector decomposition of $\hat{R}(z)$ as

$$\hat{R}(z) = |u_1\rangle \langle u_1| \frac{1}{1-z} + \sum_{k=2}^{V} |u_k\rangle \langle u_k| \frac{1}{1-z\lambda_k} \tag{24}$$

and clearly the first term will be dominant in the limit $z \to 1$ that we need to consider. With the shorthand

$$C(z) = \sum_{k=2}^{V} |u_k\rangle \langle u_k| \frac{1}{1-z\lambda_k} \tag{25}$$

for the second term, we can then write

$$\hat{R}_{ij}(z) = \frac{\sqrt{k_i k_j}}{V\langle k\rangle} \frac{1}{1-z} + C_{ij}(z) \tag{26}$$

From equation (17) we have $\hat{G}_{ij}(z) = (k_j/k_i)^{1/2} \hat{R}_{ij}(z)$, so the analogous representation for $\hat{G}(z)$ reads

$$\hat{G}_{ij}(z) = \frac{k_j}{V\langle k\rangle} \frac{1}{1-z} + \sqrt{\frac{k_j}{k_i}} C_{ij}(z) \tag{27}$$

We can now substitute these expressions into equation (16) to obtain:

$$\hat{S}_i(z) = \frac{1}{1-z} \sum_{j\in V} \left\{ \frac{k_j}{\hat{R}_{jj}(z) V\langle k\rangle (1-z)} + \frac{\sqrt{\frac{k_j}{k_i}} C_{ij}(z) V\langle k\rangle (1-z)}{k_j + C_{jj}(z) V\langle k\rangle (1-z)} \right\} \tag{28}$$

In the following we will consider first the limit $V \to \infty$ and then the limit $z \to 1$. This order of taking the two limits is important to get physical results, as we explain in more detail below. Note that the denominators in the two terms of (28) are identical but written in two different forms that will make the limit procedure clearer.

The large V-limit is simple to take in (26), giving $\lim_{V\to\infty} \hat{R}_{jj}(z) = C_{jj}(z)$. We are assuming implicitly here that $C(z)$ has a well-defined limit for $V \to \infty$. This requires in particular that λ_2 stays away from 1, i.e. that the spectrum of M has a nonzero gap $1-\lambda_2$ between the leading and first subleading eigenvalue for $V \to \infty$. This is generally true for regular [26, 27], ER [28, 29] and scale-free [30, 28] random graphs, but not for lattices, where the eigenvectors are Fourier modes whose eigenvalue approaches 1 smoothly in the large wavelength (zero wavevector) limit.

In the second term of (28), the first term in the denominator can be neglected for $V \to \infty$ at fixed $z < 1$, giving

$$\lim_{V\to\infty} \hat{S}_i(z) = \frac{1}{1-z} \sum_{j\in V} \left\{ \frac{k_j}{C_{jj}(z) V\langle k\rangle (1-z)} + \frac{\sqrt{k_j} C_{ij}(z)}{\sqrt{k_i} C_{jj}(z)} \right\} \tag{29}$$

Now we take the limit $z \to 1$, in which the second term becomes negligible compared to the first. With the assumption of a nonzero gap, $C_{jj}(z)$ also has a finite limit for $z \to 1$ so that we can define

$$\lim_{z\to 1} \left[\lim_{V\to\infty} \hat{R}_{jj}(z) \right] = \lim_{z\to 1} C_{jj}(z) = R_j \tag{30}$$

and get finally

$$\lim_{V\to\infty} \hat{S}_i(z) = \frac{1}{V\langle k\rangle(1-z)^2} \sum_{j\in V} \frac{k_j}{R_j} \tag{31}$$

as the asymptotic behaviour for $z \to 1$.

This has exactly the $1/(1-z)^2$ divergence we were expecting, and gives us the prefactor of the large n-asymptote of the number of distinct sites visited:

$$\lim_{V\to\infty} S_i(n) = B\, n \tag{32}$$

where

$$B = \frac{1}{V\langle k\rangle} \sum_{j\in V} \frac{k_j}{R_j} \tag{33}$$

We can make three observations. Firstly, if we had inverted the order of taking the limits and fixed V while taking $z \to 1$, then we would have had $\hat{R}_{jj}(z) = k_j/[V\langle k\rangle(1-z)]$ to leading order. The second term in (28) would have disappeared in the limit, so that

$$\hat{S}_i(z) = \frac{1}{1-z} \sum_{j\in V} \frac{k_j}{\hat{R}_{jj}(z) V\langle k\rangle(1-z)} = \frac{1}{1-z} V \tag{34}$$

to leading order near $z = 1$. This $1/(1-z)$ divergence of $\hat{S}_i(z)$ implies $\lim_{n\to\infty} S_i(n) = V$, a result which is clear intuitively: if we keep the graph size finite then in the limit of large times the random walk will cover the entire graph, i.e. visit all nodes at least once.

Secondly, from equation (30) we can see that the information one needs to calculate B resides in the quantities $C_{jj}(z) = \sum_{k=2}^{V} u_{k,j}^2/(1-z\lambda_k)$, where the $u_{k,j}$ are the components of the eigenvectors $|u_k\rangle$ of M and the λ_k the eigenvalues. So knowing the full spectrum of M and the associated eigenvector statistics would in principle solve our problem of determining B. While this is feasible computationally for finite and not too large V, we are not aware of a method that would work in the thermodynamic limit $V \to \infty$.

Thirdly, although the index i appears on the left hand side of equation (32), representing the initial node of the walk, it does not appear on the right. This means that the average number of distinct sites visited in the large n limit does not depend on the starting node, and therefore we can drop the index i from the left hand side of (32). In particular, even for graphs with broad degree distributions such as scale-free graphs, the number of distinct sites visited will be the same whether we start the walk from a hub (a node with high degree) or a dangling end of the graph (a node with degree one) – provided of course n is large enough.

5. The message-passing equations.

From expression (33) we see that, for a given graph, we need to calculate the quantity $\frac{k_j}{R_j}$. Although we know the entries of the inverse $\hat{R}_{ij}^{-1}(z) = \delta_{ij} - z a_{ij}(k_i k_j)^{-1/2}$, it is not straightforward to characterize $\hat{R}_{jj}(z)$. We could find the value R_j either by calculating

$\lim_{z\to 1} C_{jj}(z)$ where $C_{jj}(z) = \sum_{k=2}^{V} u_{k,j}^2/(1-z\lambda_k)$ or by directly inverting the matrix $\hat{R}^{-1}(z) = [\mathbb{1} - zD^{-1/2}AD^{-1/2}]$. Unfortunately both of these two methods are prohibitive computationally, already for individual graphs of large size V and even more so if in addition we want to average the results over an ensemble of random graphs.

Our aim, then, is to find a viable alternative method that will allow us to characterize the value of $\hat{R}_{jj}(z)$, and thus calculate $\lim_{n\to\infty} S(n)$ through (32) and (33). We draw for this on methods that have been deployed in the calculation of sparse random matrix spectra [18]. That a connection to spectral problems should exist is suggested by the fact that $z\hat{R}(z) = (z^{-1}\mathbb{1} - D^{-1/2}AD^{-1/2})^{-1}$: up to a trivial rescaling, $\hat{R}(z)$ has the structure of a resolvent (with parameter z^{-1}) for the random matrix $D^{-1/2}AD^{-1/2}$, and it is from such resolvents that spectral information is normally derived, in an approach that in the statistical physics literature goes back to at least Edwards and Jones [31]. Accordingly the two steps we will need to take mirror closely those used to find resolvents of sparse random matrices in [18]: we first write the $\hat{R}_{jj}(z)$ as variances in a Gaussian distribution with covariance matrix $\hat{R}^{-1}(z)$, and then exploit the fact that this distribution has a graphical model structure to derive cavity equations from which these variances can be found.

5.1. Multivariate Gaussian representation.

The first step is simple: we define a vector of random variables $(x_1, \ldots, x_V)$ and assign to this the zero mean Gaussian distribution

$$P(\bar{x}) \propto e^{-\bar{x}^{\mathrm{T}}\hat{R}^{-1}(z)\bar{x}/2} = e^{-\bar{x}^{T}(\mathbb{1}-zD^{-1/2}AD^{-1/2})\bar{x}/2} \tag{35}$$

The marginal distribution of any component of the vector, obtained by integrating $P(\bar{x})$ over all other components, is then also Gaussian:

$$P(x_j) \propto e^{-x_j^2/(2v_j)} \tag{36}$$

with variance $v_j = \langle x_j^2 \rangle = \hat{R}_{jj}(z)$. Our goal is now to calculate these marginal variances efficiently, i.e. without a full matrix inversion.

The key property of the probability distribution (35) is that it can be written in the form

$$P(\bar{x}) = \prod_{i\in\mathcal{V}} e^{-x_i^2/2} \prod_{(ij)\in\mathcal{E}} e^{zx_ix_j(k_ik_j)^{-1/2}} \tag{37}$$

As this factorizes into contributions associated with the nodes and edges of the underlying graph, it defines what is known as a graphical model [19]. On such a graphical model, marginal distributions can be obtained using message-passing, or cavity, equations.

5.2. Cavity equations.

For completeness, we summarize briefly the derivation of the message-passing equations, also known as sum-product algorithm [19]. We focus on trees, i.e. graphs that do not

contain any loops, where the equations are exact, and leave for later a discussion of the extent to which they apply also to large random graphs. Write generally $\phi_i(x_i)$ for the factor in $P(\bar{x})$ associated with node i and $\psi_{ij}(x_i, x_j)$ as the interaction term between nodes i and j. In our case we have:

$$\psi_{ij}(x_i, x_j) = e^{z x_i x_j (k_i k_j)^{-1/2}} \tag{38}$$

$$\phi_i(x_i) \quad = e^{-x_i^2/2} \tag{39}$$

To calculate the marginal distribution of x_j, we could imagine first removing all edge factors $\psi_{ij}(x_j, x_i)$ from $P(\bar{x})$, where i runs over all neighbours of j. The tree is now split into subtrees rooted at each neighbour i, and one can define the cavity marginal of i, $\nu_{i\to j}(x_i)$ as the marginal that is obtained from a (suitably renormalized) probability distribution containing only the factors from the relevant subtree. To get the marginal of x_j, we now just need to reinstate the missing edge factors as well as the node factor at j and integrate over the values of the nodes that we have not yet marginalized over, namely, the neighbours i:

$$P(x_j) \propto \phi_j(x_j) \prod_{i\in\partial j} \int dx_i \, \psi_{ji}(x_j, x_i) \nu_{i\to j}(x_i) \tag{40}$$

One can call the quantities $\nu_{i\to j}(x_i)$ messages sent from i to j, or cavity marginals: each message tells node j what the marginal of its neighbour i would have been if the edge between them had been severed.

The cavity marginals can now be obtained from an analogous relation. To get $\nu_{i\to j}(x_i)$, one can think of removing all edges connecting i to its neighbours l other than j; note that the edge connecting i to j has already been taken out in the definition of the cavity marginal. This generates independent subtrees rooted at the neighbours l, and the marginals at these nodes are $\nu_{l\to i}(x_l)$. Reinstating removed edge factors and marginalizing over neighbours then yields

$$\nu_{i\to j}(x_i) \propto \phi_i(x_i) \prod_{l\in\partial i\setminus j} \int dx_l \, \psi_{il}(x_i, x_l) \nu_{l\to i}(x_l) \tag{41}$$

On a tree these equations can be solved by e.g. starting at leaf nodes, where simply $\nu_{i\to j}(x_i) \propto \phi_i(x_i)$, and then sweeping through the tree in a way that calculates each message once messages have been received from all neighbours except the intended recipient of the message. Note that two messages are needed per edge, one in each direction. Once all messages have been found, the marginals can be deduced from (40).

On graphs with loops, the message-passing equations (40) and (41) are no longer exact: when we remove all edges around node, its neighbours may then still be correlated because of loops, and we cannot factorize their joint distribution into a product of cavity marginals. The cavity method, also known as Bethe-Peierls approximation [19], consists in neglecting such correlations. The set of equations (41) for the cavity marginals is then viewed as a set of fixed point equations that typically have to be iterated to convergence (see below). Clearly the marginals we deduce in the end are approximate. Nevertheless

the method remains useful for us because we expect the approximation to become exact for random graphs in the limit of large V. The reason is that typical loop lengths diverge (logarithmically) with V, so that the graphs become locally tree-like [19, 32]. The correlations that the cavity method ignores then weaken as V grows, making the approach exact for large V.

Specializing now to our Gaussian graphical model, the cavity marginals must also be Gaussian and we can write them as

$$\nu_{l\to i}(x_l) \propto e^{-x_l^2/(2v_l^{(i)})} \tag{42}$$

which defines the cavity variances $v_l^{(i)}$. Inserting (39) and (38) into the general message passing equation (41) and carrying out the resulting Gaussian integrals gives then

$$v_i^{(j)} = k_i \left(k_i - z^2 \sum_{l\in\partial i\setminus j} \frac{v_l^{(i)}}{k_l} \right)^{-1} \tag{43}$$

while for the full marginals one obtains analogously

$$v_j = k_j \left(k_j - z^2 \sum_{i\in\partial j} \frac{v_i^{(j)}}{k_i} \right)^{-1} \tag{44}$$

These two relations are the direct analogues of Eqs. (11) and (12) in [18].

The variances v_j, when calculated in the limit $z \to 1$, are the quantity of interest for our problem as $v_j = \langle x_j^2 \rangle = R_j$. They are known once the cavity variances have been obtained by solving (43).

In practice we use the rescaled cavity variances

$$m_{i\to j} = \frac{v_i^{(j)}}{k_i} \tag{45}$$

as messages from node i to node j. With this definition and using (43) for $z \to 1$ the cavity equations are:

$$m_{i\to j} = \left(k_i - \sum_{l\in\partial i\setminus j} m_{l\to i} \right)^{-1} \tag{46}$$

We solve these by iteration according to

$$m_{i\to j}^{(t+1)} = \left(k_i - \sum_{l\in\partial i\setminus j} m_{l\to i}^{(t)} \right)^{-1} \tag{47}$$

where t represents a discrete iteration time step.

Starting from a given graph $\mathcal{G}$, a suitably chosen convergence criterion and a maximum iteration time $T_{\max}$, the algorithm then works as following:

(i) Initialize the messages $m_{i\to j}^{(0)}$ randomly.

(ii) Run through all edges (ij) and find for each the updated messages $m_{i\to j}^{(t+1)}$, $m_{j\to i}^{(t+1)}$ from (47).

(iii) Increase t by one.

(iv) Repeat steps 2 and 3 until either convergence is reached or $t = T_{\max}$.

If convergence is reached, i.e. the preset convergence criterion is satisfied, one can collect the results and calculate the variances v_j using (44) and (45):

$$v_j = k_j \left(k_j - \sum_{i\in\partial j} m_{i\to j} \right)^{-1} \tag{48}$$

where $m_{i\to j}$ are the converged messages.

If we identify $v_j = \langle x_j^2 \rangle = R_j$ we can then also express directly the prefactor (33) in the linear asymptote in the number of distinct sites visited, $S(n) = Bn$, as

$$B = \frac{1}{V\langle k\rangle} \sum_{j\in V} \frac{k_j}{v_j} \tag{49}$$

$$= \frac{1}{V\langle k\rangle} \sum_{j\in V} \left(k_j - \sum_{i\in\partial j} m_{i\to j} \right) \tag{50}$$

There is one subtlety here that we have glossed over: the variances v_j are the full marginal variances $\hat{R}_{jj}(z)$, which from (26) have the form $k_j/[V\langle k\rangle(1-z)] + C_{jj}(z)$. In the calculation of B we need $R_j = \lim_{z\to 1} C_{jj}(z)$, where the contribution $\propto (1-z)^{-1}$ has been removed. Where we have taken the limit $z \to 1$ above, we therefore implicitly mean that $1 - z$ needs to lie in the range $1/V \ll 1 - z \ll 1$ where the divergent contribution to $\hat{R}_{jj}(z)$ is still small enough to be neglected compared to $C_{jj}(z)$. That it is then allowable nevertheless to set $z = 1$ directly in the cavity equations that we solve is something that has to be checked numerically: we do indeed always find finite marginals v_j from converged solutions for the cavity marginals. The divergent solution also exists as a separate fixed point, namely the trivial solution $m_{i\to j} \equiv 1$ of (46), but is not accessed in our iterative solution method.

