Some Applications of Graph Theory

A thesis submitted for the degree of Doctor of Philosophy

by

Nicole Eggemann

Department of Mathematical Sciences

Brunel University

September 2009

Acknowledgements

This work was funded by a Marie Curie Early Stage Training Fellowship (NET-ACE-programme) under grant number MEST-CT-2004-6724.

Foremost, I would like to give many thanks to my supervisor, Dr Steven Noble, for the opportunity to do my PhD studies under his supervision and as part of the NET-ACE-programme. I am extremely grateful for his constant encouragement, competent advice and for always giving very thorough and complete answers to my questions. In addition I would like to thank him for his very useful and detailed feedback, which was always much appreciated.

As part of the NET-ACE-programme I was able to go on two placements in France and one industrial placement in Bristol. These were great experiences and I am very grateful to Dr Steven Noble for arranging these placements.

Fortunately, I was able to do my first placement at the LaBRI of the University of Bordeaux in France and would like to thank Dr Nicolas Hanusse and Dr Philippe Duchon for allowing me to have this opportunity. Moreover, I would like to thank them for the many useful discussions that took place, which were particularly helpful for the second chapter of this thesis.

My second placement was at Advanced Transport Systems Ltd. in Bristol. I would like to thank Professor Martin Lowson for the very interesting and enjoyable experience I had whilst I was working there, for useful discussions and for describing the problem, which motivated the work in the fourth chapter of this thesis. Many thanks also go to the Engineering Mathematics department of the University of Bristol for allowing me to use their facilities, and to Dr Eddie Wilson for arranging this opportunity. In addition, I would like to thank him for the many helpful discussions that we had during that time.

I was very pleased about being able to do my final placement at the INRIA in Sophia Antipolis, which is close to Nice, in France and would like to thank Dr Frédéric Havet for this wonderful opportunity and experience. Furthermore, I would like to thank him for introducing me to the theory of the Frequency Assignment Problem and for the many valuable discussions that we had during my time there.

Moreover, I would like to thank the examiners Professor Colin Cooper, Professor Colin McDiarmid and Dr Ilia Krasikov for their comments, which helped improving parts of the thesis, in particular some of the proofs in Chapter 3.

Some Applications of Graph Theory

Nicole Eggemann

Submitted for the degree of Doctor of Philosophy

Department of Mathematical Sciences, Brunel University
Uxbridge

2009

Abstract

We investigate four different applications on graph theory. First we show for a generalisation of the well-known Barabási-Albert model that the expectation of the clustering coefficient of this graph process is asymptotically proportional to $\frac{\log n}{n}$, by generalising a result of Bollobás and Riordan.

Secondly, we investigate the complexity of searching for a given vertex in a scale-free graph, using only locally gathered information. We consider two kinds of models which are generalisations of the Barabási-Albert model, proving two lower bounds of $\Omega(n^{1/2})$ on the expected time to find the worst-case target, under a restrictive model of local information.

Thirdly, we consider two orientation problems in a graph, namely the minimisation of the sum of shortest paths lengths and the minimisation of the diameter. We show that it is NP-complete to determine whether a graph has an orientation for which the sum of shortest paths lengths is less than an integer specified in the input. Furthermore we describe an algorithm that runs in linear time and decides for a planar graph G whether there is an orientation such that the diameter of $\vec{G}$ is less than a fixed constant.

Fourthly, we consider the well-known k-$L(2,1)$-labelling which is a mapping from the vertex set of a graph $G = (V, E)$ into an interval of integers $\{0, \ldots, k\}$ such that any two adjacent vertices are mapped onto integers that are at least two apart, and every two vertices with a common neighbour are mapped onto distinct integers. We show that the k-$L(2,1)$-labelling is NP-complete for planar graphs and any $k \geq 4$ by reduction from Planar Cubic Two-Colourable Perfect Matching. Schaefer stated without proof that Planar Cubic Two-Colourable Perfect Matching is NP-complete. In this thesis we give a proof of this.

List of Figures

5.1 Planar graph H. 92
5.2 Three almost two-coloured perfect matchings of the subgraph of H induced by the vertices $c, e, f, m, i, j, k, o, q$. 93
5.3 Almost two-coloured perfect matchings of H. 94
5.4 Planar clause gadget K. 94
5.5 Uncrossing gadget U and its possible almost two-coloured perfect matchings where $\alpha, \beta \in \{\text{black}, \text{white}\}$ and α', β' are the opposite colours to α and β, respectively. 97
5.6 Sketch of the modification from the Not-All-Equal 3SAT problem to a planar cubic graph 98
5.7 The edge gadget G_5. 101
5.8 The edge gadget G_6. 103
5.9 The edge gadget G_7. 103
5.10 The graph H' for $k = 6$. 104
5.11 An auxiliary edge. 107
5.12 Good coloured orientation of an auxiliary edge if u and v receive the same colour. 108
5.13 Assignment of a good coloured orientation on H. 109
5.14 Construction of graph K from G for $k = 4$. 112

Contents

1 Introduction **6**

2 The Clustering Coefficient of a Scale-Free Random Graph **12**
- 2.1 Introduction . . . 12
- 2.2 The model of Móri . . . 15
- 2.3 Subgraphs of $G^n_{1,\beta}$ . . . 17
- 2.4 Calculation of Expectations . . . 23
  - 2.4.1 Expected Number of Triangles . . . 23
  - 2.4.2 Expectation of $\sum_{v\in V(G)} \binom{d(v)}{2}$ . . . 26
- 2.5 Concentration of $\sum_{v\in V(G)} \binom{d(v)}{2}$ . . . 30
- 2.6 Expected clustering coefficient . . . 38
- 2.7 Conclusion . . . 40

3 Non-Searchability of Random Power-Law Graphs **41**
- 3.1 Introduction . . . 41
- 3.2 Modelling the searching process . . . 44
- 3.3 Vertex equivalence . . . 45
- 3.4 Lower bound on the searching time for the Móri graph . . . 48
  - 3.4.1 The Móri model for $\beta > -1$ . . . 48
  - 3.4.2 Conditional equivalence in the Móri tree . . . 48
  - 3.4.3 Lower bound on searching time . . . 50
  - 3.4.4 Average searching time in the weak model . . . 53
- 3.5 Lower bound on searching time for the Cooper-Frieze model . . . 53
  - 3.5.1 The Cooper-Frieze model . . . 53
  - 3.5.2 A model for the construction of the Cooper-Frieze graph 55
  - 3.5.3 Vertex equivalence in the Cooper-Frieze model . . . 57
  - 3.5.4 Lower bound on searching time . . . 59
- 3.6 Optimal search algorithm for the Móri tree . . . 60
- 3.7 Conclusion and open problems . . . 65

4 The complexity of two graph orientation problems **67**
- 4.1 Introduction . . . 67
- 4.2 Complexity . . . 69

4.3 Complexity of the Wiener Index 71
4.4 Tree-decompositions 73
4.5 Algorithm DIAMETER 76
4.6 Minimising the diameter of any planar graph 86
4.7 Conclusion 88

5 k-$L(2,1)$-Labelling for Planar Graphs is NP-Complete for $k \geq 4$ 89
5.1 Preliminary results 91
5.2 k-$L(2,1)$-labelling for $k \geq 4$ fixed 99
5.2.1 $\lambda_{2,1}(G) = 4$ 100
5.2.2 $\lambda_{2,1}(G) = 5$ 100
5.2.3 $\lambda_{2,1}(G) \geq 6$ 102
5.2.4 Summary 106
5.3 k-$L(2,1)$-labelling for planar graphs is NP-complete for $k \geq 4$ 107

6 Conclusion and open problems 115

Bibliography 117

Chapter 1

Introduction

In the next two chapters of this thesis we investigate the clustering coefficient and the searchability of randomly growing scale-free graphs. The fourth chapter is dedicated to the determination of the complexity of two orientation problems on graphs. In the last part of this thesis we investigate the complexity of $L(2,1)$-labelling with span k on a planar graph for $k \geq 4$.

The research on the different topics arose during stays at two universities in France and a company in the United Kingdom as part of the NetACE programme. The collaboration on the searchability of randomly growing scale-free graphs was mainly carried out during a stay at the LaBRI of the Université Bordeaux. Later during an industrial placement at Advanced Transport System Ltd. the work on the determination of the complexity of two orientation problems arose. The investigation of the complexity of the $L(2,1)$-labelling on a planar graph was proposed during a stay at INRIA in Nice. The work on the clustering coefficient was mainly carried out at Brunel University.

Two chapters of this thesis are dedicated to the investigation of properties of scale-free graphs. These are graphs which have a degree distribution obeying a power-law, that is given a graph on n vertices the number of vertices of degree x is proportional to $n\left(\frac{1}{x}\right)^{\ell}$ for $\ell > 1$ [4]. For real-life networks, ℓ tends to belong to the range $[1,3]$ [4]. There is a vast literature dealing with this kind of networks and there is great interest in analysing these networks as many real-world networks in social and natural science tend to exhibit a

power-law [4]. Many examples for these networks can be found in the survey paper [4].

There are different theoretical models which describe random graphs obeying power-laws. One of these models is the model introduced by Chung and Lu [16, 17, 18], where a number of vertices is given and to each vertex a weight is associated. Then between each pair of vertices an edge appears with probability equal to the quotient of the product of the weights of these vertices and the sum of all weights. If the distribution of the weights obeys a power-law then so does asymptotically the degree distribution [17]. Another type of model is the randomly growing graph model. Unlike the model introduced by Chung and Lu, the randomly growing graph model gives an idea of the underlying evolving processes which yield scale-free networks, namely growth and preferential attachment [3]. In this thesis we focus on the latter kind of model.

In 1999 Barabási and Albert [3] described an evolving model where in each time step a new node with m out-going edges is added to the existing graph and the out-going edges are connected to the already existing vertices with probability proportional to their degrees. This model was described in a precise mathematical way by Bollobás and Riordan in [11] where it was proven that this evolving process yields a power-law degree distribution. One common generalisation of this model found in literature (for example see [10]) is when the probability for the choice of the out-going edges is proportional to the degree of the existing vertices plus a constant. In this thesis we work with such a model as described by Móri [61]. Móri allows any constant $\beta > -1$ and showed that a graph obtained by this process exhibits a power-law with exponent $3 + \beta$. Moreover Móri showed in [62] that with high probability for any $\epsilon > 0$ the maximum degree of the Móri graph is $O(n^{\frac{1}{2+\beta}+\epsilon})$. This result will be essential in our investigation of the searchability and the clustering coefficient of the Móri graph. Another model for a randomly growing graph which we will investigate was introduced by Cooper and Frieze [20] and is an even more general model than the Móri model. In this model the number of newly created edges at each time step is given by a random variable. These edges are out-going of either a newly created vertex or an old vertex

depending on random variables. These out-going edges choose their parent vertices either uniformly at random or preferentially depending on another random variable. Cooper and Frieze [20] showed that this model yields scale-free graphs.

Another classification for real-world networks is the notion of 'small-world' networks. In 1998 Watts and Strogatz [79] described small-world networks as those having both small average paths length and being highly clustered. As described in [79] many real-world networks are small-world, e.g. the power grid of the western United States and the collaboration graph of film actors. Watts and Strogatz defined in [79] the local clustering coefficient of a vertex as the proportion of pairs of neighbours of a vertex that are neighbours themselves. Then they defined the clustering coefficient of the graph as the average over all local clustering coefficients. In [10] Bollobás and Riordan defined the clustering coefficient as a weighted average of the local clustering coefficients where the weights are the maximal possible number of links between neighbours. As seen from [10] this definition is equivalent to defining the clustering coefficient as the quotient of three times the number of triangles in the graph and the number of pairs of adjacent edges.

In Chapter 2 we determine the clustering coefficient of the Móri graph for $\beta > 0$. For $\beta = 0$ the Móri graph process corresponds to the Barabási-Albert model [3]. Bollobás and Riordan [10] showed that if $\beta = 0$ the expected clustering coefficient is asymptotically proportional to $(\log n)^2/n$. We generalise their result and show that for $\beta > 0$ the expected clustering coefficient is asymptotically proportional to $\log n/n$. Interestingly the clustering coefficient obeys a discontinuity at $\beta = 0$ and since it converges to zero when n goes to infinity the Móri graphs are not small-world.