5.3. Regular graph case.

Before going on to numerical results for more general random graph ensembles, we briefly use the expression for the topology dependent prefactor (50) to consider the particular case of a regular graph, i.e. a graph where $\forall i \in \mathcal{V}$ we have $k_i = k$. In the infinite graph size limit the graph is then effectively (up to negligible long loops) a regular tree, where each node is equivalent to all others. The quantities of interest in (46), (48) must then be the same $\forall i \in \mathcal{V}$: we can write $k_i = k$, $v_i^{(j)} = v^{(j)}$, $m_{i\to j} = m$ and $v_j = v$. The fixed point cavity equations (46) thus reduce to:

$$m = \left(k - \sum_{l\in\partial i\setminus j} m \right)^{-1} = [k - (k-1)m]^{-1}$$

We obtain a second order equation in m:

$$m^2(k-1) - mk + 1 = 0 \tag{51}$$

with solutions $m = 1/(k-1)$ or $m = 1$. The first solution is the one we require; the second one is the trivial solution discussed above that gives divergent variances in (48). From $m = 1/(k-1)$ one can find the cavity variances and from there the full variances

$$v = \frac{k-1}{k-2} \tag{52}$$

Substituting into the expressions (49) for the prefactor B we obtain:

$$B = \frac{k-2}{k-1} \tag{53}$$

This result agrees with the one derived for Bethe lattices of connectivity k [12]. This is as expected, given that the cavity method is exact on tree graphs.

6. Simulations.

We performed numerical simulations to test the predictions from our cavity approach for the number of distinct sites visited. We used four types of graph structures: regular random graphs (Reg), Erdős-Rényi (ER) [33], scale-free (SF) using a preferential attachment scheme [16] and a dedicated graph ensemble (RER) where graphs are built starting from a k_0-regular random graph, with edges then added independently with probability p as in the ER model; if $d = pV$ then the final average degree of such a graph is $\langle k \rangle = k_0 + d$ for large V. This graph ensemble thus interpolates between the regular and ER cases and is similar to the one analyzed in [34, 15] with the difference that here we start from a regular graph instead of a ring or a lattice. As for the preferential attachment we used the following procedure: start with a graph of m_0 vertices and introduce sequentially $V - m_0$ new vertices by attaching each of them to m already existing nodes. The probability to pick a certain node i as one of these m neighbors is proportional to its degree, $P(k_i) \sim k_i$; thus high degree nodes will be more likely to be picked and hence they will increase their degree while the graph grows. These scheme leads to a power-law degree distribution $P(k) \sim k^{-\gamma}$ with $\gamma = 2.9 \pm 0.1$ [16]; we empirically observe this value in our simulations. We also tried other generation methods for scale-free graphs that yield different values of γ, but as the results were qualitatively similar to those for preferential attachment we only show the latter as representative for our scale-free graph simulations.

For each of these graph topologies we investigated three fixed sizes $V = 10^3, 10^4, 10^5$ and different average degrees. For ER graphs we only used the giant connected component of each graph sampled, but the average degrees we consider are large enough ($\langle k \rangle \geq 4$) for this to reduce V by at most by 2%. The other types of graph have only one connected component by construction. For each given graph we evaluated the cavity prediction (50) from a converged solution of the cavity equations (46). The iterative solution using (47) converged quickly, in typically around 10 iteration

steps. We used as convergence criterion the following: convergence is reached if $\max_{(ij)\in\mathcal{E}} |m_{i\to j}^{(t+1)} - m_{i\to j}^{(t)}| < \epsilon$ for y consecutive times, where we set $y = 10$ and $\epsilon = 10^{-5}$. The results for B were averaged over $1,000$ different graph instances for $V = 10^3, 10^4$ and 100 instances for the bigger graphs of size $V = 10^5$.

The cavity predictions were compared against direct simulations of unbiased random walks. Each walk starts at a randomly picked vertex and we keep track of the number of distinct visited sites as the walk progresses, with individual steps performed using the transition probabilities $w_{ij} = \frac{a_{ij}}{k_i}$ defined in section 2. We averaged the results over the same graph instances as used to generate the cavity predictions. Note that for each instance of a given graph type, only a single walk was performed starting from a randomly chosen initial site. Note that while the cavity prediction depends only on the topology of each graph, for the direct simulations there is an additional source of randomness arising from the particular random walk trajectory that is obtained on a given graph.

The issue of how the cavity predictions depend on graph size V deserves a brief comment. We argued that the method should become exact in the limit $V \to \infty$, and so a priori should extrapolate our predictions for B to this limit. We found, however, that for our relatively large graph sizes the predictions for different V agreed within the error bars. Thus we did not perform a systematic extrapolation and simply used the predictions for $V = 10^4$, as the largest graph size for which we could obtain a statistically large sample (1000 graph instances) of data. The fact that already $V = 10^3$, our smallest size, is large enough to obtain results that are essentially indistinguishable from those for $V \to \infty$ is consistent with findings from cavity predictions in other contexts, see e.g. [35, 36]. An alternative approach to evaluating the cavity predictions would have been to move from specific graph instances to solving the limiting ($V \to \infty$) integral equations for the distribution of messages across the graph. These equations can be read off more or less directly from the cavity equations, see e.g. [35, 37], or obtained from replica calculations [38] and then solved numerically using population dynamics. Given the good agreement between the predictions for our three different V this approach would be expected to give identical predictions, so we did not pursue it.

6.1. Simulations versus cavity predictions.

Our first task is to verify that the cavity equations do indeed correctly predict the prefactor B for random walks on large graphs. In figure 1 we plot the average number of sites $S(n)$ visited for ER graphs of degree $k = 4$, 7 and 10. We plot $S(n)$ versus Bn, with B the value taken from the cavity predictions, so that the data points should lie on the diagonal $y = x$ if the cavity predictions are accurate. We see in figure 1 that this is indeed the case, for graphs of size $V = 10^4$. Similar levels of agreement are obtained for the other graph ensembles and sizes. The numerical data thus fully support our argument that the cavity predictions will be exact for large V, and show that in fact V does not have to be excessively large to reach good quantitative agreement between the

predictions and direct simulations.

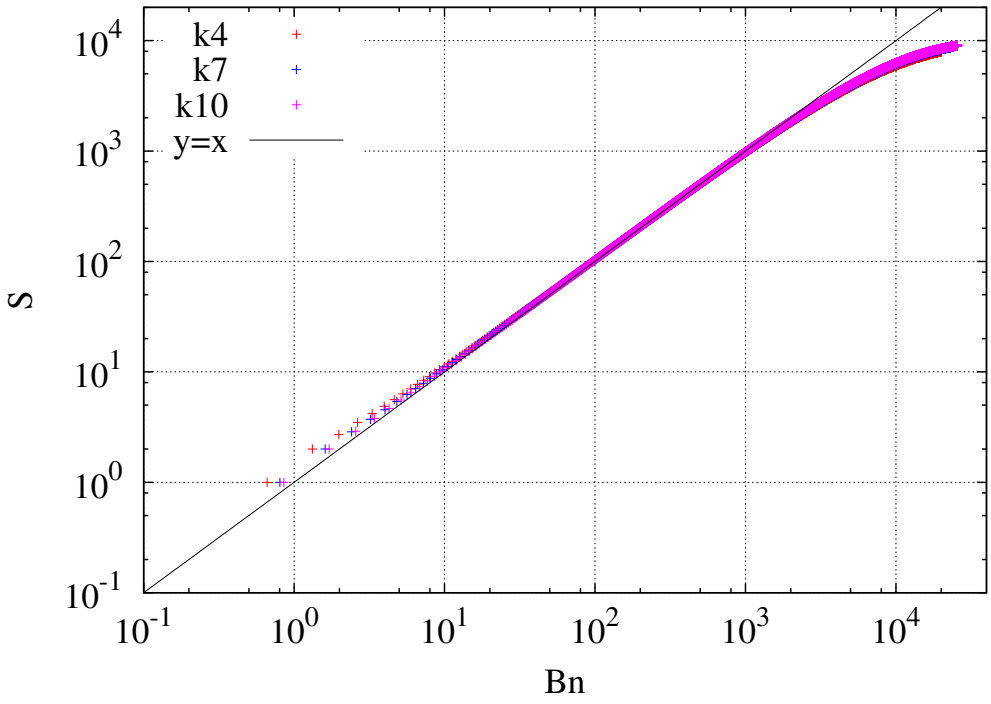


Figure 1: Average number of distinct sites visited, $S(n)$ for random walks on ER graphs of size $V = 10^4$. $S(n)$ is plotted against Bn with the prefactor B as predicted by the cavity method (50), for different average degrees $\langle k\rangle = 4, 7, 10$ as shown in the legend. In the linear regime, before the random walk starts to saturate the graph, data points lie on the diagonal, showing excellent agreement between predictions and direct simulations.

6.2. Dependence on graph topology.

We next look more systematically at how the prefactor B in the large n-behaviour $S(n) = Bn$ depends on the topology of the graphs we study. In figure 2 we report the dependence of the cavity prediction for B on average node degree $\langle k\rangle$, for the four different graph ensembles we studied. We found that for each graph type a hyperbolic fit of the form $B(\langle k\rangle) = \frac{\langle k\rangle - c_1}{\langle k\rangle - c_2}$ gives a good description of the data, with the parameters c_1, c_2 dependent on the graph topology but best fit values always satisfying $c_1 = c_2 + 1$. Thus we could interpret the generic graph result as the one for a regular graph with effective degree $\langle k\rangle - c_2 + 1$. This is intriguing as it suggests that the effect of changing the average degree is quite similar between the different graph types.

Looking at quantitative differences between graph ensembles, we observe that the prefactor B is smallest for given connectivity (average degree) when the graph is regular. Heterogeneity in the node degrees thus generically seems to *increase* the number of distinct sites a random walk will visit, a result that seems to us non-trivial and would be interesting to investigate as a broader conjecture: could there be a lower bound

$B \geq (\langle k \rangle - 2)/(\langle k \rangle - 1)$? If this were the case, one may wonder whether this is related to the spectral gap of a given graph, which is maximal for regular random graphs [39, 40]. Indeed the impact of the gap would appear in the numerator of the prefactor (33) through equation (30) and by using the definition $C_{jj}(z) = \sum_{k=2}^{V} u_{k,j}^2/(1 - z\lambda_k)$. Nonetheless the gap contribution could be balanced off by the square of the eigenvector entries $u_{k,j}^2$ of the matrix R which can be of order $O(1)$ or $O(1/V)$ depending on the eigenvector localization or delocalization respectively. For instance scale-free graphs have been shown empirically to be localized (when considering the adjacency matrix), i.e. only a few eigenvector entries are non-zero and these correspond to the high degree nodes [28], whereas for ER graph the amplitude of the eigenvalue entries is evenly distributed among all the nodes; this difference can be detected for instance by calculating the inverse participation ratio [38, 28]. In order to make a more rigorous statement one would then need to consider these two aspects at the same time but the absence of a general analytical characterization for either the eigenvalues or the eigenvector entries makes this difficult.

One could also ask whether at given $\langle k \rangle$, B is always increasing with some measure of spread of degrees such as the variance $\langle k^2 \rangle - \langle k \rangle^2$. For our admittedly limited choice of graph ensembles it is certainly true that the scale-free graphs (SF), which have the broadest degree distributions, also give the largest B. Below them are the ER graphs. The RER graphs, finally, with their character intermediate between regular and ER, also have prefactors B that lie between those of the ER and regular graphs.

6.3. Finite-size effects and scaling.

We can use our numerical simulation results to enquire also about finite-size effects, describing the behaviour of $S(n)$ on graphs of large but finite size V. Our derivation of B and its prediction using cavity techniques was done taking a large V-limit so cannot make statements about this regime; instead we will have to rely on physical intuition to construct a suitable finite-size scaling ansatz.

From inspection of the numerical simulations, we can distinguish a number of time regimes. Initially $S(n)$ is linear in n with prefactor 1. This is greater than the large n prediction Bn with a prefactor $B < 1$, because the random walker has not yet had much opportunity to return to previous sites; in particular one has, trivially, $S(1) = 1$, ignoring the starting site v_0.

For larger n one finds the predicted linear growth with prefactor $B < 1$, i.e. $S(n) = Bn$. Once Bn becomes comparable to V, a crossover to sublinear growth takes place, and finally $S(n)$ approaches V as the walker visits all sites for asymptotically large n. These regimes, with the exception of the trivial small n-range, can be clearly distinguished in figure 3, which shows results for fixed graph size $V = 10^4$ and graphs with $\langle k \rangle = 4$; plots for other graph sizes and average degrees look qualitatively identical.

A plausible scaling ansatz that encompasses the various regimes – again without

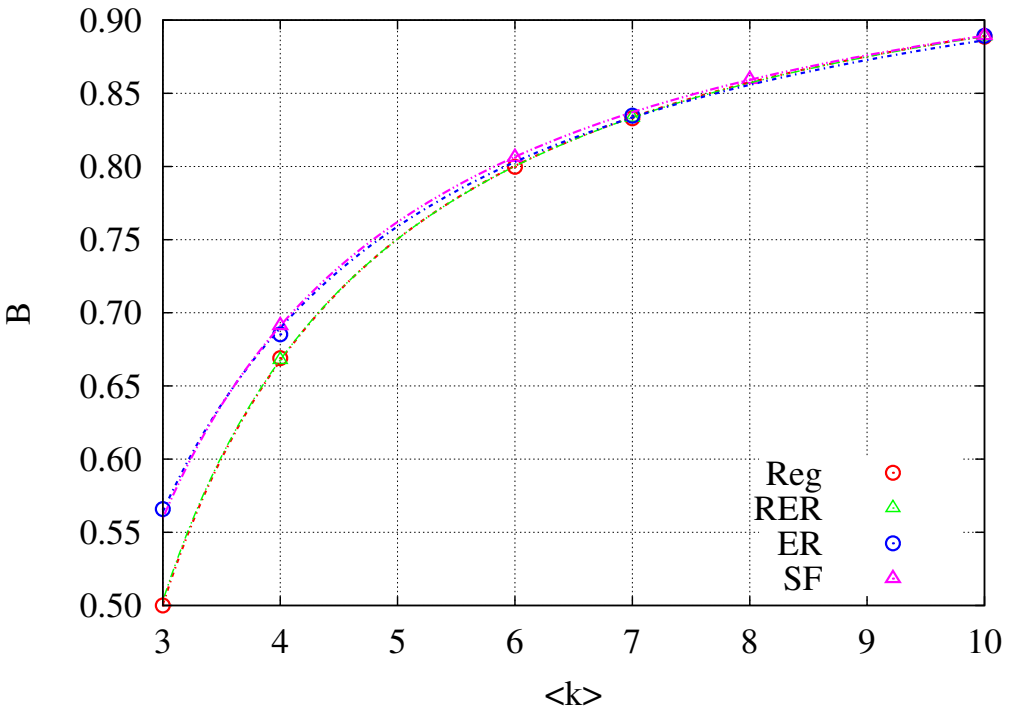


Figure 2: Prefactor B predicted by cavity method as a function of average degree, for different graph types as shown in the legend. The lines represent hyperbolic fits; see text for details. Note that the results for Reg and RER are essentially on top of each other, and the same is true for ER and SF.

the initial small n-piece – is

$$S(n,V) = Bn\, f\left(\frac{Bn}{V}\right) \tag{54}$$

where the limiting behaviour of the scaling function must be

$$f(x) \approx \begin{cases} 1 & x \ll 1 \\ \frac{1}{x} & x \gg 1 \end{cases} \tag{55}$$

to reproduce $S(n,V) \approx Bn$ and $S(n,V) \approx V$ when n is much smaller and much larger than V, respectively.