This chapter is joint work with Steven Noble.

In Chapter 3 we investigate the expected searching time in scale-free randomly evolving graphs. We consider the Móri model and the Cooper-Frieze model and investigate the expected number of steps needed by searching processes for the worst case choice of initial and target vertex. We are interested in a lower bound on the expected searching time in such a scale-free graph for two different searching model. In both models the process starts with

the knowledge of the identity of the initial vertex and whenever a new vertex is explored the process gets to know the edges adjacent to the explored vertex and in the *strong model* the identities of the vertices adjacent to these edges while in the *weak model* the process does not know the identities of the vertices adjacent to these edges. In the Móri model for $\beta > -1$ and the Cooper-Frieze model we are able to prove that in expectation the searching process needs to explore at least $\Omega(\sqrt{n})$ vertices before reaching the target vertex n in the weak model. We will show that due to Móri's result on the maximum degree [62] we are able to give a lower bound of $\Omega(n^{\frac{\beta}{2\beta+4}-\epsilon})$ for the Móri model with $\beta > 0$ and $\epsilon > 0$ for a searching process in the *strong model*. Moreover for the Móri tree for $\beta > 0$ in which each vertex has exactly one out-going edge we present a searching strategy which operates in the strong searching model and needs in expectation $\Theta(n)$ steps. We show that for any other searching strategy in the strong model the number of steps needed to find the target stochastically dominates the number of steps needed to find the target by the strategy we present. From these results we can conclude that the graphs given by the Móri model and Cooper-Frieze model are not easily searchable.

This chapter is joint work with Philippe Duchon, Nicolas Hanusse and Steven Noble. A preliminary version of this chapter was briefly announced in [25] and is published in [26].

Chapter 4 is about the complexity of two orientation problems. The problems appear in urban light rail networks where one is confronted with a network consisting of a number of stations and track linking them. This track between two stations can only be directed in one direction and crossings are not allowed. The question arising is how this track has to be directed so that the average or maximal travel time is minimised. Since the distances between the stations are short this problem can be considered as a planar nonweighted graph where vertices represent the stations and the track linking them the edges. Then the problem transforms to finding an orientation of a planar graph so that the sum of shortest paths length or diameter is minimised. The sum of shortest paths length is also called Wiener Index, a notion appearing mainly in chemistry. The problem concerning the minimal Wiener Index of

a planar graph remains an open problem but we show that given any graph it is NP-complete to decide whether the Wiener Index is at most a constant specified in the input. The proof is a reduction from a result by Chvátal and Thomassen [19] who showed that it is NP-complete to decide whether a graph can be oriented such that its diameter is at most two. This reduction is very short and most of the chapter is devoted to the problem concerning the question whether there is an orientation of a planar graph so that the diameter is less than a constant. We show that this problem is solvable in linear time. The proof involves results on tree-decomposition [72, 73, 74] and graph minor theory [56]. In order to prove the statement we describe an algorithm that takes as an input a graph G and a tree-decomposition of G with small tree-width and decides in linear time whether the graph G has diameter at most l. If there is no tree-decomposition of small tree-width we are able to prove, for a planar graph, that the diameter is greater than l by graph minor theory.

This chapter is joint work with Steven Noble.

In Chapter 5 we investigate the complexity of the planar $L(2,1)$-labelling problem of span k for $k \geq 4$. An $L(2,1)$-labelling of span k of a graph G is a mapping from the vertex set of G into an interval of integers $\{0,\ldots,k\}$ such that any two adjacent vertices are mapped onto integers that are at least 2 apart, and every two vertices with a common neighbour vertex are mapped onto distinct integers. This problem appears in broadcasting networks when radio transmitters need to be assigned frequencies with the aim of avoiding interference and minimising bandwidth [67]. Then in the corresponding graph model of this problem the vertices represent the transmitters and the edges express which pairs of transmitters are too close to each other so that an undesired interference may occur, even if the frequencies assigned to them differ by 1. This model was introduced by Roberts [67] and since then the concept has been intensively studied (see the survey of Yeh [82]). In [30] it was shown that the problem is NP-complete for general graphs. It was shown by Bodlaender *et al.* [7] that the problem is still NP-complete for planar graphs and even $k \geq 8$. In this chapter we show that it is NP-complete for any $k \geq 4$. We show this result by a reduction from Not-All-Equal 3SAT which was

proven by Schaefer [76] to be NP-complete. In order to obtain this reduction we first give a proof that Planar Cubic Perfect Two-Colourable Matching is NP-complete. Schaefer [76] mentioned that Not-All-Equal 3SAT can be reduced to Planar Cubic Two-Colourable Perfect Matching and therefore that Planar Cubic Two-Colourable Perfect Matching is NP-complete but did not give details of the proof. The first part of Chapter 5 contains a reduction of Not-All-Equal 3SAT to Planar Cubic Two-Colourable Perfect Matching. In the second part of the chapter for any $k \geq 4$ we reduce Planar Cubic Two-Colourable Perfect Matching to $L(2,1)$-labelling with span k for planar graphs and show in this manner that $L(2,1)$-labelling with span k for planar graphs is NP-complete.

This chapter is joint work with Frédéric Havet and Steven Noble.

Chapter 2

The Clustering Coefficient of a Scale-Free Random Graph

2.1 Introduction

This chapter is joint work with Steven Noble, Brunel University. Recently there has been a great deal of interest in the structure of real world networks, especially the internet. Many mathematical models have been proposed: most of these describe graph processes in which new edges are added by some form of preferential attachment. There is a vast literature discussing empirical properties of these networks but there is also a growing body of more rigorous work. A wide-ranging account of empirical properties of networks can be found in [4]; a good survey of rigorous results can be found in [10] or in the recent book [27].

In [79] Watts and Strogatz defined 'small-world' networks to be those having small path length and being highly clustered, and discovered that many real world networks are small-world networks, e.g. the power grid of the western USA and the collaboration graph of film actors.