In figure 4 we check to what extent the finite-size scaling (54) captures our simulation data. We show results for graph sizes $V = 10^3, 10^4, 10^5$ and two values for the average degree $\langle k\rangle = 4, 10$. By plotting $S(n)/(Bn)$ vs Bn/V with B predicted from the cavity equations, we directly have a graphical representation of the scaling function $f(x)$. Very good agreement is seen between the three different graph sizes: these all collapse onto the same curve, except the initial regime discussed above where $S(n) \approx n$ and hence $S(n)/(Bn) > 1$. Beyond this we observe a plateau at $S(n)/(Bn) = 1$, which in a different guise verifies our claim above that the cavity method does indeed predict the prefactor B correctly. For $x = Bn/V$ growing towards unity, the curves drop below

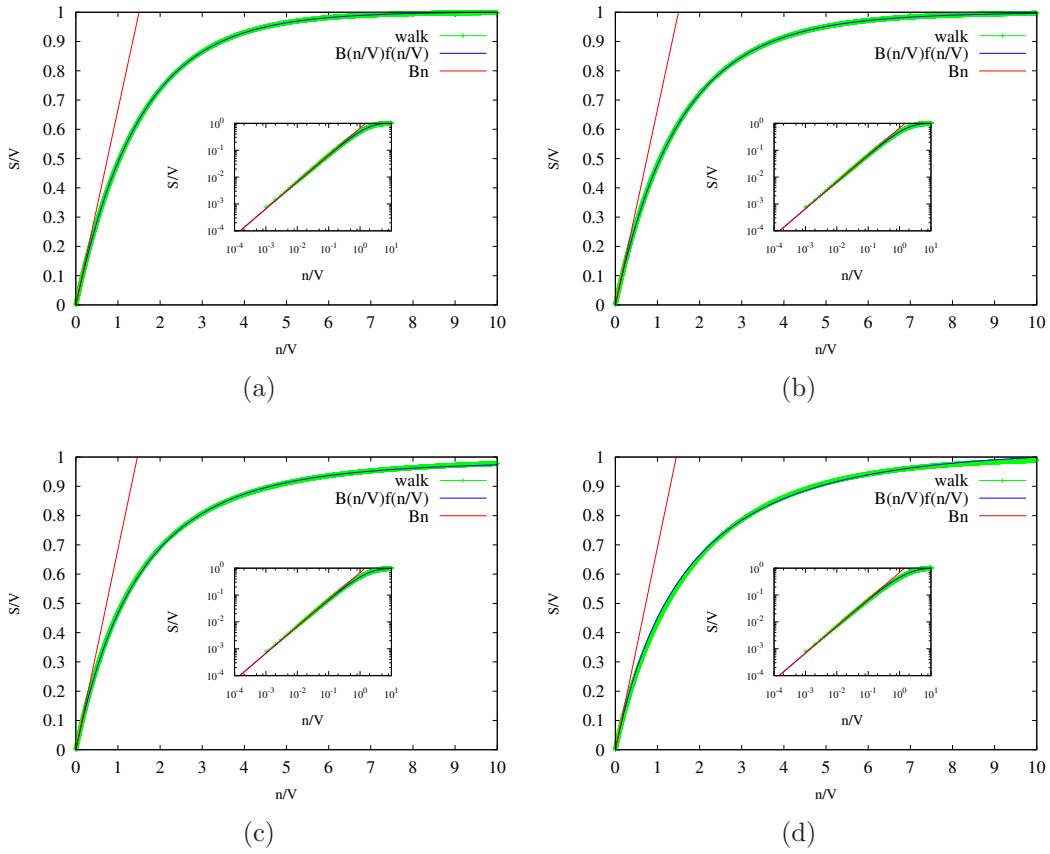


Figure 3: Finite size effects: we show the walker behavior by plotting $S(n)/V$, i.e. the fraction of distinct sites visited, derived from direct simulations vs n/V. Results are from averages over 1000 instances of graphs of fixed size $V = 10^4$ and average degree $\langle k \rangle = 4$, for different graph topologies: a) Regular; b) RER; c) ER; d) SF. The dashed red lines show the cavity predictions Bn for the linear growth with n, a regime which is clearer in the log-log plot insets. Beyond that one observes a slow crossover, with $S(n)/V$ eventually approaching unity. Solid lines show our phenomenological scaling fits.

this plateau as expected, indicating the start of the saturation regime. Asymptotically the scaling function $f(x)$ then approaches $1/x$, reflecting the final saturation of $S(n)$ at the upper bound V.

More surprising, and not required by our ansatz per se, is that we see in figure 4 good collapse also between graphs of different average degree: using Bn/V as the argument of the scaling function seems sufficient to absorb all the variation with $\langle k \rangle$, without further changes in $f(x)$. The only exception is provided by the scale free graphs, which we discuss in more detail below.

Encouraged by the good agreement of the numerical data with the ansatz (54), we attempt to find simple fits to the scaling function $f(x)$. The simulation data show that the crossover starts off with a roughly exponential departure from the small x-plateau

$f(x) \approx 1$, which suggests a scaling function of the form $f(x) = a/\ln(b + (e^a - b)e^{ax})$, where a and b are fitting parameters. Figure 4 shows that this form fits the data extremely well, and except for the scale-free graphs the fits can be performed even with fixed $b = 1$, leaving a single fit parameter.

We comment finally in more detail on the case of SF graphs. Here we see that the data in figure 4 do not collapse perfectly for different V in the intermediate regime where $x = Bn/V$ is order unity or somewhat smaller. In addition, the crossover in $f(x)$ is slower, with $f(x)$ lower in the crossover region than for the other three graph types. We conjecture that both of these effects are due to the presence of many small loops in SF graphs, for example triangles (loops of length 3). To support this hypothesis, we calculated the average number of triangles present in the different types of graph, taking averages over 100 graph instances of size $V = 10^3$. We found results in the same range for Reg, ER and RER graphs, where the average percentage of nodes that are part of at least one triangle does not exceed 2%, 7% and 37% for $\langle k \rangle = 4, 6, 10$ whereas for SF graphs the relevant fractions of nodes reach 9%, 24% and 51% for the same average degrees. These results confirm that SF graphs generated via preferential attachment contain a higher number of short loops than the other topologies. In fact it has been shown by spectral arguments [28] that, even though the fraction of nodes in triangles will tend to zero for $V \to \infty$, the growth rate of the number of loops of length $l \geq 4$ exceeds all polynomial growth rates, thus these graphs do not become locally treelike for large V. Therefore it is somewhat surprising that the cavity predictions for B are quantitatively accurate even for SF graphs.

7. Conclusions.

We have presented an analytical expression for the topology dependent prefactor B governing the linear regime for the average number of distinct sites $S(n)$ visited by a long (large n) random walk on a large random graph. We adapted the general results derived for $S(n)$ in terms of generating functions, as used to study d-dimensional lattices, to the case of random networks. We then combined message-passing techniques and the properties of Gaussian multivariate distributions to derive an expression for B that is valid for locally tree-like graph structures, and found good agreements between the theoretical predictions and direct numerical simulations. An intriguing feature of the results is that at fixed average degree $\langle k \rangle$, B seems smallest for regular graphs and increases with the width of the degree distribution, and one may conjecture that the regular graph result $B = (k-2)/(k-1)$ is in fact a lower bound.

We analysed finite-size effects for $S(n, V)$ and proposed a simple scaling ansatz to capture these. Apart from a trivial small n-regime, one finds a linear regime $S \approx Bn$ with prefactor B in accord with our predictions; an asymptotic regime $Bn \gg V$ where the random walk saturates and $S \to V$; and a crossover in between. Our data provides excellent support for the scaling description, except possibly for SF graphs built via preferential attachment, and we were able to provide a simple two-parameter (in fact

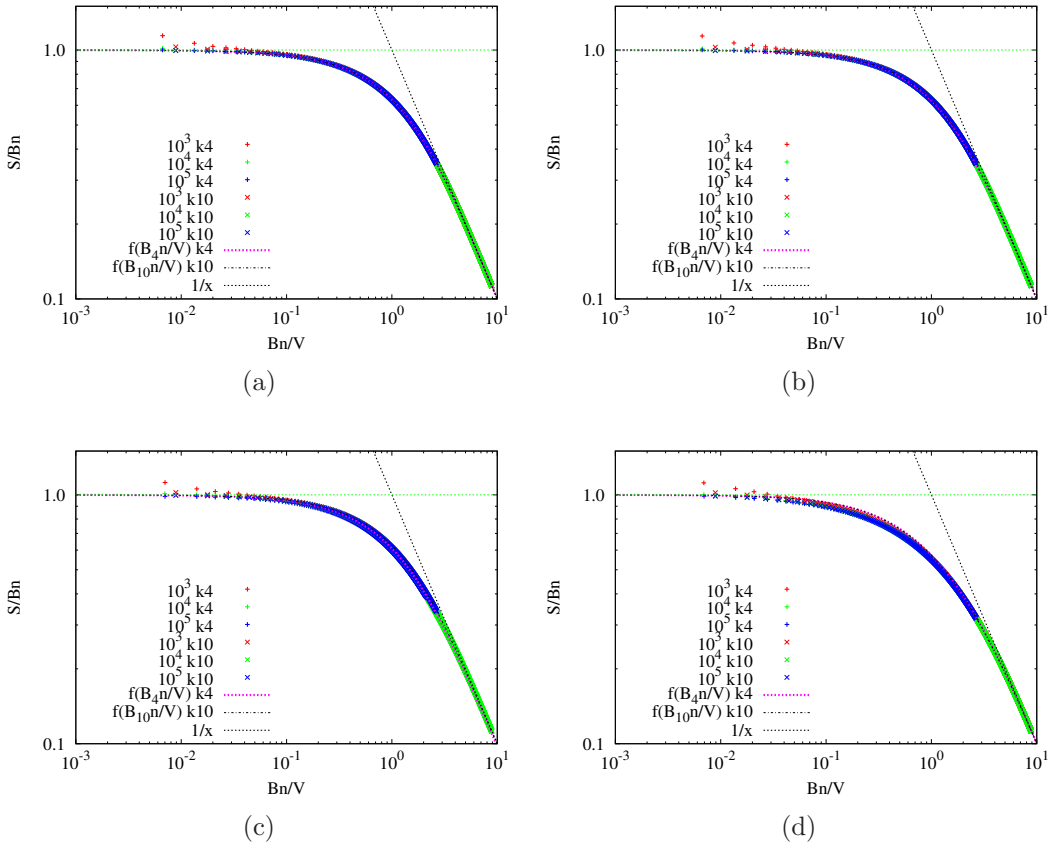


Figure 4: Finite-size scaling of number of distinct sites visited, showing $y = S(n)/(Bn)$ versus $x = Bn/V$. Data from direct simulations (symbols), with B predicted from the cavity equations, are shown for graphs of sizes $V = 10^3, 10^4, 10^5$ and average degrees $\langle k \rangle = 4, 10$. The graph topologies are: (a) Regular; (b) RER; (c) ER; (d) SF. Very good collapse onto a master curve $y = f(x)$ is seen between the different average degrees and – in (a,b,c) – also different V. The initial plateau at $y = 1$ shows the agreement between direct simulations and cavity predictions. For larger x saturation sets in, with $f(x) \approx 1/x$ asyptotically (dotted black line).

often one-parameter) fit for the scaling function.

The accurate results we obtained using message-passing techniques may open new perspectives in the analysis of random walks on networks. The cavity method we applied to study random walks on networks could be considered as a valid alternative tool to analyse other types of quantities related to this problem. For instance one could develop further the model by considering a set of N independent random walkers over a random network and studying the behavior of the average number of distinct or common visited sites, as has been done in the case of lattices [7, 8, 41]. This could give insights into the occupancy statistics of packet-switched networks where packets of data move by independently hopping along nodes to transmit informations between users. The general character of our analysis suggests to us that it should be feasible to adapt it to the study

of this or similar types of questions that arise in the study of random walks on networks.

Acknowledgement

This work is supported by the Marie Curie Training Network NETADIS (FP7, grant 290038), the ANR Grant No. 2011-BS04-013-01 WALKMAT and in part by the Indo-French Centre for the Promotion of Advanced Research under Project No. 4604-3. We acknowledge very helpful conversations with Reimer Kühn.

Appendices

A. The graph representation of the Gaussian covariate distribution.

We can rewrite the joint distribution (35) using $\hat{R}^{-1}_{ij}(z) = \delta_{ij} - z\frac{a_{ij}}{\sqrt{k_i k_j}}$. In this way we can separate the node and edge contributions respectively to obtain a graphical model representation:

$$\begin{aligned}
P(\bar{x}) &\sim e^{-\bar{x}^T \hat{R}^{-1}(z)\bar{x}/2} \\
&= \exp\left(-\sum_i x_i [\hat{R}^{-1}(z)\bar{x}]_i/2\right) \\
&= \exp\left(-\sum_i x_i [\sum_j \hat{R}^{-1}_{ij}(z) x_j]/2\right) \\
&= \exp\left(-\sum_i x_i [\sum_j (\delta_{ij} - z\frac{a_{ij}}{\sqrt{k_i k_j}}) x_j]/2\right) \\
&= \exp\left(-\sum_i x_i [x_i - z\sum_{j\in\partial i} \frac{x_j}{\sqrt{k_i k_j}}]/2\right) \\
&= \exp\left(-\sum_i \left\{\frac{1}{2}x_i^2 - \frac{1}{2}z x_i \sum_{j\in\partial i} \frac{x_j}{\sqrt{k_i k_j}}\right\}\right) \\
&= \prod_{i\in\mathcal{V}} e^{-\frac{1}{2}x_i^2} \prod_{(ij)\in\mathcal{E}} e^{z\frac{x_i x_j}{\sqrt{k_i k_j}}}
\end{aligned} \tag{56}$$

B. Regular graph case.

We calculate an exact expression for the topology dependent prefactor in the case of a regular graph. Using $k_i = k$, $v_i^{(j)} = v^{(j)}$, $m_{i\to j} = m$, $v_j = v$, (48) and (51) we get:

$$\begin{aligned}
v &= k\,[k - \sum_{k\in\partial i} m]^{-1} \\
&= k\,[k - km]^{-1}
\end{aligned}$$

$$= k \left[k \left(1 - \frac{1}{k-1} \right) \right]^{-1}$$
$$= k \frac{k-1}{k(k-2)} \tag{57}$$

We substitute into the expressions (49) for the prefactor B to obtain:

$$B = \frac{1}{Vk} \sum_{j \in V} \frac{k}{v}$$
$$= \frac{1}{Vk} \frac{Vk(k-2)}{k-1}$$
$$= \frac{k-2}{k-1} \tag{58}$$

Therefore the large time limit of the average number of distinct sites of a random walk on a k-regular graph is:

$$\lim_{n \to \infty} S(n) = \left(\frac{k-2}{k-1} \right) n \tag{59}$$

References.

[1] Jasch F and Blumen A. Target problem on small-world networks. *Physical Review E*, 63(4):041108, 2001.

[2] Jasch F and Blumen A. Trapping of random walks on small-world networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 64(6 Pt 2):066104–066104, 2001.

[3] Beeler Jr JR. Distribution functions for the number of distinct sites visited in a random walk on cubic lattices: Relation to defect annealing. *Physical Review*, 134(5A):A1396, 1964.

[4] Klafter J and Blumen A. Models for dynamically controlled relaxation. *Chemical physics letters*, 119(5):377–382, 1985.

[5] Cattuto C, Barrat A, Baldassarri A, Schehr G, and Loreto V. Collective dynamics of social annotation. *Proceedings of the National Academy of Sciences*, 106(26):10511–10515, 2009.

[6] Yeung C H and Saad D. Networking—a statistical physics perspective. *Journal of Physics A: Mathematical and Theoretical*, 46(10):103001, 2013.

[7] Larralde H, Trunfio P, Havlin S, Stanley H E, and Weiss G H. Number of distinct sites visited by n random walkers. *Physical Review A*, 45(10):7128, 1992.

[8] Kundu A, Majumdar S N, and Schehr G. Exact distributions of the number of distinct and common sites visited by n independent random walkers. *Physical review letters*, 110(22):220602, 2013.

[9] Vineyard G H. The number of distinct sites visited in a random walk on a lattice. *Journal of Mathematical Physics*, 4(9):1191–1193, 1963.

[10] Montroll E W and Weiss G H. Random walks on lattices. ii. *Journal of Mathematical Physics*, 6(2):167–181, 1965.

[11] Dvoretzky A and Erdös P. Proceedings of the second berkeley symposium. 1951.

[12] Hughes Barry D and Sahimi M. Random walks on the bethe lattice. *Journal of Statistical Physics*, 29(4):781–794, 1982.

[13] Burioni R and Cassi D. Random walks on graphs: ideas, techniques and results. *Journal of Physics A: Mathematical and General*, 38(8):R45, 2005.

[14] Redner S. *A guide to first-passage processes*. Cambridge University Press, 2001.

[15] Watts D J and Strogatz S H. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[16] Barabási A-L and Albert R. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[17] Mézard M, Parisi G, and Virasoro M A. *Spin glass theory and beyond*, volume 9. World scientific Singapore, 1987.

[18] Rogers T, Castillo I P, Kühn R, and Takeda K. Cavity approach to the spectral density of sparse symmetric random matrices. *Physical Review E*, 78(3):031116, 2008.

[19] Mézard M and Montanari A. *Information, physics, and computation*. Oxford University Press, 2009.

[20] Noh J D and Rieger H. Random walks on complex networks. *Physical review letters*, 92(11):118701, 2004.

[21] Fisher M E. Walks, walls, wetting, and melting. *Journal of Statistical Physics*, 34(5-6):667–729, 1984.

[22] Chung F RK. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[23] Dirac P A M. A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 35:416–418, 7 1939.

[24] Frobenius G. Über matrizen aus nicht negativen elementen. *Sitzungsber. Königl. Preuss. Akad. Wiss. Berlin*, page 456–477, 1912.

[25] Perron O. Zur theorie der matrices. *Mathematische Annalen*, 64(2):248–263, 1907.

[26] Friedman J. On the second eigenvalue and random walks in random d-regular graphs. *Combinatorica*, 11(4):331–362, 1991.

[27] Broder A and Shamir E. On the second eigenvalue of random regular graphs. In *Foundations of Computer Science, 1987., 28th Annual Symposium on*, pages 286–294. IEEE, 1987.

[28] Farkas I J, Derényi I, A-L Barabási, and Vicsek T. Spectra of "real-world" graphs: Beyond the semicircle law. *Physical Review E*, 64(2):026704, 2001.

[29] Füredi Z and Komlós J. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.

[30] Chung F, Lu L, and Vu V. Spectra of random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 100(11):6313–6318, 2003.

[31] Edwards S F and Jones R C. The eigenvalue spectrum of a large symmetric random matrix. *Journal of Physics A: Mathematical and General*, 9(10):1595, 1976.

[32] Wormald N C. Models of random regular graphs. *London Mathematical Society Lecture Note Series*, pages 239–298, 1999.

[33] Erdős P and Rényi A. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.

[34] Monasson R. Diffusion, localization and dispersion relations on "small-world" lattices. *The European Physical Journal B-Condensed Matter and Complex Systems*, 12(4):555–567, 1999.

[35] Urry M J and Sollich P. Random walk kernels and learning curves for Gaussian process regression on random graphs. *The Journal of Machine Learning Research*, 14(1):1801–1835, 2013.

[36] Urry M J and Sollich P. Replica theory for learning curves for Gaussian processes on random graphs. *Journal of Physics A: Mathematical and Theoretical*, 45(42):425005, 2012.

[37] Sollich P, Tantari D, Annibale A, and Barra A. Extensive load in multitasking associative networks. *arXiv:1404.3654*, 2014.

[38] Kühn R. Spectra of sparse random matrices. *Journal of Physics A: Mathematical and Theoretical*, 41(29):295002, 2008.

[39] Alon N. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.

[40] Nilli A. On the second eigenvalue of a graph. *Discrete Mathematics*, 91(2):207–210, 1991.

[41] Majumdar S N and Tamm M V. Number of common sites visited by n random walkers. *Physical Review E*, 86(2):021135, 2012.

A matrix product algorithm for stochastic dynamics on locally tree-like graphs

Thomas Barthel,[1,2] Caterina De Bacco,[2] and Silvio Franz[2]

[1] *Department of Physics, Duke University, Durham, NC 27708, USA*
[2] *Laboratoire de Physique Théorique et Modèles Statistiques, Université Paris-Sud, CNRS UMR 8626, 91405 Orsay Cedex, France*

(Dated: July 27, 2015)

We describe and demonstrate an algorithm for the efficient simulation of generic stochastic dynamics of classical degrees of freedom defined on the vertices of a locally tree-like graph. Networks with cycles are treated in the framework of the cavity method. Such models correspond for example to spin-glass systems, Boolean networks, neural networks, or other technological, biological, and social networks. Building upon ideas from quantum many-body theory, the algorithm is based on a matrix product approximation of the so-called edge messages – conditional probabilities of vertex variable trajectories. The matrix product edge messages (MPEM) are constructed recursively. Computation costs and precision can be tuned by controlling the matrix dimensions of the MPEM in truncations. In contrast to Monte Carlo simulations, the approach has a better error scaling and works for both, single instances as well as the thermodynamic limit. As we demonstrate at the example of Glauber dynamics, due to the absence of cancellation effects, observables with small expectation values can be evaluated reliably, allowing for the study of decay processes and temporal correlations.

PACS numbers: 64.60.aq, 02.50.-r, 02.70.-c

Introduction. – In the last years, we have seen increased efforts of statistical physicists to tackle stochastic dynamical processes in networks in order to study various phenomena [1, 2] such as ordering processes, the spreading of epidemics and opinions, synchronization, collective behavior in social networks, stability under perturbations, or avalanche dynamics.

A drastic simplification can be achieved when short cycles in the network, defined by interaction terms, are very rare. This is the case for locally tree-like graphs such as random regular graphs, Erdős-Rény graphs, and Gilbert graphs. For such random graphs with N vertices, almost all cycles have length $\gtrsim \log N$ such that their effect is negligible in the thermodynamic limit [3]. For static problems, this has been exploited in the so-called cavity method [4], where conditional nearest-neighbor probabilities are computed iteratively within the Bethe-Peierls approximation. The method was very successfully applied to study for example equilibrium properties of spin glasses [4], computationally hard satisfiability problems [5, 6], and random matrix ensembles [7].

This big success has motivated the generalization of the cavity method to dynamical problems [8, 9], which is known as the dynamic cavity method or dynamic belief propagation. As the number of possible trajectories and, hence, the computational complexity increase exponentially in time, applications have however been quite restricted to either very short times [8, 10], oriented graphs [8], or unidirectional dynamics with local absorbing states [9, 11–13]. In the latter case, one can exploit that vertex trajectories can be parametrized by a few switching times. Another idea has been to neglect temporal correlations completely as in the one-step method [8, 14–16] or to retain only some $\Delta t = 1$ correlations as in the 1-step Markov ansatz [17]. While this works well sometimes for stationary states at high temperatures, such approximations are usually quite severe for short to intermediate times or low temperatures.

In this paper, we present an efficient novel algorithm for the solution of the parallel dynamic cavity equations for generic (locally tree-like) graphs and generic bidirectional dynamics. The central objects in the dynamic cavity method are conditional probabilities for vertex trajectories of nearest neighbors – the so-called edge messages. As temporal correlations are decaying in time and/or time difference $|t - t'|$, we can approximate each edge message by a matrix product, i.e., there is one matrix for every edge, edge state, and time step, encoding the temporal correlations in the corresponding part of the evolution. It turns out that the dimensions of these matrices do not have to be increased exponentially in time. One can obtain quasi-exact results with much smaller matrix dimensions. Computation costs and precision can be tuned by controlling the dimensions in truncations. The idea of exploiting the decay of temporal correlations to approximate edge messages in matrix product form is in analogy with the use of matrix product states [18–22] for the simulation of strongly correlated, mostly one-dimensional, quantum many-body systems. These have been used very successfully in algorithms like the density-matrix renormalization group [23, 24] to compute for example quantum ground-state properties, often with machine precision [25]. Besides lifting the restrictions of the aforementioned approaches, the matrix product edge-message (MPEM) algorithm can also outperform Monte Carlo simulations (MC) of the dynamics in important respects. In particular, besides allowing for the simulation of single instances, alternatively, one can work directly in

the thermodynamic limit. Perhaps more importantly, it has a favorable error scaling. While statistical errors in MC decay very slowly with the number of samples N_s as $1/\sqrt{N_s}$, MPEM yields also observables with absolutely small expectation values with very good precision which is essential for the study of decay processes and temporal correlations.

The dynamic cavity method. – Let σ_i^t denote the state of vertex i at time step t, and $\boldsymbol{\sigma}^t := (\sigma_1^t, \sigma_2^t, \dots)$ the state of the full system at time t. Given the state probabilities $P(\boldsymbol{\sigma}^t)$ for time t, we evolve to the next time step, $P(\boldsymbol{\sigma}^{t+1}) = \sum_{\boldsymbol{\sigma}^t} W(\boldsymbol{\sigma}^{t+1}|\boldsymbol{\sigma}^t) P(\boldsymbol{\sigma}^t)$, by applying the stochastic transition matrix W. As vertex i interacts only with its nearest neighbors $j \in \partial i$, the probability for σ_i^{t+1} only depends on the states σ_j^t of these vertices at the previous time step such that the global transition matrix W is a product of local transition matrices w_i,

$$W(\boldsymbol{\sigma}^{t+1}|\boldsymbol{\sigma}^t) = \prod_i w_i(\sigma_i^{t+1}|\boldsymbol{\sigma}_{\partial i}^t). \tag{1}$$

Here $\sum_{\sigma_i} w_i(\sigma_i|\boldsymbol{\sigma}'_{\partial i}) = 1$, and $\boldsymbol{\sigma}_{\partial i}^t$ is the state of the nearest neighbors of vertex i at time t. In the cavity method [4, 8, 9], one neglects cycles of the (locally tree-like) graph according to the Bethe-Peierls approximation to reduce this computationally complex evolution to the dynamic cavity equation [8, 9]

$$\mu_{i\to j}(\bar\sigma_i^{t+1}|\bar\sigma_j^t) = \sum_{\{\bar\sigma_k^t\}_{k\in\partial i\setminus\{j\}}} p_i(\sigma_i^0)\Big[\prod_{s=0}^{t} w_i(\sigma_i^{s+1}|\boldsymbol{\sigma}_{\partial i}^s)\Big] \times \Big[\prod_{k\in\partial i\setminus\{j\}} \mu_{k\to i}(\bar\sigma_k^t|\bar\sigma_i^{t-1})\Big] \tag{2}$$

which only involves the so-called edge messages μ for the edges of a single vertex i. For simplicity, we have assumed that vertices are uncorrelated in the initial state such that $P(\boldsymbol{\sigma}^0) = \prod_i p_i(\sigma_i^0)$. The edge messages $\mu_{i\to j}(\bar\sigma_i^t|\bar\sigma_j^{t-1})$ in the dynamic cavity equation (2) are conditional probabilities for the trajectories $\bar\sigma_i^t := (\sigma_i^0, \sigma_i^1, \dots, \sigma_i^t)$ and $\bar\sigma_j^{t-1}$ on edge (i,j). Specifically, if we consider a tree graph and cut off everything "right" of vertex j as indicated in Figure 1 by the dashed line, $\mu_{i\to j}(\bar\sigma_i^t|\bar\sigma_j^{t-1})$ denotes the conditional probability of a trajectory $\bar\sigma_i^t$ on vertex i, given the trajectory $\bar\sigma_j^{t-1}$ on vertex j. Eq. (2) constructs $\mu_{i\to j}(\bar\sigma_i^{t+1}|\bar\sigma_j^t)$ out of the edge messages $\mu_{k\to i}(\bar\sigma_k^t|\bar\sigma_i^{t-1})$ of the previous time step. This is exact for tree graphs and covers locally tree-like graphs in the Bethe-Peierls approximation. Although we have gained a lot in the sense that the computational complexity is now linear in the system size, it is still exponential in time t, if we were to encode the edge messages without any approximation.

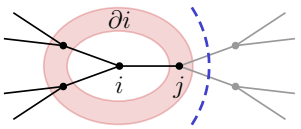


Figure 1: (Color online) Part of a locally tree-like interaction graph with vertex degrees $z = 3$.

Canonical from of an MPEM. – To circumvent this exponential increase of computation costs, we can exploit the decay of temporal correlations and approximate the exact edge message by a matrix product

$$\mu_{i\to j}(\bar\sigma_i^t|\bar\sigma_j^{t-1}) = A_{i\to j}^{(0)}(\sigma_j^0)\Big[\prod_{s=1}^{t-1} A_{i\to j}^{(s)}(\sigma_i^{s-1}|\sigma_j^s)\Big] \times A_{i\to j}^{(t)}(\sigma_i^{t-1}) A_{i\to j}^{(t+1)}(\sigma_i^t). \tag{3}$$

The particular choice of assigning vertex variables $\{\sigma_i^s\}$ and $\{\sigma_j^s\}$ to the $M_s \times M_{s+1}$ matrices $A_{i\to j}^{(s)}(\sigma_i^{s-1}|\sigma_j^s)$ occurring in the matrix product (3), is advantageous for the implementation of the recursion relation (2) for MPEMs as will become clear in the following. In order for the matrix product to yield a scalar, we set $M_0 = M_{t+2} = 1$.

MPEM evolution. – The time-evolution starts at $t = 0$ with $\mu_{i\to j}(\sigma_i^0) = p_i(\sigma_i^0)$. Using the dynamic cavity equation (2), we iteratively build matrix product approximations for edge messages for time $t+1$ from those for time t. It is simple to insert the matrix product ansatz (3) for the edge messages in the dynamic cavity equation, but not trivial to bring the resulting edge message again into the canonical MPEM form as required for the subsequent evolution step. The specific assignment of the vertex variables to matrices in Eq. (3) has been chosen such that all contractions (products and sums over vertex variables) occurring in the cavity equation are time-local in the sense that, given MPEMs $\mu_{k\to i}(\bar\sigma_k^t|\bar\sigma_i^{t-1})$ in canonical form for all neighbors $k \in \partial i\setminus\{j\}$, the resulting $\mu_{i\to j}(\bar\sigma_k^{t+1}|\bar\sigma_i^t)$ can be written in (non-canonical) matrix product form as

$$\mu_{i\to j}(\bar\sigma_i^{t+1}|\bar\sigma_j^t) = C_{i\to j}^{(0)}(\sigma_i^0)\Big[\prod_{s=1}^{t+1} C_{i\to j}^{(s)}(\sigma_i^s|\sigma_j^{s-1})\Big]. \tag{4}$$

As depicted in Figure 2b, the tensors $C_{i\to j}^{(s)}$ for $1 \le s \le t$ are obtained by contracting a local transition matrix $w_i(\sigma_i^s|\boldsymbol{\sigma}_{\partial i}^{s-1})$ with tensors $A_{k\to i}^{(s)}$ from the time-t MPEMs. This contraction entails a sum over the $z-1$ common indices σ_k^{s-1}, where $z = |\partial i|$ is the vertex degree. Assuming for the simplicity of notation that the matrix dimensions M_s for all time-t MPEMs are identical, the resulting matrices $C_{i\to j}^{(s)}(\sigma_i^s|\sigma_j^{s-1}) = \sum_{\boldsymbol{\sigma}_{\partial i\setminus\{j\}}^{s-1}} w_i(\sigma_i^s|\boldsymbol{\sigma}_{\partial i}^{s-1}) \times \Big[\bigotimes_{k\in\partial i\setminus\{j\}} A_{k\to i}^{(s)}(\sigma_k^{s-1}|\sigma_i^s)\Big]$ have left and right indices of dimensions $\bar M_s = (M_s)^{z-1}$ and $\bar M_{s+1} = (M_{s+1})^{z-1}$, respectively. The contraction for tensor $C^{(t+1)}$ is very similar and $C_{i\to j}^{(0)}(\sigma_i^0) = p_i(\sigma_i^0)\Big[\bigotimes_{k\in\partial i\setminus\{j\}} A_{k\to i}^{(0)}(\sigma_i^0)\Big]$.

MPEM truncation. – In preparation for the next time step, we now need to bring the evolved edge message (4)

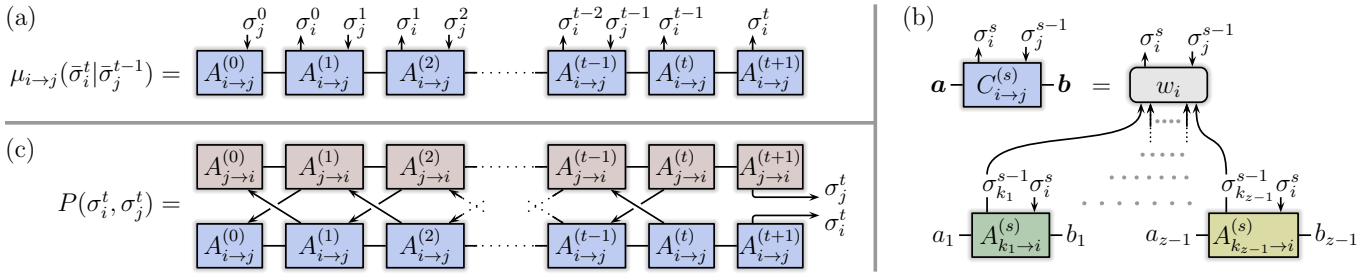


Figure 2: (Color online) (a) Graphical representation of a matrix product edge message in canonical form (3). Connecting lines indicate summations over indices. (b) For each time step (2), tensors of the evolved matrix product $\mu_{i\to j}(\bar\sigma_i^{t+1}|\bar\sigma_j^t)$ in Eq. (4) are built by contracting the local transition matrix w_i with MPEM tensors of messages $\mu_{k\to i}$, incident to vertex i, where $k\in\partial i\setminus\{j\}=\{k_1,\dots,k_{z-1}\}$, and $\boldsymbol{a}:=(a_1,\dots,a_{z-1})$. (c) Evaluation of probabilities $P(\sigma_i^t,\sigma_j^t)$ as in Eq. (8).

back into canonical form (3). Furthermore, we need to introduce a controlled approximation that reduces the matrix dimensions because they would otherwise grow exponentially in time. Both, a reordering of the vertex variables σ_i^s and σ_j^s in the matrix product (4) and a controlled truncation of matrix dimensions can be achieved by sweeping through the matrix product and doing certain singular value decompositions (SVD) [26] of the tensors.