There are conflicting definitions of the clustering coefficient appearing in the literature. See [10] for a discussion of the relationships between them.

We define the clustering coefficient, $C(G)$ of a graph G as follows:

$$C(G) = \frac{3 \times \text{ number of triangles in } G}{\sum_{v \in V(G)} \binom{d(v)}{2}},$$

where $d(v)$ is the degree of vertex v.

The reason for the three in the numerator is to ensure that the clustering coefficient of a complete graph is one. This is the maximum possible value for a simple graph. However our graphs will not be restricted to simple graphs and so the clustering coefficient can exceed one. For instance if we take three vertices and join each pair by m edges then the clustering coefficient is $m^2/(2m-1)$. Note that the clustering coefficient of a graph with at most m edges joining any pair of vertices is at most m.

In this chapter we establish rigorous results describing the asymptotic behaviour of the clustering coefficient for one class of model. Our graph theoretic notation is standard. Since our graphs are growing, we let $d_t(v)$ denote the total degree of vertex v at time t. Sometimes we omit t when the context is clear.

The Barabási–Albert model (BA model) [3] is perhaps the most widely studied graph process governed by preferential attachment. A new vertex is added to the graph at each time-step and is joined to m existing vertices of the graph chosen with probabilities proportional to their degrees. A key observation [3] is that in many large real-world networks, the proportion of vertices with degree d obeys a power-law, that is the number of vertices of degree d is proportional to $n\left(\frac{1}{d}\right)^{\ell}$ for an n-vertex graph and ℓ a constant. Graphs having this property are often called *scale-free graphs*. Moreover, it has been observed [4] that for real-life networks, ℓ tends to belong to the range $[1, 3]$.

In [11] Bollobás *et al.* gave a mathematically precise description of the BA model and showed rigorously that for $d \leq n^{\frac{1}{15}}$, the proportion of vertices with degree d asymptotically almost surely obeys a power-law. Furthermore, in [9] it was proved for $m \geq 2$ that the graph is connected with high probability and that the diameter is asymptotically $\log n / \log\log n$ while for $m = 1$ the diameter of the largest component is approximately $\log n$ (see also [10]). A

dominating set of a graph is a set S of vertices such that every member of V is adjacent to a member of S. Cooper *et al.* [21] showed that the size of the smallest dominating set for graphs obtained by the BA model is $\Omega(n)$.

The most natural generalisation of the BA model is to take the probability of attachment to v at time $t+1$ to be proportional to $d_t(v)+a$, where a is a constant representing the inherent attractiveness of a vertex. Buckley and Osthus [13] generalised the results in [11] to the case where the attractiveness is a positive integer. Many more results on these variations of the basic preferential model can be found in [10]. Other variants of the BA model have been proposed [81, 23, 24] and in all of them, scale-free random graphs are obtained. Extensions to more general models combining uniform and preferential attachment have been studied in [61, 62, 20, 55, 53]. In these papers, roughly speaking each new vertex chooses with respective probabilities p and $1-p$ whether to use uniform or preferential attachment. Depending on the value of the parameter p, the power-law distribution can be observed and is proved by Cooper and Frieze [20] (in a very general model, which we describe fully in the next chapter, in which extra links between old vertices can also be added) or by Móri [61, 62]. In these last models, the authors also give the asymptotic limit of the maximum degree, which tends to be of order n^c for some explicit constant c depending on the model parameters (but not equal to what one would obtain by extrapolating the power-law up to $\Pr(d) = 1/n$).

Bollobás and Riordan showed [10] that the expectation of the clustering coefficient of the model from [11] is asymptotically proportional to $(\log n)^2/n$. They also considered in [10] a slight variant of the model from [11]. Their results imply that for this model the expectation of the clustering coefficient is also asymptotically proportional to $(\log n)^2/n$. We work with a model depending on two parameters β, m, which to the best of our knowledge was first studied rigorously by Móri in [61]. In a sense, that we make precise in the next section, Bollobás and Riordan's model is almost the special case of Móri's model corresponding to $\beta = 0$.

Our main result is to show that for $\beta > 0$, asymptotically the expectation of the clustering coefficient is proportional to $\log n/n$. The main strategy of

our proof follows [10] and we use very similar notation. In Section 2.2 we give a definition of the model that we use and explain its relationship with the model studied in [10]. Section 2.3 contains results that give the probability of the appearance of a small subgraph. We obtain the expectation of the number of triangles appearing and of $\sum_v \binom{d(v)}{2}$ in Section 2.4. These two sections follow [10] quite closely. The overall aim is to express the expectation of the clustering coefficient as three times the quotient of the expectation of the number of triangles and the expectation of $\sum_v \binom{d(v)}{2}$. We justify doing this in Section 2.6 and make use of a concentration result proved in Section 2.5 using martingale methods. Bollobás and Riordan [10] used a similar strategy and mentioned that they also used martingale methods.

2.2 The model of Móri

We now describe in detail Móri's generalisation of the BA model [62]. Our definition involves a finer probability space than was described in [62] but the underlying graph process $(G^n_{m,\beta})$ is identical. The process depends on two parameters: m the outdegree of each vertex except the first and $\beta \in \mathbb{R}$ such that $\beta > 0$. (In [62], Móri imposed the weaker condition that $\beta > -1$). We emphasise that β is fixed and in particular does not depend on n.

We first define the process when $m = 1$. Let $G^1_{1,\beta}$ consist of a single vertex v_1 with no edges. The graph $G^{n+1}_{1,\beta}$ is formed from $G^n_{1,\beta}$ by adding a new vertex v_{n+1} together with a single directed edge e. The tail of e is v_{n+1} and the head is determined by a random variable f_{n+1}. We diverge slightly from [62] in our description of f_{n+1}.