Let us shortly explain the notion of truncations at the example of a matrix product $\gamma(\boldsymbol{n}):=A_0^{n_0}A_1^{n_1}\cdots A_t^{n_t}$, where $A_s^{n_s}$ is an $M_s\times M_{s+1}$ matrix and $M_0=M_{t+1}=1$. In order to reduce in a controlled way, e.g., the left matrix dimension M_u of $A_u^{n_u}$, we suggest to do an SVD of the matrix product such that

$$\gamma(\boldsymbol{n})=:\Gamma_{\boldsymbol{n}_L,\boldsymbol{n}_R}=\sum_{k=1}^{M_u}Y_{\boldsymbol{n}_L,k}\lambda_k Z_{k,\boldsymbol{n}_R} \tag{5}$$

where we have grouped the variables to $\boldsymbol{n}_L:=(n_0,\dots,n_{u-1})$ and $\boldsymbol{n}_R:=(n_u,\dots,n_t)$. Y and Z are isometric matrices; $Y^\dagger Y=\mathbb{1}$ and $ZZ^\dagger=\mathbb{1}$. Now, truncating some of the singular values $\lambda_1\ge\lambda_2\ge\cdots\ge\lambda_{M_u}\ge 0$, such that only the M_u' largest are retained, we obtain the controlled approximation

$$\gamma_{\rm trunc}(\boldsymbol{n}):=\sum_{k\le M_u'}Y_{\boldsymbol{n}_L,k}\lambda_k Z_{k,\boldsymbol{n}_R}$$
$$\text{with error}\quad \|\gamma-\gamma_{\rm trunc}\|^2=\sum_{k>M_u'}\lambda_k^2. \tag{6}$$

Note that this truncation scheme guarantees the minimum possible two-norm loss $\|\Delta\gamma\|\equiv\left(\sum_{\boldsymbol{n}}\Delta\gamma^2(\boldsymbol{n})\right)^{1/2}$ for the given new matrix dimension M_u'.

While it seems very desirable to discard unimportant information and control the growth of computation cost through such truncations, the SVD (5) appears to be an insurmountable task. Assuming that each variable n_s can take d different values and that $2u\le t+1$, the cost for the SVD would scale exponentially in time like d^{t+u+1} [26]. However, the beauty of matrix products is that such an SVD can in fact be done sequentially with linear costs of order tdM^3 ($M:=\max_s M_s$) as follows. First, we do an exact transformation of the matrix product to bring it to the form $\gamma(\boldsymbol{n})=Y_0^{n_0}\cdots Y_{u-1}^{n_{u-1}}\tilde A_u^{n_u}Z_{u+1}^{n_{u+1}}\cdots Z_t^{n_t}$, where tensors Y_s and Z_s obey the left and right orthonormality constraints

$$\sum_n (Y_s^n)^\dagger Y_s^n=\mathbb{1}\quad\text{and}\quad\sum_n Z_s^n(Z_s^n)^\dagger=\mathbb{1}, \tag{7}$$

respectively. This is achieved through a sequence of SVDs. It starts with the SVD $A_0^{n_0}=:Y_0^{n_0}\Lambda_0V_0$, where Λ_0 is a diagonal matrix of singular values, V_0 is isometric according to $V_0V_0^\dagger=\mathbb{1}$ and Y_0 obeys Eq. (7). The sweep continues with the SVD $\Lambda_0V_0A_1^{n_1}=:Y_1^{n_1}\Lambda_1V_1$ and so on until the computation of Y_{u-1}. Analogously, we do a second sequence of SVDs starting from the right and ending with $A_{u+1}^{n_{u+1}}U_{u+2}\Lambda_{u+2}=:U_{u+1}\Lambda_{u+1}Z_{u+1}^{n_{u+1}}$ and, finally, define the central tensor as $\tilde A_u^{n_u}:=\Lambda_{u-1}V_{u-1}A_u^{n_u}U_{u+1}\Lambda_{u+1}$. After this somewhat laborious preparation, we can do the actual truncation, based on the SVD $\tilde A_u^{n_u}=:U_u\Lambda Z_u^{n_u}$ with the same singular values $\lambda_1\ge\cdots\ge\lambda_{M_u}$ as in Eq. (5). With the $M_u\times M_u'$ matrix $[\Lambda_{\rm trunc}]_{kk'}:=\delta_{kk'}\lambda_k$, the truncated matrix product (6) takes the form $\gamma_{\rm trunc}(\boldsymbol{n})=Y_0^{n_0}\cdots Y_{u-2}^{n_{u-2}}(Y_{u-1}^{n_{u-1}}U_u\Lambda_{\rm trunc})Z_u^{n_u}\cdots Z_t^{n_t}$.

With this tool in hand, we can now truncate the evolved edge message (4) and bring it back into canonical form (3). In a first sweep from right ($s=t+1$) to left ($s=0$), using SVDs, we can sequentially impose the right orthonormality constraints [see Eq. (7)] on the C-tensors. In a subsequent sweep from left to right, again based on SVDs, we can now truncate the tensors to decrease bond dimensions from $\bar M_s$ to something smaller. What is left, is to reorder the indices $\{\sigma_i^s\}$ and $\{\sigma_j^s\}$ of the vertex variables. In a sweep from right to left, we go from the variable assignment $(\sigma_i^0)(\sigma_i^1|\sigma_j^0)\dots(\sigma_i^{t+1}|\sigma_j^t)$ in the truncated and orthonormalized version $\tilde C_{i\to j}^{(0)}(\sigma_i^0)\prod_{s=1}^{t+1}\tilde C_{i\to j}^{(s)}(\sigma_i^s|\sigma_j^{s-1})$ of the MPEM (4) to the assignment $(\sigma_i^0\sigma_j^0)\dots(\sigma_i^t|\sigma_j^t)(\sigma_i^{t+1})$

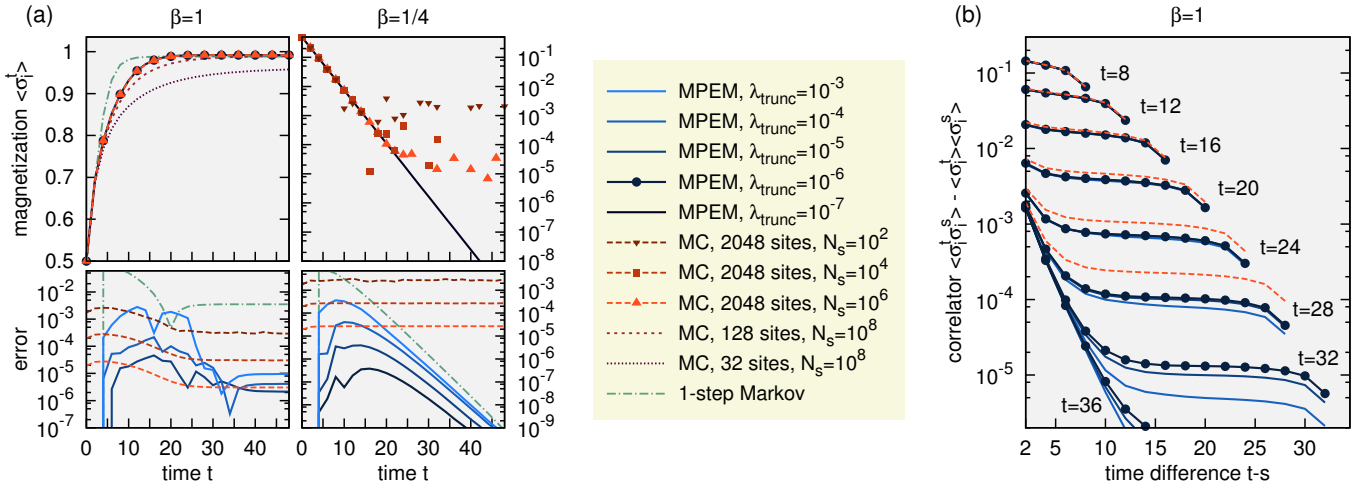


Figure 3: (Color online) (a) Magnetization and, (b), connected temporal correlations for Glauber dynamics on $z = 3$ random regular graphs of different sizes for MC and in the thermodynamic limit for the MPEM and 1-step Markov approaches. Because of odd-even effects in the dynamics, only even time steps are shown. For MC (with N_s samples), the errors of the magnetization [lower panels in (a)] are quantified by the standard deviation of the magnetization, i.e., under ignorance of remaining finite-size effects. For MPEM and 1-step Markov, errors are quantified by the deviation from the result of the most precise (quasi-exact) MPEM simulation (truncation threshold $\lambda_{\rm trunc} = 10^{-6}$ for $\beta = 1$ and $\lambda_{\rm trunc} = 10^{-7}$ for $\beta = 1/4$). In plot (b), the three MPEM curves for $\lambda_{\rm trunc} = 10^{-4}, 10^{-5}, 10^{-6}$ overlap up to time $t = 24$.

in the matrix product

$$\mu_{i\to j}(\bar\sigma_i^{t+1}|\bar\sigma_j^t) \overset{\rm trunc}{\approx} \Big[\prod_{s=0}^{t} D_{i\to j}^{(s)}(\sigma_i^s|\sigma_j^s)\Big] D_{i\to j}^{(t+1)}(\sigma_i^{t+1}).$$

At the right boundary, we start with an SVD and controlled truncation $\tilde C_{i\to j}^{(t+1)}(\sigma_i^{t+1}|\sigma_j^t) \approx: U^{(t+1)}(\sigma_j^t) \times\Lambda_{\rm trunc}^{(t+1)} D_{i\to j}^{(t+1)}(\sigma_i^{t+1})$, and continue with $\tilde C_{i\to j}^{(t)}(\sigma_i^t|\sigma_j^{t-1}) \times U^{(t+1)}(\sigma_j^t)\Lambda_{\rm trunc}^{(t+1)} \approx: U^{(t)}(\sigma_j^{t-1})\Lambda_{\rm trunc}^{(t)} D_{i\to j}^{(t)}(\sigma_i^t|\sigma_j^t)$ and so on until ending at $s = 0$. In an analogous final sweep from left to right, we change to the canonical variable assignment $(\sigma_j^0)(\sigma_i^0|\sigma_j^1)\dots(\sigma_i^{t-1}|\sigma_j^t)(\sigma_i^t|\sigma_j^{t+1})$ as in Eq. (3). After executing these steps for all edge messages, the next evolution step from $t+1$ to $t+2$ can follow.

Evaluation of observables. – The joint probability of trajectories $\bar\sigma_i^t$ and $\bar\sigma_j^{t-1}$ for the vertices of an edge (i,j) is given by the product of the two corresponding edge messages. After marginalization, one obtains for example the probability for the edge state (σ_i^t,σ_j^t) at time t as

$$P(\sigma_i^t,\sigma_j^t) = \sum_{\bar\sigma_i^{t-1},\bar\sigma_j^{t-1}} \mu_{i\to j}(\bar\sigma_i^t|\bar\sigma_j^{t-1})\mu_{j\to i}(\bar\sigma_j^t|\bar\sigma_i^{t-1}). \quad (8)$$

In the MPEM approach, this can be evaluated efficiently, as indicated in Figure 2c, by executing the contractions sequentially from left ($s = 0$) to right ($s = t-1$). Similarly, one can for example also compute temporal correlators $\langle\sigma_i^t\sigma_i^s\rangle$ from probabilities $P(\sigma_i^t,\sigma_i^s)$.

Exemplary application. – Figure 3 compares the simulation of Glauber dynamics [27] using our MPEM algorithm to MC simulations and to the 1-step Markov approximation [17]. Specifically, we have Ising spins interacting ferromagnetically on $z = 3$ random regular graphs, with local transition matrices $w_i(\sigma_i^{t+1}|\boldsymbol{\sigma}_{\partial i}^t) = \exp(\beta\sum_{j\in\partial i}\sigma_i^{t+1}\sigma_j^t)/Z$. In the initial state, all spins have magnetization $\langle\sigma_i^0\rangle = 1/2$, i.e., $p_i(\uparrow) = 3/4$. Besides being applicable for single instances of finite graphs, the MPEM approach gives also direct access to the thermodynamic limit. For disordered systems this can be done in a population dynamics scheme. The homogeneous case, considered here, is particularly simple as all edges of the graph are equivalent in the thermodynamic limit. Hence, one can work with a single MPEM. Figure 3a shows the evolution of the magnetization. In the ferromagnetic phase ($\beta = 1$), it approaches a finite equilibrium value, whereas it decays to zero in the paramagnetic phase ($\beta = 1/4$). As shown for $\beta = 1$, MC simulations contain finite size effects which become small for the system with 2048 sites. MC errors decrease slowly when increasing the number of samples N_s as $1/\sqrt{N_s}$. This is problematic for observables with small absolute values where cancellation effects make it difficult to get a precise estimate. This is, e.g., apparent in the magnetization decay for $\beta = 1/4$ which, in contrast, is very precisely captured with MPEM. In these simulations, we control the MPEM precision by keeping only singular values λ_k above a threshold, specified by $\lambda_k/\sum_{k'}\lambda_{k'} > \lambda_{\rm trunc}$. Decreasing $\lambda_{\rm trunc}$, increases precision and computation costs. The 1-step Markov approximation [17] is not suited to handle temporal correlations. At long times it performs well for $\beta = 1/4$ and fairly good for $\beta = 1$, but deviates rather strongly at earlier times. Figure 3a shows the connected temporal correla-

tion function $\langle \sigma_i^t \sigma_i^s \rangle - \langle \sigma_i^t \rangle \langle \sigma_i^s \rangle$ for $\beta = 1$ as a function of $t - s$ for several times. Its decay behavior can be difficult to impossible to capture with MC. In the example, MC deviations are often orders of magnitude above those of the numerically cheaper MPEM simulations.

Conclusion. – The novel MPEM algorithm, based on matrix product approximations of edge messages allows for an efficient and precise solution of the dynamic cavity equations. Besides lifting restrictions of earlier approaches, mentioned in the introduction, it gives direct access to the thermodynamic limit, and its error scaling is favorable to that of MC simulations. We think that it is a very valuable tool, particularly, as it yields temporal correlations and other decaying observables with unprecedented precision and gives access to low-probability events. This opens a new door for the study of diverse dynamic processes and inference or dynamic optimization problems for physical, technological, biological, and social networks.

We thank G. Del Ferraro for providing data to benchmark our 1-step Markov simulation for Figure 3 and acknowledge supported by the Marie Curie Training Network NETADIS (FP7, grant 290038).

[1] A. V. Alain Barrat, Marc Barthélemy, *Dynamical Processes on Complex Networks* (Cambridge University Press, New York, 2008).
[2] M. E. J. Newman, *Networks: An Introduction* (Oxford University Press, Oxford, 2010).
[3] M. Mézard and A. Montanari, *Information, Physics, and Computation* (Oxford University Press, New York, 2009).
[4] M. Mézard and G. Parisi, Eur. Phys. J. B **20**, 217 (2001).
[5] M. Mézard, G. Parisi, and R. Zecchina, Science **297**, 812 (2002).
[6] M. Mézard and R. Zecchina, Phys. Rev. E **66**, 056126 (2002).
[7] T. Rogers, I. P. Castillo, R. Kühn, and K. Takeda, Phys. Rev. E **78**, 031116 (2008).
[8] I. Neri and D. Bollé, J. Stat. Mech. **2009**, P08009 (2009).
[9] B. Karrer and M. E. J. Newman, Phys. Rev. E **82**, 016101 (2010).
[10] Y. Kanoria and A. Montanari, Ann. Appl. Probab. **21**, 1694 (2011).
[11] A. Y. Lokhov, M. Mézard, and L. Zdeborová, Phys. Rev. E **91**, 012811 (2015).
[12] F. Altarelli, A. Braunstein, L. Dall'Asta, A. Lage-Castellanos, and R. Zecchina, Phys. Rev. Lett. **112**, 118701 (2014).
[13] M. Shrestha and C. Moore, Phys. Rev. E **89**, 022805 (2014).
[14] E. Aurell and H. Mahmoudi, J. Stat. Mech. **2011**, P04014 (2011).
[15] E. Aurell and H. Mahmoudi, Phys. Rev. E **85**, 031119 (2012).
[16] P. Zhang, J. Stat. Phys. **148**, 502 (2012).
[17] G. Del Ferraro and E. Aurell, arXiv:1409.4684 (2014).
[18] L. Accardi, Phys. Rep. **77**, 169 (1981).
[19] M. Fannes, B. Nachtergaele, and R. F. Werner, Comm. Math. Phys. **144**, 443 (1992).
[20] S. Rommer and S. Östlund, Phys. Rev. B **55**, 2164 (1997).
[21] U. Schollwöck, Ann. Phys. **326**, 96 (2011).
[22] I. V. Oseledets, SIAM J. Sci. Comput. **33**, 2295 (2011).
[23] S. R. White, Phys. Rev. Lett. **69**, 2863 (1992).
[24] S. R. White, Phys. Rev. B **48**, 10345 (1993).
[25] U. Schollwöck, Rev. Mod. Phys. **77**, 259 (2005).
[26] G. H. Golub and C. F. Van Loan, *Matrix Computations, 3rd ed.* (Johns Hopkins University Press, Baltimore, Maryland, 1996).
[27] R. J. Glauber, J. Math. Phys. **4**, 294 (1963).