Label the edges of $G^n_{1,\beta}$ with $e_2, \ldots, e_n$ so that e_i is the unique edge whose tail is v_i. Now let

$$\Omega_{n+1} = \{(1, v), \ldots, (n, v), (2, h), \ldots, (n, h), (2, t), \ldots, (n, t)\},$$

where the meaning of the elements in Ω_{n+1} will become clear shortly. We

define f_{n+1} to take values in Ω_{n+1} so that for $1 \leq i \leq n$,

$$\Pr(f_{n+1} = (i, v)) = \frac{\beta}{(2+\beta)n - 2} \tag{2.1}$$

and for $2 \leq i \leq n$,

$$\Pr(f_{n+1} = (i, h)) = \Pr(f_{n+1} = (i, t)) = \frac{1}{(2+\beta)n - 2}. \tag{2.2}$$

The head of the new edge added to the graph at time $n+1$ is called the *parent vertex* of v_{n+1} and is determined as follows. If $f_{n+1} = (i, v)$ then the parent vertex is v_i and we say that the choice of parent vertex has been made *uniformly*. If $f_{n+1} = (i, h)$ then the parent vertex is the head of e_i and if $f_{n+1} = (i, t)$ then the parent vertex is the tail of e_i, that is v_i. When one of the last two cases occurs, we say that the choice of parent vertex has been made *preferentially* by copying the head or tail, as appropriate, of e_i. Suppose we think of an edge as being composed of two half-edges so that each half-edge retains one endpoint of the original edge. Then the parent vertex is chosen, either by choosing one of the n vertices of $G^n_{1,\beta}$ uniformly at random or by choosing one of the $2n-2$ half-edges of $G^n_{1,\beta}$ uniformly at random and selecting the vertex to which the half-edge is attached.

The definition implies that for $1 \leq i \leq n$, the probability that the parent vertex of v_{n+1} is v_i is equal to

$$\frac{d_n(v_i) + \beta}{(2+\beta)n - 2}. \tag{2.3}$$

We might have defined f_{n+1} to be a random variable denoting the index of the parent vertex of v_{n+1} and taking probabilities as given in (2.3). Indeed for much of the sequel we will abuse notation and assume that we did define f_{n+1} in this way. However it is useful to have the finer definition when we prove the concentration results in Section 2.5.

We extend this model to a random graph process $(G^n_{m,\beta})$ for $m > 1$ as follows: run the graph process $(G^t_{1,\beta})$ and form $G^n_{m,\beta}$ by taking $G^{nm}_{1,\beta}$ and merging the first m vertices to form v_1, the next m vertices to form v_2 and so on.

Notice that our definition will not immediately extend to the case $-1 < \beta \leq 0$ because when $n = 1$, the denominator of the expression in (2.3) is at most zero and so the process cannot start. One way to get around this problem is to define $G^2_{1,0}$ to be the graph with two vertices joined by a single edge and then let the process carry on from there (as in [61, 62]). We need to use this construction in Section 3.4. In the same way the finer definition in (2.1) and (2.2) can be extended to the case $\beta = 0$. We require this in Section 3.6. A second possibility used in [10], is to attach an artificial half-edge to v_1 at the beginning. This half-edge remains present all through the process so that the sum of the vertex degrees at time n is $2n - 1$ rather than $2n - 2$ as in the model we use. However it turns out that the choice of which alternative to use makes no difference to the asymptotic form of the expectation of the clustering coefficient and so the results from [10] are directly comparable with ours.

In the following we only consider properties of the underlying undirected graph. However, it is helpful to have the extra notation and terminology of directed graphs to simplify the reading of some of the proofs.

2.3 Subgraphs of $G^n_{1,\beta}$

Let S be a labelled directed forest, in which each vertex has either one or no out-going edge and each directed edge (v_i, v_j) has $i > j$. The restrictions on S are precisely those that ensure that S can occur as a subgraph of the evolving Móri tree with $m = 1$. We call such an S a *possible forest*.

In this section we generalise the calculation in [10] to calculate the probability that such a graph S is a subgraph of $G^n_{1,\beta}$ for $\beta > 0$. We will follow the method and notation of [10] closely.

We emphasise that we are not computing the probability that $G^n_{1,\beta}$ contains a subgraph isomorphic to S; the labels of the vertices of S must correspond to the vertex labels of $G^n_{1,\beta}$ for S to be considered to be a subgraph of $G^n_{1,\beta}$.

Denote the vertices of S by $v_{s_1}, \ldots, v_{s_k}$, where $s_j < s_{j+1}$ for $1 \leq j \leq k-1$.

Furthermore, let

$$V^- = \{v_i \in V(S) : \text{ there is a } j > i \text{ such that } (v_j, v_i) \in E(S)\}$$

and

$$V^+ = \{v_i \in V(S) : \text{ there is a } j < i \text{ such that } (v_i, v_j) \in E(S)\}.$$

Let $d_S^{\text{in}}(v)$ ($d_S^{\text{out}}(v)$) denote the in-degree (out-degree) of v in S. In particular, $d_S^{\text{out}}(v)$ is either zero or one. For $t \geq i$, let $R_t(i) = |\{j > t : (v_j, v_i) \in E(S)\}|$. Observe that $R_i(i) = d_S^{\text{in}}(v_i)$. Moreover, let $c_S(i) = \sum_{k=1}^{i-1} R_{i-1}(k)$. Hence $c_S(i)$ is the number of edges in $E(S)$ from $\{v_i, \ldots, v_n\}$ to $\{v_1, \ldots, v_{i-1}\}$.

Lemma 2.4. *Let $\beta > 0$ and S be a possible forest. Then for $t \geq s_k$ the probability that S is subgraph of $G_{1,\beta}^t$ is given by*

$$\Pr(S \subset G_{1,\beta}^t) = \frac{\beta}{\beta + d_S^{\text{in}}(v_1)} \prod_{\substack{1 \leq i \leq t: \\ v_i \in V^-(S)}} \frac{\Gamma(1 + \beta + d_S^{\text{in}}(v_i))}{\Gamma(1+\beta)} \cdot \prod_{\substack{1 < i \leq t: \\ v_i \in V^+}} \frac{1}{(2+\beta)(i-1) - 2} \prod_{\substack{1 < i \leq t: \\ v_i \notin V^+}} \left(1 + \frac{c_S(i)}{(2+\beta)(i-1) - 2}\right).$$

Proof. The proof is a generalisation of the proof for the analogous result in the case $\beta = 0$ in [10] but we include it for completeness.

Let S_t be the subgraph of S induced by the vertices $\{v_1, \ldots, v_t\} \cap V(S)$. We need to define the following random variables

$$X_t = \prod_{(v_l, v_j) \in E(S_t)} I_{(v_l, v_j) \in E(G_{1,\beta}^t)} \prod_{i \leq t} \frac{\Gamma(d_t(v_i) + \beta + R_t(i))}{\Gamma(d_t(v_i) + \beta)}$$

and

$$Y_t = \prod_{(v_l, v_j) \in E(S_{t+1})} I_{(v_l, v_j) \in E(G_{1,\beta}^{t+1})} \prod_{i \leq t} \frac{\Gamma(d_{t+1}(v_i) + \beta + R_{t+1}(i))}{\Gamma(d_{t+1}(v_i) + \beta)},$$

where I_A is the indicator of the event A.

Note that $d_t(v_j)$ for $1 \leq j \leq t$ and X_t are functions of the random variables $f_2, \ldots, f_t$ while Y_t is a function of the random variables $f_2, \ldots, f_{t+1}$. However, for all j, $R_t(j)$ is deterministic.

Observe that

$$X_{t+1} = \frac{\Gamma(d_{t+1}(v_{t+1}) + \beta + R_{t+1}(t+1))}{\Gamma(d_{t+1}(v_{t+1}) + \beta)} Y_t = \frac{\Gamma(1 + \beta + R_{t+1}(t+1))}{\Gamma(1+\beta)} Y_t.$$

First, assume that there is no $r \leq t$ such that $(v_{t+1}, v_r) \in E(S)$ and so the new edge added at time $t+1$ cannot belong to S. This implies that for $i \leq t$, $R_t(i) = R_{t+1}(i)$ and $\prod_{(v_l,v_j)\in E(S_t)} I_{(v_l,v_j)\in E(G^t_{1,\beta})} = \prod_{(v_l,v_j)\in E(S_{t+1})} I_{(v_l,v_j)\in E(G^{t+1}_{1,\beta})}$. Furthermore for all $i \leq t$ with $i \neq f_{t+1}$, we have $d_{t+1}(v_i) = d_t(v_i)$. We also have $d_{t+1}(v_{f_{t+1}}) = d_t(v_{f_{t+1}}) + 1$.

For the moment fix $f_2, \ldots, f_t$ so that X_t is completely determined. Now,

$$Y_t = \left(1 + \frac{R_t(f_{t+1})}{d_t(v_{f_{t+1}}) + \beta}\right) X_t.$$

Thus

$$\begin{aligned}
\mathbf{E}\left[Y_t - X_t | f_2, \ldots, f_t\right] &= \sum_{r=1}^{t} \frac{R_t(r)}{d_t(v_r) + \beta} \Pr(f_{t+1} = r) X_t \\
&= \frac{\sum_{r=1}^{t} R_t(r)}{(2+\beta)t - 2} X_t.
\end{aligned}$$

By taking expectation with respect to $f_2, \ldots, f_t$ we obtain

$$\mathbf{E}\left[Y_t\right] = \left(1 + \frac{\sum_{r=1}^{t} R_t(r)}{(2+\beta)t-2}\right) \mathbf{E}\left[X_t\right] = \left(1 + \frac{c_S(t+1)}{(2+\beta)t-2}\right) \mathbf{E}\left[X_t\right]$$

and

$$\mathbf{E}\left[X_{t+1}\right] = \frac{\Gamma(1+\beta+R_{t+1}(t+1))}{\Gamma(1+\beta)} \left(1 + \frac{c_S(t+1)}{(2+\beta)t-2}\right) \mathbf{E}\left[X_t\right]. \tag{2.5}$$

Now suppose (v_{t+1}, v_r) is an edge of S for some $r < t+1$. If $f_{t+1} \neq r$ then $X_{t+1} = 0$ so we will suppose that $f_{t+1} = r$. Then for all $i \leq t$ with $i \neq r$, $d_{t+1}(v_i) = d_t(v_i)$, and $d_{t+1}(v_r) = d_t(v_r) + 1$. Furthermore for all $i \leq t, i \neq r$ $R_{t+1}(i) = R_t(i)$, but $R_{t+1}(r) = R_t(r) - 1$.

Hence providing $f_{t+1} = r$, we have

$$Y_t = \frac{1}{d_t(v_r) + \beta} X_t.$$

So

$$\mathbf{E}\left[Y_t | f_2, \ldots, f_t\right] = \frac{d_t(v_r) + \beta}{(2+\beta)t - 2} \frac{X_t}{d_t(v_r) + \beta} = \frac{X_t}{(2+\beta)t - 2}.$$

Thus

$$\mathbf{E}\left[X_{t+1} | f_2, \ldots, f_t\right] = \frac{1}{(2+\beta)t - 2} \frac{\Gamma(1 + \beta + R_{t+1}(t+1))}{\Gamma(1+\beta)} X_t.$$

So by taking expectation with respect to $f_2, \ldots, f_t$,

$$\mathbf{E}\left[X_{t+1}\right] = \frac{1}{(2+\beta)t - 2} \frac{\Gamma(1 + \beta + R_{t+1}(t+1))}{\Gamma(1+\beta)} \mathbf{E}[X_t]. \tag{2.6}$$

Note that $X_1 = \frac{\Gamma(\beta + R_1(1))}{\Gamma(\beta)}$ and that for $t \geq s_k$, we have $\Pr(S \subset G^t_{1,\beta}) = \mathbf{E}\left[X_t\right]$. Using (2.5) and (2.6) and noting that $R_i(i) = 0$ for $v_i \notin V^-$, we have for $t \geq s_k$

$$\begin{aligned}\Pr(S \subset G^t_{1,\beta}) = \frac{\Gamma(\beta + R_1(1))}{\Gamma(\beta)} \prod_{\substack{1 < i \leq t: \\ v_i \in V^-}} \frac{\Gamma(1 + \beta + R_i(i))}{\Gamma(1+\beta)} \\ \cdot \prod_{\substack{1 < i \leq t: \\ v_i \in V^+}} \frac{1}{(2+\beta)(i-1) - 2} \prod_{\substack{1 < i \leq t: \\ v_i \notin V^+}} \left(1 + \frac{c_S(i)}{(2+\beta)(i-1) - 2}\right).\end{aligned}$$

This is easily seen to be equivalent to the expression in the statement of the lemma. □

We now provide a more convenient form for the probability given in Lemma 2.4. This calculation is very similar to the analogous one in [10]. Before stating this result we recall three elementary estimates.