Rare events statistics of random walks on networks: localization and other dynamical phase transitions

Caterina De Bacco,[1] Alberto Guggiola,[2] Reimer Kühn,[3] and Pierre Paga[3]

[1]*LPTMS, Centre National de la Recherche Scientifique et Université Paris-Sud 11, 91405 Orsay Cedex, France.*
[2]*LPTENS, Unité Mixte de Recherche (UMR 8549) du CNRS et de l'ENS, associée à l'UPMC Université Paris 06, 24 Rue Lhomond, 75231 Paris Cedex 05, France.*
[3]*Department of Mathematics, King's College London*

Rare event statistics for random walks on complex networks are investigated using the large deviations formalism. Within this formalism, rare events are realized as typical events in a suitably deformed path-ensemble, and their statistics can be studied in terms of spectral properties of a deformed Markov transition matrix. We observe two different types of phase transition in such systems: (i) rare events which are singled out for sufficiently large values of the deformation parameter may correspond to *localized* modes of the deformed transition matrix; (ii) "mode-switching transitions" may occur as the deformation parameter is varied. Details depend on the nature of the observable for which the rare event statistics is studied, as well as on the underlying graph ensemble. In the present letter we report on the statistics of the average degree of the nodes visited along a random walk trajectory in Erdős-Rényi networks. Large deviations rate functions and localization properties are studied numerically. For observables of the type considered here, we also derive an analytical approximation for the Legendre transform of the large-deviations rate function, which is valid in the large connectivity limit. It is found to agree well with simulations.

Random walks are dynamical processes widely used to analyze, organize or perform important tasks on networks such as searches [1, 2], routing or data transport [3–5]. Their popularity is due to their cheap implementation, as they rely only on local information, such as the state of the neighborhood of a given node of the network. This ensures network scalability and allows fast data transmission without the need for large storage facilities at nodes, such as big routing tables in communication networks. These features make random walks an efficient tool to explore networks characterized by a high cost of information. Examples are sensor networks [6] where many signaling packets are needed to acquire wider networks status information. In peer-to-peer networks the absence of a central server storing file locations requires users to perform repeated local searches in order to find a file to download, and various random walk strategies have been proposed as a scalable method [7–9] in this context. Less attention has been paid to characterize rare events associated with random walks on networks. Yet the occurrence of a rare event can have severe consequences. In hide-and-seek games for instance [10], rare events represent situations where the seeker finds either most (or unusually many) of the hidden targets, or conversely none (or unusually few). In the context of cyber-security, where one is concerned with worms and viruses performing random walks through a network, a rare event would correspond to a situation where unusually many sensible nodes are successfully attacked and infected, which may have catastrophic consequences for the integrity of an entire IT infrastructure. Characterizing the statistics of rare events for random walks in complex networks and its dependence on network topology is thus a problem of considerable technological importance. A variant of this problem was recently analyzed for biased random walks in complex networks [11]. That paper addressed rare fluctuations in single node occupancy for an ensemble of independent (biased) walkers in the stationary state of the system. By contrast, our interest here is in rare event statistics of *path averages*, or equivalently of time integrated variables. Rare event statistics of this type has been looked at for instance in the context of kinetically constrained models of glassy relaxation [12]; relations to constrained ensembles of trajectories were explored in [13] for Glauber dynamics in the 1d Ising chain. While these studies were primarily concerned with the use of large deviations theory as a tool to explore dynamical phase transitions in homogeneous systems, our focus here is on the interplay between rare event statistics and the heterogeneity of the underlying system.

In the present Letter we use large deviations theory to study rare events statistics for path averages of observables associated with sites visited along trajectories of random walks. Within this formalism, rare events are realized as typical events in a suitably deformed path-ensemble [12, 14]. Their statistics can be studied in terms of spectral properties of a deformed version of the Markov transition matrix for the original random walk model, the relevant information being extracted from the algebraically largest eigenvalue of the deformed transition matrix. Such deformation may direct random walks to subsets of a network with vertices of either atypically high or atypically low coordination. It also amplifies the heterogeneity of transition matrix elements for large values of the deformation parameter and we observe that, as a consequence, the eigenvector corresponding to the largest eigenvalue of the deformed transition matrix may exhibit a *localization transition*, indicating that rare large fluctuations of path averages are typically realized by trajectories that remain localized on small subsets of the network. Within localized phases, we also encounter a second type of dynamical phase transition related to *switching between modes* as the deformation parameter used to select rare events is varied. Our methods allow us to study the role that network topology and heterogeneity play in selecting these special paths, as well as

to infer properties of paths actually selected to realize extreme events.

The model. We consider a complex network with adjacency matrix A, with entries $a_{ij} = 1$ if the edge (ij) exists, $a_{ij} = 0$ otherwise. The transition matrix W of an unbiased random walk has entries $W_{ij} = a_{ij}/k_j$ where k_j is the degree of node j and W_{ij} is the probability of a transition from j to i.

Writing $\mathbf{i}_\ell = (i_0, i_1, \cdots, i_\ell)$ a path of length ℓ, quantities of interest are empirical path-averages of the form

$$\hat{\phi}_\ell = \frac{1}{\ell} \sum_{t=1}^{\ell} \xi_{i_t} \,, \tag{1}$$

where the ξ_i are quenched random variables associated with the vertices $i = 1, \ldots, N$ of the graph, which could be independent of, be correlated with, or be deterministic functions of the degrees k_i of the vertices. It is expected that the $\hat{\phi}_\ell$ are for large ℓ sharply peaked about their mean

$$\bar{\phi}_\ell = \frac{1}{\ell} \sum_{\mathbf{i}_\ell} P(\mathbf{i}_\ell) \sum_{t=1}^{\ell} \xi_{i_t} = \left\langle \frac{1}{\ell} \sum_{t=1}^{\ell} \xi_{i_t} \right\rangle \tag{2}$$

where $P(\mathbf{i}_l)$ denotes the probability of the path $\mathbf{i}_l$.

The average (2) can be obtained from the *cumulant generating function* $\psi_\ell(s) = \ell^{-1} \ln \sum_{\mathbf{i}_\ell} P(\mathbf{i}_\ell)\, \mathrm{e}^{s \sum_{t=1}^{\ell} \xi_{i_t}}$ as $\bar{\phi}_\ell = \psi'_\ell(s)|_{s=0}$. Here, we are interested in rare events, for which the empirical averages $\hat{\phi}_\ell$ take values ϕ which differ significantly from their mean $\bar{\phi}_\ell$. Large deviations theory predicts that for $\ell \gg 1$ the probability density $P(\phi)$ for such an event scales exponentially with path-length ℓ, $P(\phi) \sim \mathrm{e}^{-\ell I(\phi)}$, with a *rate function* $I(\phi)$ which, according to the Gärtner-Ellis theorem [14] is obtained as a Legendre transform $I(\phi) = \sup_s \{s\phi - \psi(s)\}$ of the limiting cumulant generating function $\psi(s) = \lim_{\ell \to \infty} \psi_\ell(s)$, provided that this limit exists and that it is differentiable. We shall see that the second condition may be violated, and that the derivative $\psi'(s)$ may develop discontinuities at certain s-values, entailing that we observe regions where $I(\phi)$ is strictly linear and only represents the convex hull of the true rate function [14].

In order to evaluate $\psi_\ell(s)$, we express path probabilities using the Markov transition matrix W and a distribution $\mathbf{p_0} = (p_0(i_0))$ of initial conditions as $P(\mathbf{i}_\ell) = [\prod_{t=1}^{\ell} W_{i_t i_{t-1}}]\, p(i_0)$, entailing that $\psi_\ell(s)$ can be evaluated in terms of a deformed transition matrix $W(s) = (\mathrm{e}^{s\xi_i} W_{ij})$ as $\psi_\ell(s) = \ell^{-1} \ln \sum_{i_\ell, i_0} [W^\ell(s)]_{i_\ell i_0}\, p(i_0)$. Using a spectral decomposition of the deformed transition matrix one can write this as

$$\psi_\ell(s) = \ln \lambda_1 + \frac{1}{\ell} \ln \left[(\mathbf{1}, \mathbf{v}_1)(\mathbf{w}_1, \mathbf{p}_0) + \sum_{\alpha(\neq 1)} \left(\frac{\lambda_\alpha}{\lambda_1} \right)^{\ell} (\mathbf{1}, \mathbf{v}_\alpha)(\mathbf{w}_\alpha, \mathbf{p}_0) \right]. \tag{3}$$

Here the λ_α are eigenvalues of $W(s)$, the $\mathbf{v}_\alpha$ and $\mathbf{w}_\alpha$ are the corresponding right and left eigenvectors, $\mathbf{1} = (1, \ldots, 1)$, and the bracket notation $(\cdot, \cdot)$ is used to denote an inner product. Eigenvalues are taken to be sorted in decreasing order $\lambda_1 \geq |\lambda_2| \geq |\lambda_3| \cdots \geq \lambda_N$, with the first inequality being a consequence of the Perron-Frobenius theorem [15]. This concludes the general framework. For the remainder of this Letter, we will restrict our attention to the case where $\xi_i = f(k_i)$.

For long paths, the value of the cumulant generating function is dominated by the leading eigenvalue $\lambda_1 = \lambda_1(s)$ of the transition matrix $W(s)$, so $\psi(s) = \log \lambda_1(s)$. In the $s = 0$ case, the eigenvalue problem is trivial, as the column-stochasticity of the transition matrix yields a left eigenvector $w_i \equiv 1$ corresponding to the maximal eigenvalue $\lambda_1 = 1$. The associated right eigenvector is $v_i \propto k_i$. For nonzero s, such closed form expressions are in general not known. Performing a direct matrix diagonalization is quite daunting for large system sizes N, even if one exploits methods that calculates only the first eigenvalue [16]. Hence we are interested in fast viable approximations. Here we describe one such approximation expected to be valid for networks in which vertex degrees are typically large.

Degree-based approximation. We start by considering the left eigenvectors $\mathbf{w}$ instead of the right eigenvectors, for which the eigenvalue equation can be written as

$$\lambda\, w_j = \frac{1}{k_j} \sum_{i \in \partial j} w_i\, \mathrm{e}^{s f(k_i)} \,. \tag{4}$$

This system of equations can be simplified by considering a degree-based approximation for the first eigenvector, where one assumes that the values of w_i only depend on the degree of the node i: $w_i = w(k_i)$. If the average degree is large enough and the degree distribution is not too heterogeneous, we can write the eigenvalue equation (4) by appeal to the law of large numbers as

$$\lambda_1(s)\, w(k) = \sum_{k'} P(k'|k)\, w(k')\, e^{s f(k')} \tag{5}$$

where $P(k'|k)$ is the probability for the neighbor of a node of degree k to have degree k'.

In an Erdős-Rényi (ER) ensemble [17], and more generally in any configuration model ensemble, we have $P(k'|k) = P(k') \frac{k'}{\langle k \rangle}$. In this case the right-hand side of (5) does not depend on k and the $w(k)$ are in fact k-independent. The eigenvalue equation then simplifies to

$$\lambda_1(s) = \left\langle \frac{k}{\langle k \rangle} e^{s f(k)} \right\rangle \,, \tag{6}$$

where the average is over the degree distribution $P(k)$. This approximation yields excellent results for large mean connectivities $c = \langle k \rangle$ on ER graphs, and more generally for configuration models without low degree nodes. This is illustrated in figure 1, where we plot a comparison with numerical simulations for ER graphs with $c = 30$. In figure 1 and throughout the remainder of the paper simulation results are obtained as averages over 1000 samples.

Eigenvector localization. Because of the heterogeneity of the underlying system, one finds the random walk transition matrix to exhibit localized states, both for fast and slow relaxation modes [18], even in the undeformed system, although

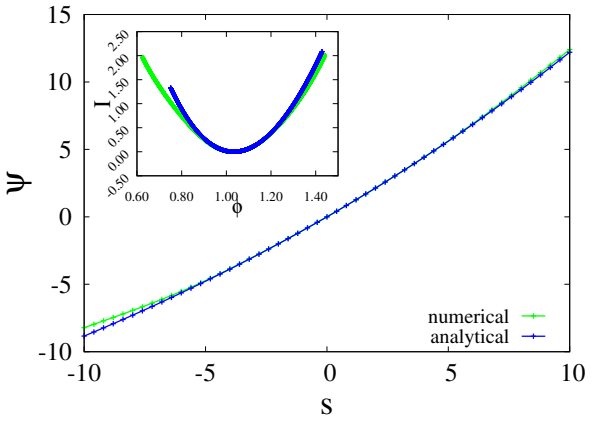


FIG. 1. (Colour online) Cumulant generating function $\psi(s)$ for ER networks with $c = 30$ and $f(k_i) = k_i/c$, comparing the large-degree approximation (6) (blue line) with results of a numerical simulation (green line). The inset shows the corresponding rate functions.

the eigenvector corresponding to the largest eigenvalue (the equilibrium distribution) is typically delocalized. However, given the nature of the deformed transition matrix, one expects the deformed random walk for large $|s|$ to be localized around vertices where $sf(k_i)$ is very large; hence we anticipate that in the deformed system, even the eigenvector corresponding to the largest eigenvalue *may* become localized for sufficiently large $|s|$. In order to investigate this effect quantitatively we look at the inverse participation ratio of the eigenvector corresponding to the largest eigenvalue λ_1 of $W(s)$. Denoting by v_i its i-th component, we have

$$\mathrm{IPR}[\mathbf{v}] = \frac{\sum_i v_i^4}{\left[\sum_i v_i^2\right]^2} \tag{7}$$

One expects $\mathrm{IPR}[\mathbf{v}] \sim N^{-1}$ for a delocalized vector, whereas $\mathrm{IPR}[\mathbf{v}] = O(1)$ if $\mathbf{v}$ is localized.

Results on random graphs. We performed numerical simulations to evaluate $\lambda_1(s)$ and the $\mathrm{IPR}[\mathbf{v}_1(s)]$ for several types of network, defined by their random graph topology. In the present letter we restrict ourselves to discussing results for ER networks. We found that other network ensembles such as scale-free random graphs give qualitatively similar results; we will report on these in an extended version of this letter.

We looked at various examples for the function $f(k_i)$ but in the present letter we only report results for the normalized degree $f(k_i) = k_i/c$; other deterministic types of degree-dependent functions exhibit similar behavior, thus focusing on the normalized degree is sufficient to capture the important aspects of this problem. We restrict our simulations to the largest (giant) component of the graphs, in order to prevent spurious effects of isolated nodes or small disconnected clusters (e.g. dimers) dominating $\lambda_1(s)$ and the IPR for negative s, as these would represent trivial instances of rare events, where a walker starts, and is thus stuck on a small disconnected component of the graph. From here on, the network size given must be understood as the size of the networks from which the giant component is extracted.

Fig. 2 shows the existence of two localized regimes for sufficiently large values of $|s|$, with IPRs on the localized side of both transitions increasing with system size. Results can be understood, as for large $|s|$ the deformed random walk is naturally attracted to the nodes with the largest (resp. smallest) degrees for positive (resp. negative) s. Thus for large negative s the deformed walk tends to be concentrated at the end of the longest dangling chain, whereas for large positive s it will be concentrated at the site with the largest available coordination. On an ER network where the large-degree tail of the degree distribution decays very fast, such a high degree vertex is likely to be connected to vertices whose degrees are lower, even significantly lower, than that of the highest degree vertex in the network, which leads to IPRs approaching 1 in the large N limit. Conversely, for negative s, the deformed random walk will be attracted to the ends of dangling chains in the network, with the probability of escape from a chain decreasing with its length (with the length of the longest dangling chain increasing with system size). This can explain that IPRs initially saturate at 1/2 for large systems. Only upon further decreasing s to more negative values will the asymmetry of the deformed transition matrices, to and away from the end of a dangling chain, induce that further weight of the dominant eigenvector to become concentrated on the end-site, leading to a further increase of the IPR.

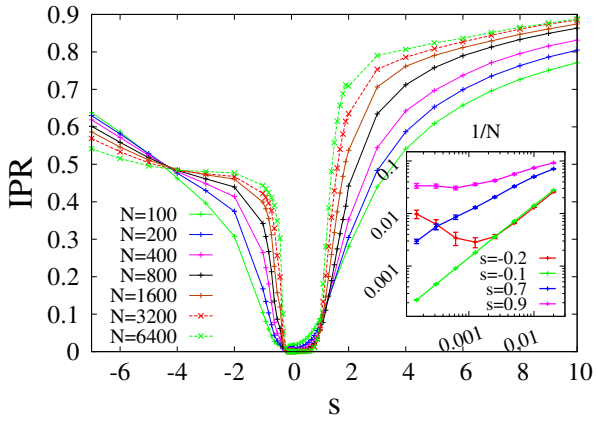


FIG. 2. IPR[v] as a function of the deformation parameter s for ER graphs with $c = 6$, and $f(k_i) = k_i/c$. The inset exhibits the N^{-1}-scaling of IPRs for 4 different values of the deformation parameter s, chosen in pairs on either side of *two* localization transitions, one at negative, and one at positive s.

From the values of $\lambda_1(s)$ we also derived the large deviation rate functions for path averages of the normalized degree $f(k_i) = k_i/c$, for various systems sizes and average connectivities. In fig. 3 we report $I(\phi)$ for an ER network at a low connectivity of $c = 3$. While the right branch of $I(\phi)$ is for large N well approximated by a parabola, our results show the emergence of a linear region on the left branch, which

becomes more pronounced as the system size is increased. This is a signature of a non-differentiable point of $\psi(s)$ at a point s^* estimated to be at $s^* = -0.060 \pm 0.002$: at this point the Gärtner-Ellis theorem cannot be used to evaluate the rate function, and the linear branch only represents the convex envelope of the true $I(\phi)$ [14]. The latter can either coincide with its convex envelope, or it can indeed be non-convex. However this information cannot be accessed by the theorem. The emergence of a jump-discontinuity in $\psi'(s)$ is due to a level crossing of the two largest eigenvalues, where the system switches between two modes that correspond to the largest eigenvalue on either side of s^*. In finite systems the crossing is an 'avoided crossing' due to level repulsion, but the two largest eigenvalues become asymptotically degenerate at s^* in the $N \to \infty$ limit, leading to a divergence of the correlation length $\xi(s) = [\ln(\lambda_1(s)/\lambda_2(s)]^{-1}$ at s^*, in close analogy with phenomenology of second order phase transitions, the divergence being logarithmic in N in the present case.

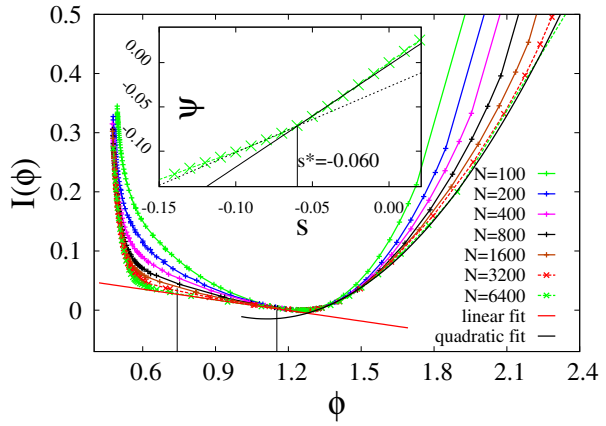


FIG. 3. Rate function $I(\phi)$ for ER graphs with $c = 3$, and $f(k_i) = k_i/c$ for system sizes ranging from $N = 100$ to $N = 6400$. In the inset, we show $\psi(s)$ in the vicinity of the non-differentiable point. For the largest system size, a linear fit of the convex envelope of the left branch and a quadratic fit of the right branch of $I(\phi)$ are shown as well.

Conclusions and future perspectives. In this Letter we have analyzed rare events statistics for path averages of observables associated with sites visited along random walk trajectories on complex networks. Results are obtained by looking at spectral properties of suitably deformed transition matrices. The main outcome of our analysis is the possible emergence of two types of dynamical phase transitions in low mean degree systems: localization transitions which entail that large deviations from typical values of path averages may be realized by localized modes of a deformed transition matrix, and *mode-switching transitions* signifying that the modes (eigenvectors) in terms of which large deviations are typically realized may switch as the deformation parameter s and thus the actual scale of large deviations are varied. Results of numerical simulations consistently support these claims. We also developed an analytical approximation valid for networks in which degrees are typically large.

Our work opens up the perspective to study a broad range of further interesting problems. On a technical level, one would want to implement more powerful techniques, such as derived in [19], to obtain the largest eigenvalue in the present problem class for larger system sizes. Then there is clearly the need to systematically study the dependence of the phenomena reported here on the degree statistics, and on the nature of the observables for which path averages are looked at. We have gone some way in this direction and will report results in an extended version of the present paper. In particular one might wish to look at observables which, rather then being deterministic functions of the degree, are only statistically correlated with the degree, or at observables taking values on *edges between* nodes [13, 14]. This could be of interest in applications such as traffic or information flows on networks subject to capacity constraints on edges. Moreover, given the nature of the mode-switching transition observed in the present letter, it is clearly conceivable that *several such transitions* could be observed in a single system, depending of course on the nature of the observables studied and on the topological properties of the underlying networks. Finally, critical phenomena associated with the localization transition and with mode-switching transitions also deserve further study. We believe that this list could go on.

This work was supported by the Marie Curie Training Network NETADIS (FP7, grant 290038).

[1] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman. Search in power-law networks. *Phys. Rev. E*, 64:046135, 2001.

[2] R. Guimerà, A. Díaz-Guilera, F. Vega-Redondo, A. Cabrales, and A. Arenas. Optimal network topologies for local search with congestion. *Phys. Rev. Lett.*, 89(24):248701, 2002.

[3] S. D. Servetto and G. Barrenechea. Constrained random walks on random graphs: routing algorithms for large scale wireless sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 12–21. ACM, 2002.

[4] B. Tadić, S. Thurner, and G. J. Rodgers. Traffic on complex networks: Towards understanding global statistical properties from microscopic density fluctuations. *Phys. Rev. E*, 69(3):036102, 2004.

[5] B. Tadić and S. Thurner. Information super-diffusion on structured networks. *Physica A: Stat. Mech. and its App.*, 332:566–584, 2004.

[6] H. Tian, H. Shen, and T. Matsuzawa. Randomwalk routing for wireless sensor networks. In *Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005. Sixth International Conference on*, pages 196–200. IEEE, 2005.

[7] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making gnutella-like p2p systems scalable. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 407–418. ACM, 2003.

[8] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. In *Proceed-*

ings of the 16th international conference on Supercomputing, pages 84–95. ACM, 2002.

[9] N. Bisnik and A. Abouzeid. Modeling and analysis of random walk search algorithms in p2p networks. In *Hot topics in peer-to-peer systems, 2005. HOT-P2P 2005. Second International Workshop on*, pages 95–103. IEEE, 2005.

[10] K. Sneppen, A. Trusina, and M. Rosvall. Hide-and-seek on complex networks. *Europhys. Lett.*, 69(5):853, 2005.

[11] V. Kishore, M. S. Santhanam, and R. E. Amritkar. Extreme events and event size fluctuations in biased random walks on networks. *Phys. Rev. E*, 85:056120, 2012.

[12] J. P. Garrahan, R. L. Jack, V. Lecomte, E. Pitard, K. van Duijvendijk, and F. van Wijland. First-order dynamical phase transition in models of glasses: an approach based on ensembles of histories. *J. Phys. A*, 42:075007, 2009.

[13] R. L. Jack and P. Sollich. Large deviations and ensembles of trajectories in stochastic models. *Progr. of Theor. Phys. Supplement*, 184:304–317, 2010.

[14] H. Touchette. The large deviation approach to statistical mechanics. *Phys. Rep.*, 478(1):1–69, 2009.

[15] F. R. Gantmacher. *Applications of the Theory of Matrices*. Interscience, New York, 1959.

[16] Cornelius C. Lanczos. *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Governm. Press Office, 1950.

[17] P. Erdős and A. Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–ñ61, 1960.

[18] R. Kühn. Spectra of random stochastic matrices and relaxation in complex systems. *Europhys. Lett.*, 109:60003, 2015.

[19] Y. Kabashima, H. Takahashi, and O. Watanabe. Cavity approach to the first eigenvalue problem in a family of symmetric random sparse matrices. *J. Phys. Conf. Ser.*, 233:012001, 2010.

Non-equilibrium statistical mechanics of the heat bath for two Brownian particles

Caterina De Bacco,[1] Fulvio Baldovin,[2] Enzo Orlandini,[2] and Ken Sekimoto[3,4]

[1]*Laboratoire de Physique Thèorique et Modèles Statistiques, CNRS et Université Paris-Sud 11, UMR8626, Bât.100, 91405 Orsay Cedex, France*
[2]*Dipartimento di Fisica e Astronomia G. Galilei and Sezione INFN, Università di Padova, Via Marzolo 8, I-35100 Padova, Italy*
[3]*Matières et Systèmes Complexes, CNRS-UMR7057, Université Paris-Diderot, 75205 Paris, France*
[4]*Gulliver, CNRS-UMR7083, ESPCI, 75231 Paris, France*

We propose a new look at the heat bath for two Brownian particles, in which the heat bath as a 'system' is both perturbed and sensed by the Brownian particles. Non-local thermal fluctuation give rise to bath-mediated static forces between the particles. Based on the general sum-rule of the linear response theory, we derive an explicit relation linking these forces to the friction kernel describing the particles' dynamics. The relation is analytically confirmed in the case of two solvable models and could be experimentally challenged. Our results point out that the inclusion of the environment as a part of the whole system is important for micron- or nano-scale physics.

PACS numbers: 05.40.Jc, 05.20.Dd, 05.40.Ca 45.20.df

Introduction — Known as the thermal Casimir interactions [1] or the Asakura-Oosawa interactions [2], a fluctuating environment can mediate static forces between the objects constituting its borders. Through a unique combination of the generalized Langevin equation and the linear response theory, we uncover a link between such interactions and the correlated Brownian motions with memory, both of which reflect the spatiotemporal non-locality of the heat bath.

The more fine details of Brownian motion are experimentally revealed, the more deviations from the idealized Wiener process are found (see, for example, [3]). When two Brownian particles are trapped close to each other in a heat bath (see Fig.1), the random forces on those objects are no more independent noises but should be correlated. Based on the projection methods [4–6] we expect the generalized Langevin equations to apply [7–10]:

$$M_J \frac{d^2X_J(t)}{dt^2} = -\frac{\partial U}{\partial X_J} - \sum_{J'=1}^{2} \int_0^t K_{J,J'}(t-\tau)\frac{dX_{J'}(\tau)}{d\tau}d\tau + \epsilon_J(t), \quad (1)$$

where X_J ($J = 1$ and 2) are the position of the Brownian particles with the mass being M_J, and $K_{J,J'}(s)$ and $\epsilon_J(t)$ are, respectively, the friction kernel and the random force. $U(X_1, X_2)$ is the static interaction potential between the Brownian particles. If the environment of the Brownian particles at the initial time $t = 0$ is in canonical equilibrium at temperature T, the noise and the frictional kernel should satisfy the fluctuation-dissipation (FD) relation of the second kind with the Onsager symmetries [7, 11]:

$$\langle \epsilon_J(t)\epsilon_{J'}(t')\rangle = k_B T\, K_{J,J'}(t-t'), \quad (2)$$

$$K_{J,J'}(s) = K_{J',J}(s) = K_{J,J'}(-s), \quad (3)$$

where J and J' are either 1 or 2 independently. This model (1) is a pivotal benchmark model for the correlated Brownian motion, although the actual Brownian motions could be more complicated (see, for example, [3, 12]). But "the physical meaning of the random force autocorrelation function is in this case far from clear..." even now and "A proper derivation of the effective potential could be of great help in clarifying this last point" [10]. In addition to the bare potential $U_0(X_1, X_2)$ independent of the heat bath, the potential U, which is in fact the free energy as function of X_J, may contain a bath-mediated interaction potential $U_b(X_1, X_2)$ so that

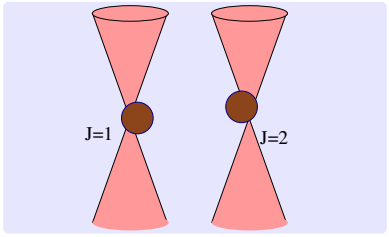


FIG. 1. Two Brownian particles (filled disks, $J = 1$ and $J = 2$) are trapped by an external potential, such as through optical traps (vertical cones), and interact through both the direct and the heat bath-mediated interactions.

$$U(X_1, X_2) = U_0(X_1, X_2) + U_b(X_1, X_2). \quad (4)$$

In this Letter we propose the relation

$$K_{1,2}(0) = -\frac{\partial}{\partial X_1}\frac{\partial}{\partial X_2}U_b(X_1, X_2), \quad (5)$$

where the both sides of this relation should be evaluated at the equilibrium positions of the Brownian particles, $X_J = \langle X_J\rangle_{\rm eq}$. This relation implies that the bath-mediated static interaction is always correlated with the frictional one. Our approach is to regard the heat bath as the weakly non-equilibrium system which is both perturbed and sensed by the mesoscopic Brownian particles. From this point of view (5) is deduced from so called 'general sum-rule theorem' [13] of the linear response theory of non-equilibrium statistical mechanics [14]. While the FD relation of the second kind (2) is well known as an outcome of this theory, the other aspects have not been fully

explored. Below we give a general argument supporting (5), and then give two analytically solvable examples for which the claim holds exactly.

General argument — While the spatial dimensionality is not restrictive in the following argument, we will use the notations as if the space were one-dimensional. Suppose we observe the force $F_{1,2}$ on the $J = 1$ particle as we move the $J = 2$ particle from $\langle X_2\rangle_{\rm eq}$ at $t = -\infty$ to $X_2(t)$ at t. Due to the small perturbation $X_2(t)-\langle X_2\rangle_{\rm eq}$, the average force at that time, $\langle F_{1,2}\rangle_t$, is deviated from its its equilibrium value, $\langle F_{1,2}\rangle_{\rm eq}$. The linear response theory relates these two through the response function, $\Phi_{1,2}(s)$ as

$$\langle F_{1,2}\rangle_t - \langle F_{1,2}\rangle_{\rm eq} = \int_{-\infty}^{t} \Phi_{1,2}(t-\tau)\,(X_2(\tau) - \langle X_2\rangle_{\rm eq})d\tau. \qquad (6)$$

(Within the linear theory the force is always measured at $X_1 = \langle X_1\rangle_{\rm eq}$.) The complex admittance $\chi_{1,2}(\omega) = \chi'_{1,2}(\omega) + i\chi''_{1,2}(\omega)$ is defined as the Fourier-Laplace transformation of $\Phi_{1,2}(s)$:

$$\chi_{1,2}(\omega) = \int_0^{+\infty} e^{i\omega s - \varepsilon s}\Phi_{1,2}(s)ds, \qquad (7)$$

where ε is a positive infinitesimal number (i.e., +0). If $\chi_{1,2}(\infty) = 0$, which is the case in the present , the causality of $\Phi_{1,2}(t)$, or the analyticity of $\chi_{1,2}(\omega)$ in the upper half complex plane of ω, impose the general sum rule [13],

$$\mathcal{P}\int_{-\infty}^{+\infty} \frac{\chi''_{1,2}(\omega)}{\omega}\frac{d\omega}{\pi} = \chi'_{1,2}(0), \qquad (8)$$

where $\mathcal{P}$ on the left hand side (l.h.s.) denotes to take the principal value of the integral across $\omega = 0$. The significance of (8) is that it relates the dissipative quantity (l.h.s.) and the reversible static response (right hand side (r.h.s.)) of the system.

Now we suppose, along the thought of Onsager's mean regression hypothesis [15], that the response of the heat bath to the fluctuating Brownian particles, which underlies (1), is essentially the same as the response to externally specified perturbations described by (6). Thus the comparison of (6) with (1) gives

$$\Phi_{1,2}(t) = -\frac{dK_{1,2}(t)}{dt}, \qquad (9)$$

or, in other words, $K_{1,2}$ is the relaxation function corresponding to $\Phi_{1,2}$. With this linkage between the Langevin description and the linear response theory, the static reversible response $\chi'_{1,2}(0)$ of the force $\langle F_{1,2}\rangle - \langle F_{1,2}\rangle_{\rm eq}$ to the static displacement $X_2 - \langle X_2\rangle_{\rm eq}$ can be identified with the r.h.s. of (5). As for the l.h.s. of (8), we can show by (9) and (7) that it is equal to $K_{1,2}(0)$. The argument presented here is to be tested both analytically/numerically and experimentally. At least for the two models presented below the claim (5) is analytically confirmed.