Lemma 2.7. *1. For all $x \geq 0$, $\log(1+x) = x + O(x^2)$.*

2. For all $n_1, n_2 \in \mathbb{Z}$ with $1 \leq n_1 \leq n_2$, $\sum_{k=n_1}^{n_2} 1/k = \log(n_2/n_1) + O(1/n_1)$.

3. *Let $a > 0$ and let $a' > a$. Then for all $x \geq a'$ we have $-\log(1-a/x) \leq \frac{aa'}{(a'-a)x}$.*

Lemma 2.8. *Let $\beta > 0$ and S be a possible forest. Then for $t \geq s_k$ the probability that S is a subgraph of $G^t_{1,\beta}$ is given by*

$$\begin{aligned}
&\Pr(S \subset G^t_{1,\beta}) \\
&\quad = \frac{\beta}{d^{\text{in}}_S(v_1)+\beta} \prod_{i:v_i \in V^-} \frac{\Gamma(1+d^{\text{in}}_S(v_i)+\beta)}{\Gamma(1+\beta)} \\
&\qquad \cdot \prod_{(v_i,v_j)\in E(S):i>j} \frac{1}{(2+\beta)(i^{1+\beta}j)^{1/(2+\beta)}} \exp\left(O\left(\sum_{j=2}^{k} c_S(s_j)^2/(j-1)\right)\right).
\end{aligned}$$

Proof. First note that if $c \geq 0$ then by using Lemma 2.7 we have

$$\begin{aligned}
&\log\left(\prod_{i=s+1}^{t}\left(1+\frac{c}{(2+\beta)(i-1)-2}\right)\right) \\
&\quad = \sum_{i=s+1}^{t}\left(\frac{c}{(2+\beta)(i-1)-2} + O\left(\frac{c^2}{((2+\beta)(i-1)-2)^2}\right)\right) \qquad (2.9) \\
&\quad = \frac{c}{2+\beta}\log(t/s) + O(c^2/s).
\end{aligned}$$

Furthermore (2.9) remains true if we replace the upper limit of the sum on the left-hand side with $t-1$. Let D be the final term in the expression in Lemma 2.4, that is,

$$D = \prod_{\substack{1<i\leq t:\\ v_i \notin V^+}} \left(1+\frac{c_S(i)}{(2+\beta)(i-1)-2}\right).$$

If $i \leq s_1$, then $c_S(i) = 0$, and if $s_j + 1 \leq i \leq s_{j+1}$, then $c_S(i) = c_S(s_{j+1})$. So

by using (2.9), we obtain

$$\log D = \log\left(\prod_{j=2}^{k}\prod_{\substack{i\notin V^+:\\ s_{j-1}+1\le i\le s_j}}\left(1+\frac{c_S(s_j)}{(2+\beta)(i-1)-2}\right)\right)$$
$$= \sum_{j=2}^{k}\frac{c_S(s_j)}{2+\beta}\log(s_j/s_{j-1}) + O\left(\sum_{j=2}^{k}\frac{c_S(s_j)^2}{s_{j-1}}\right).$$

We now need the following simple calculation.

$$\frac{1}{(2+\beta)(i-1)-2} = \frac{1}{(2+\beta)i}\cdot\frac{1}{1-\frac{4+\beta}{(2+\beta)i}}$$
$$= \frac{1}{(2+\beta)i}\cdot\exp\left(-\log\left(1-\frac{4+\beta}{(2+\beta)i}\right)\right).$$

Hence, providing $i \ge 2$, we may use the final part of Lemma 2.7 to obtain

$$\frac{1}{(2+\beta)(i-1)-2} = \frac{1}{(2+\beta)i}\cdot\exp(O(1/i)).$$

Note that whenever $v_i \in V^+$ we have $i = s_j$ for some j such that $c_S(s_j) \ge 1$. So we have

$$\Pr(S\subset G^t_{1,\beta}) = \frac{\beta}{\beta+d^{\text{in}}_S(v_1)}\prod_{i:v_i\in V^-}\frac{\Gamma(1+\beta+d^{\text{in}}_S(v_i))}{\Gamma(1+\beta)}\prod_{i:v_i\in V^+}\frac{1}{(2+\beta)i}$$
$$\prod_{j=2}^{k}(s_j/s_{j-1})^{c_s(s_j)/(2+\beta)}\exp\left(O\left(\sum_{j=2}^{k}c_S(s_j)^2/s_{j-1}\right)\right). \tag{2.10}$$

To complete the proof we must rewrite the third and fourth terms of (2.10). In order to do so we need to know the exponents of s_i for each i. The exponent of s_1 is $-c_S(s_2)/(2+\beta)$, the exponent of s_k is $c_S(s_k)/(2+\beta) - d^{\text{out}}_S(v_{s_k})$ and for $2 \le i \le k-1$, the exponent of s_i is $(c_S(s_i) - c_S(s_{i+1}))/(2+\beta) - d^{\text{out}}_S(v_{s_i})$.