Solvable model I: Hamiltonian system— As the first example that confirms the relation (5) we take up a Hamiltonian model inspired by the classic model of Zwanzig [8], see Fig. 2(a). Instead of a single Brownian particle [8] we put the two Brownian particles with masses M_J ($J = 1, 2$) which interact with the 'bath' consisting of light mass 'gas' particles. While Fig. 2(a) gives the general idea, the solvable model is limited to the one-dimensional space. Each gas particle, e.g. i-th one, has a mass m_i ($\ll M_J$) and is linked to at least one of the Brownian particles, $J = 1$ or 2, through Hookean springs of the spring constant $m_i\omega_{i,J}^2 (> 0)$ and the natural length, $\ell_{i,J}$. In Fig. 2(a) these links are represented by the dashed lines. The Hamiltonian of this purely mechanical model consists of three parts, $H = H_B + H_b + H_{bB}$, with

$$H_B = \frac{P_1^2}{2M_1} + \frac{P_2^2}{2M_2} + U_0(X_1, X_2), \qquad (10)$$

$$H_b = \sum_i \frac{p_i^2}{2m_i}, \quad H_{bB} = \sum_i \frac{m_i}{2}\sum_{J=1}^{2}\omega_{i,J}^2(q_i - X_J - \ell_{i,J})^2, \qquad (11)$$

where the pairs $(X_J, P_J = M_J\,dX_J/dt)$ and $(x_i, p_i = m_i\,dx_i/dt)$ denote, respectively, the positions and momenta of the heavy (J) and light (i) particles. The Brownian particles obey the following dynamics :

$$M_J\frac{d^2X_J}{dt^2} = -\frac{\partial U_0}{\partial X_J} + \sum_i m_i\omega_{i,J}^2(q_i - X_J - \ell_{i,J}). \qquad (12)$$

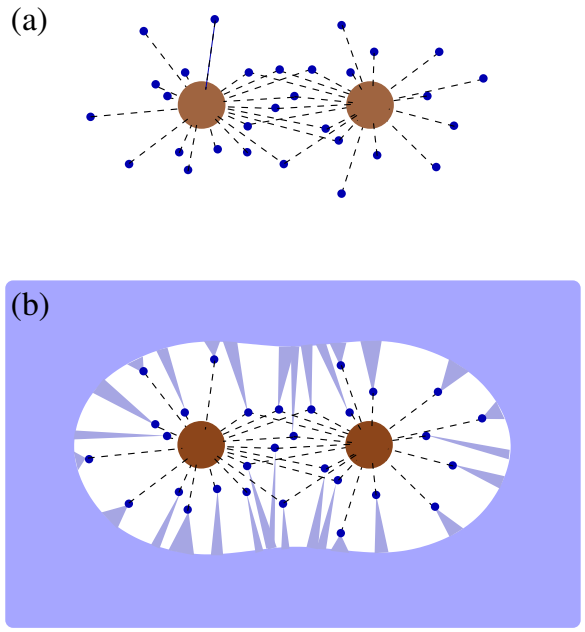


FIG. 2. *(a)* Hamiltonian model of two Brownian particles which is analytically solvable for one dimensional space with harmonic coupling. Each light mass particle (thick dot) is linked to at least one of the Brownian particles (filled disks) with Hookean springs (dashed lines). *(b)* Langevin model of two Brownian particles. Unlike the Hamiltonian model, each light mass particle receives the random force and frictional force from the background (shaded zone) and its inertia is ignored.

Given the initial values of (q_i, p_i) at $t = 0$, the Hamilton equations for $(q_i(t), p_i(t))$, which reads

$$m_i \frac{d^2 q_i}{dt^2} = -m_i \sum_{J=1}^{2} \omega_{i,J}^2 (q_i - X_J(t) - \ell_{i,J}), \tag{13}$$

can be solved in supposing that the histories of $X_J(s)$ ($J = 1$ and 2) for $0 \le s \le t$ are given. In order to assure the compatibility with the initial canonical equilibrium of the heat bath, we assume the vanishing initial velocity for the Brownian particles, $dX_J/dt|_{t=0} = 0$. Substituting each q_i in (12) by its formal solution thus obtained, the dynamics of $X_J(t)$ is rigorously reduced to (1), where the friction kernels $K_{J,J'}(s)$ are

$$K_{J,J'}(s) = \sum_i \frac{m_i \omega_{i,J}^2 \omega_{i,J'}^2}{\tilde{\omega}_i^2} \cos(\tilde{\omega}_i s), \tag{14}$$

and the noise term $\epsilon_J(t)$ is

$$\epsilon_J(t) \equiv \sum_i m_i \omega_{i,J}^2 \left\{ \tilde{q}_i(0) \cos(\tilde{\omega}_i t) + \frac{d\tilde{q}_i(0)}{dt} \frac{\sin(\tilde{\omega}_i t)}{\tilde{\omega}_i} \right\}, \tag{15}$$

with $\tilde{\omega}_i^2 \equiv \omega_{i,1}^2 + \omega_{i,2}^2$ and

$$\tilde{q}_i(t) \equiv q_i(t) - \sum_{J=1}^{2} \frac{\omega_{i,J}^2}{\tilde{\omega}_i} [\ell_{i,J} + X_J(t)]. \tag{16}$$

To our knowledge this is the first concrete model that demonstrates (1). Only those gas particles linked to the both Brownian particles satisfy $\omega_{i,1}^2 \omega_{i,2}^2 > 0$ and contribute to $K_{1,2}(s)$. While the generalized Langevin form (1) holds for an individual realization without any ensemble average, the statistics of $\epsilon_J(t)$ must be specified. We assume that at $t = 0$ the bath variables $\tilde{q}_i(0)$ and $\tilde{p}_i(0)$ (=$p_i(0)$ because we defined $dX_J/dt|_{t=0} = 0$) belong to the canonical ensemble of a temperature T with the weight $\propto \exp(-(H_b + H_{bB})/k_B T)$. Then the noises $\epsilon_J(t)$ satisfy the FD relation of the second kind (2). and the Onsager symmetries (3).

In this solvable model, the heat bath-mediated static potential U_b which supplements U_0 to make $U = U_0 + U_b$ is found to be

$$U_b(X_1 - X_2) = \frac{k_b}{2}(X_1 - X_2 - L_b)^2, \tag{17}$$

where

$$k_b = \sum_i \frac{m_i \omega_{i,1}^2 \omega_{i,2}^2}{\tilde{\omega}_i^2}, \quad L_b = \frac{1}{k_b} \sum_i \frac{m_i \omega_{i,1}^2 \omega_{i,2}^2 (\ell_{i,1} - \ell_{i,2})}{\tilde{\omega}_i^2}. \tag{18}$$

Note that U_b depends on X_1 and X_2 only through $X_1 - X_2$, that is, it possesses the translational symmetry (see later). While this form appears in the course of deriving (1), its origin can be simply understood from the following identity:

$$H_{bB} = \sum_i \frac{m_i \tilde{\omega}_i^2}{2} \tilde{q}_i^2 + U_b(X_1 - X_2). \tag{19}$$

Finally, our claim (5) is confirmed by (14) for $K_{1,2}(0)$ and by (17) and (18) for the $U_b''(X) = k_b$. In the standard language of the linear response theory, the 'displacement' A conjugate to the external parameter $X_2(t) - \langle X_2 \rangle_{\rm eq}$ is $A = \sum_i m_i \omega_{i,2}^2 (q_i - X_2 - \ell_{i,2})$ and the flux as the response is $B = \sum_i m_i \omega_{i,1}^2 (q_i - X_1 - \ell_{i,1})$ [14]. Direct calculation gives $\chi_{1,2}(\omega) = \sum_i (m_i \omega_{i,1}^2 \omega_{i,2}^2)/[\tilde{\omega}_i^2 - (\omega + i\varepsilon)^2]$.

A remark is in order about the translational symmetry of $U_b(X)$. In the original Zwanzig model [8], the factor corresponding to $q_i - X_J - \ell_{i,J}$ in (11) was $q_i - c_i X_J$ with an arbitrary constant c_i and the natural length $\ell_{i,J}$ set to be zero arbitrarily. In order that the momentum in the heat bath is locally conserved around two Brownian particles, we needed to set $c_i = 1$ and explicitly introduce the natural length $\ell_{i,J}$, especially for those gas particles which are coupled to the both Brownian particles, i.e. with $\omega_{i,1}^2 \omega_{i,2}^2 > 0$. We note that the so-called dissipative particle dynamics modeling [16–18] also respects the local momentum conservation.

Solvable model II: Langevin system.— The second example that confirms the relation (5) is constructed by modifying the first one, see Fig. 2(b). There, we replace the Hamiltonian evolution of each light mass particle (13) by the over-damped stochastic evolution governed by the Langevin equation;

$$0 = -\gamma_i \frac{dq_i}{dt} + \xi_i(t) - m_i \sum_{J=1}^{2} \omega_{i,J}^2 (q_i - \ell_{i,J} - X_J(t)), \tag{20}$$

where γ_i is the friction constant with which the i-th gas particle is coupled to a 'outer'-heat bath of the temperature T. $\xi_i(t)$ is the Gaussian white random force from the outer-heat bath obeying $\langle \xi_i(t) \rangle = 0$, and $\langle \xi_i(t) \xi_{i'}(t') \rangle = 2\gamma_i k_B T \delta(t - t') \delta_{i,i'}$. This outer-heat bath may represent those degrees of freedom of the whole heat bath which are not directly coupled to the Brownian particles, while the variables (q_i, p_i) represent those freedom of our primary interest as the 'system'. (Similar idea has already been proposed in different contexts, see [19] §6.3 and §7.1, and also [20–22].) Integrating (20) for $q_i(t)$ and substituting the result into the r.h.s. of (12), we again obtain (1) and (2) with the same bath-mediated static potential as before, i.e., U_b defined by (17) and (18). (In this over-damped model, $m_i \omega_{i,J}^2$ simply represents the spring constant between the i-th light mass and the J-th Brownian particle.) The friction kernel and the noise term of the present model are, however, different: instead of (14) and (15), they read, respectively,

$$K_{J,J'}(s) = \sum_i \frac{m_i \omega_{i,J}^2 \omega_{i,J'}^2}{\tilde{\omega}_i^2} e^{-\frac{|s|}{\tau_i}}, \tag{21}$$

$$\epsilon_J(t) = \sum_i m_i \omega_{i,J}^2 \int_0^\infty \frac{e^{-\frac{s}{\tau_i}}}{\gamma_i} \xi_i(t - s) ds, \tag{22}$$

where $\tau_i = \gamma_i/(m_i \tilde{\omega}_i^2)$. Because the form of $K_{1,2}(0)$ as well as $U_b(X)$ are unchanged from the first model, our claim (5) is again confirmed.

Discussion : Implication of (5) — Being consistent with this relation, no bath-mediated interactions appeared in the phenomenological approaches [23–25] where the Stokesian fluid model is supplemented by the thermal random forces satisfying the FD relation, because the bath had no memory.

The above solvable models, though being artificial, represent certain non-local aspects of the more realistic heat baths. The cross frictional kernel $K_{1,2}(s)$ and the bath mediated potential $U_b(X)$ are generated by those microscopic degrees of freedom which couple to both the Brownian particles. This picture is reminiscent of the quantum system interacting with electromagnetic fields (see, for example, [26]).

From operational point of view, the relation (5) implies that we cannot control the friction kernels or friction coefficients without changing the bath-mediated interaction between the Brownian particles. As a demonstration, if all the $\omega_{i,J}$ of the light particles are changed by a multiplicative factor λ, i.e. $\omega_{i,J} \mapsto \lambda\,\omega_{i,J}$, then both $K_{J,J'}(s)$ and $U_b(X)$ should be changed to $\lambda^2 K_{J,J'}(\lambda s)$ and $\lambda^2 U_b(X)$, respectively.

Especially about the work W of operations, (5) implies that the work W_K to change the off-diagonal friction kernel $K_{1,2}$ cannot be isolated from the work W_U to change the bath-mediated interaction potential, U_b. In the above solvable models, the total work, $W = W_K + W_U$, to change the parameters, $\{\omega_{i,J}\}$, can be given as the Stieltjes integrals along the time-evolution of the whole degrees of freedom:

$$W = \sum_i \sum_{J=1}^{2} \int_\Gamma \frac{\partial H_{bB}}{\partial \omega_{i,J}} d\omega_{i,J}(t), \tag{23}$$

where $\int_\Gamma$ indicates to integrate along the process where all the dynamical variables $p_{i'}, \tilde{q}_{i'}$ and X_J in the integrals evolves according to the system's dynamics under time dependent parameters $\{\omega_{i,J}\}$. The operational inseparability of the work into W_K and W_U justifies the fact that, on the level of the stochastic energetics [19], we could not access the work to change the friction coefficients. On the microscopic level, however, the above models allow to identify W_K: First W_U is given by the above framework [19]:

$$W_U = \sum_i \sum_{J=1}^{2} \int_\Gamma \frac{\partial U_b}{\partial \omega_{i,J}} d\omega_{i,J}(t), \tag{24}$$

because U_0 does not depend on $\omega_{i,J}$. Combining (24) with (23) as well as the identity (19), the kinetic part of the work, W_K, is found to be

$$W_K = \sum_i \sum_{J=1}^{2} \int_\Gamma \frac{\partial}{\partial \omega_{i,J}} \left(\sum_{i'} \frac{m_{i'} \tilde{\omega}_{i'}^2}{2} \tilde{q}_{i'}^2 \right) d\omega_{i,J}(t), \tag{25}$$

where $\tilde{q}_i$ are defined in (16). The result again shows that, unless we have an access to the microscopic fluctuations in the heat bath, W_K is not measurable.

In conclusion we propose, with supporting examples, that a bath-mediated effective potential between the Brownian particles, U_b, should accompany the off-diagonal frictional memory kernel, $K_{1,2}(s)$, with a particular relation (5) due to the general sum rule of the linear response theory. This relation should be tested experimentally and/or numerically on the one hand, and the generalization to other models [3, 12] should be explored on the other hand. For example, in the reaction dynamics of protein molecules or of colloidal particles, non-local fluctuations of the solvent may play important roles both kinetically and statically. The consciousness of the environment as a part of the whole system is important not only in the ecology but also at the micron- or nano-scale physics.

This work is supported by the Marie Curie Training Network NETADIS (FP7, grant 290038) for CDB. KS acknowledges Antoine Fruleux for fruitful discussions. CDB and KS thank ICTP (Trieste, Italy) for providing with the opportunity to start the collaboration.

[1] M. E. Fisher and P.-G. de Gennes, Comptes Rendus de lAcadmie des sciences **287**, 207 (1978).
[2] S. Asakura and F. Oosawa, The Journal of Chemical Physics **22**, 1255 (1954).
[3] R. Huang, I. Chavez, K. M. Taute, B. Lukić, S. Jeney, M. G. Raizen, and E. Florin, Nature Physics **7**, 576 (2011).
[4] H. Mori, Prog. Theor. Phys. **33**, 423 (1965).
[5] R. Zwanzig, J. Chem. Phys. **33**, 1338 (1960).
[6] R. Zwanzig, Phys. Rev. **124**, 983 (1961).
[7] K. Kawasaki, J. Phys. A **6**, 1289 (1973).
[8] R. Zwanzig, J. Stat. Phys. **9**, 215 (1973).
[9] R. Zwanzig, *Nonequilibrium Statistical Mechanics* (Oxford University Press, London, 2001).
[10] G. Ciccotti and J.-P. Ryckaert, J. Stat. Phys. **26**, 73 (1980).
[11] S. Nordholm and R. Zwanzig, J. Stat. Phys. **13**, 347 (1975).
[12] L. Bocquet and J. Piasecki, J. Stat. Phys. **87**, 1005 (1997).
[13] R. Kubo, Journal of the Physical Society of Japan **12**, 570 (1957).
[14] R. Kubo, M. Toda, and N. Hashitsume, *Statistical physics II : nonequilibrium statistical mechanics*, 2nd ed. (Springer-Verlag, Berlin, 1991).
[15] L. Onsager, Phys. Rev. **38**, 2265 (1931).
[16] P. Hoogerbrugge and K. Koelman, Europhys. Lett. **19**, 155 (1992).
[17] P. Español and P. B. Warren, Europhys. Lett. **30**, 191 (1995).
[18] C. Lowe, Europhys. Lett. **47**, 145 (1999).
[19] K. Sekimoto, *Stochastic Energetics (Lecture Notes in Physics, vol. 799)* (Springer, 2010).
[20] K. Sekimoto, F. Takagi, and T. Hondou, Phys. Rev. E **62**, 7759 (2000).
[21] A. Sarracino, D. Villamaina, G. Gradenigo, and A. Puglisi, EPL (Europhysics Letters) **92**, 34001 (2010).
[22] S. Kawai and T. Komatsuzaki, J. Chem. Phys. **134**, 114523 (2011).
[23] D. L. Ermak and J. A. McCammon, J. Chem. Phys. **69**, 1352 (1978).
[24] B. Cichocki and B. U. Felderhof, J. Chem. Phys. **94**, 556 (1991).
[25] E. R. Dufresne, T. M. Squires, M. P. Brenner, and D. G. Grier, Phys. Rev. Lett. **85**, 3317 (2000).
[26] R. P. Feynman, *Statistical mechanics: a set of lectures by R. P. Feynman*, 2nd ed. (Westview Press, 1998).