Using the definition of c_S we see that $-c_S(s_2)/(2+\beta) = -d^{\text{in}}_S(v_{s_1})/(2+\beta)$, $c_S(s_k)/(2+\beta) - d^{\text{out}}_S(v_{s_k}) = -(1+\beta)d^{\text{out}}_S(v_{s_k})/(2+\beta)$ and for $2 \le i \le k-1$,

$c_S(s_i) - c_S(s_{i+1}) = d_S^{\text{out}}(v_{s_i}) - d_S^{\text{in}}(v_{s_i})$. Therefore

$$-d^{\text{out}}(v_{s_i}) + \frac{c_S(s_i)}{2+\beta} - \frac{c_S(s_{i+1})}{2+\beta} = \frac{-(1+\beta)d_S^{\text{out}}(v_{s_i}) - d_S^{\text{in}}(v_{s_i})}{2+\beta}.$$

Thus

$$\begin{aligned}
&\prod_{\substack{1<i\le t:\\ v_i\in V^+}} \frac{1}{(2+\beta)i} \prod_{j=2}^{k} (s_j/s_{j-1})^{c_s(s_j)/(2+\beta)} \\
&\qquad = \frac{1}{(2+\beta)^{|V^+|}} \prod_{j=1}^{k} s_j^{-((1+\beta)d_S^{\text{out}}(v_{s_j}) + d_S^{\text{in}}(v_{s_j}))/(2+\beta)} \\
&\qquad = \prod_{i>j:(v_i,v_j)\in E(S)} \frac{1}{(2+\beta)(i^{1+\beta}j)^{1/(2+\beta)}}.
\end{aligned}$$

Substituting this into (2.10) gives the result. □

2.4 Calculation of Expectations

Recall that the clustering coefficient $C(G)$ of a graph G is given by

$$C(G) = \frac{3\times \text{ number of triangles in } G}{\sum_{v\in V(G)} \binom{d(v)}{2}}.$$

In this section we calculate the expectations of the numerator and denominator of this expression.

2.4.1 Expected Number of Triangles

We adapt the methods used in [10] to the case $\beta > 0$. For fixed $a < b < c$, we first calculate the expected number of triangles in $G_{m,\beta}^n$ on vertices v_a, v_b, v_c. Let $G_{1,\beta}^{mn}$ be the underlying tree used to form $G_{m,\beta}^n$. Label the vertices of the tree $v_1', \ldots, v_{mn}'$. A triangle on v_a, v_b, v_c arises if there are vertices v_{a_1}', v_{a_2}' with $(a-1)m+1 \le a_1, a_2 \le am$, v_{b_1}', v_{b_2}' with $(b-1)m+1 \le b_1, b_2 \le bm$ and v_{c_1}', v_{c_2}' with $(c-1)m+1 \le c_1, c_2 \le cm$ such that v_{b_1}' sends its out-going edge to v_{a_1}', v_{c_1}' sends its out-going edge to v_{a_2}' and v_{c_2}' sends its out-going edge

to v'_{b_2}. For this to be possible, we need $c_1 \neq c_2$. Let S be the graph with vertices $v'_{a_1}, v'_{a_2}, v'_{b_1}, v'_{b_2}, v'_{c_1}, v'_{c_2}$ and edges (v'_{b_1}, v'_{a_1}), (v'_{c_1}, v'_{a_2}) and (v'_{c_2}, v'_{b_2}). Write $a_1 = am - l_1$, $a_2 = am - l_2$, $b_1 = bm - l_3$, $b_2 = bm - l_4$, $c_1 = cm - l_5$ and $c_2 = cm - l_6$. The cases where $a_1 = a_2$ and $a_1 \neq a_2$ are slightly different. We concentrate on the former to begin with.

We have $d_S^{\text{in}}(v_{a_1}) = 2$, $d_S^{\text{in}}(v_{b_2}) = 1$ and otherwise $d_S^{\text{in}}(v) = 0$. Suppose that $a_1 > 1$. Then applying Lemma 2.8 we see that

$$\begin{aligned}&\Pr(S \subseteq G_{1,\beta}^{mn}) \\ &\quad = \frac{\Gamma(3+\beta)\Gamma(2+\beta)}{(\Gamma(1+\beta))^2} \frac{1}{(2+\beta)^3} \left(\frac{1}{a_1 a_2 b_2 (b_1 c_1 c_2)^{1+\beta}} \right)^{1/(2+\beta)} \exp(O(1/a)).\end{aligned} \tag{2.11}$$

The same expression holds when $a_1 = 1$ because the extra multiplicative term of $\beta/(2+\beta)$ may be absorbed into the error term. Note that for $-1 \leq x \leq 1$, we have $e^x = 1 + O(x)$. Furthermore $1/a_i = 1/(am)(1+O(1/a))$, $1/b_i = 1/(bm)(1+O(1/a))$ and $1/c_i = 1/(cm)(1+O(1/a))$. So we may rewrite (2.11) as follows:

$$\Pr(S \subseteq G_{1,\beta}^{mn}) = \frac{(1+\beta)^2}{(2+\beta)^2} \frac{1}{m^3} \left(\frac{1}{a^2 b^{2+\beta} c^{2+2\beta}} \right)^{1/(2+\beta)} (1 + O(1/a)).$$

In this case where $a_1 = a_2$, there are $m^4(m-1)$ ways to choose $a_1, a_2, b_1, b_2, c_1, c_2$ so that there is a corresponding triangle on v_a, v_b, v_c in $G_{m,\beta}^n$.

Now we suppose that $a_1 \neq a_2$. We have $d_S^{\text{in}}(v_{a_1}) = d_S^{\text{in}}(v_{a_2}) = d_S^{\text{in}}(v_{b_2}) = 1$ and otherwise $d_S^{\text{in}}(v) = 0$. Applying Lemma 2.8 and carrying out similar calculations to those above we obtain

$$\Pr(S \subseteq G_{1,\beta}^{mn}) = \frac{(1+\beta)^3}{(2+\beta)^3} \frac{1}{m^3} \left(\frac{1}{a^2 b^{2+\beta} c^{2+2\beta}} \right)^{1/(2+\beta)} (1 + O(1/a)).$$

In this case there are $m^4(m-1)^2$ ways to choose $a_1, a_2, b_1, b_2, c_1, c_2$.

Let $N_{a,b,c}$ denote the number of triangles on v_a, v_b, v_c in $G_{m,\beta}^n$. From the

calculations above, we see that

$$\mathbf{E}\left[N_{a,b,c}\right] = \left(m(m-1)\frac{(1+\beta)^2}{(2+\beta)^2} + m(m-1)^2\frac{(1+\beta)^3}{(2+\beta)^3}\right)\left(\frac{1}{a^2b^{2+\beta}c^{2+2\beta}}\right)^{1/(2+\beta)} \cdot (1+O(1/a)). \tag{2.12}$$

Now let N be the number of triangles in $G^n_{m,\beta}$. Then

$$\mathbf{E}\left[N\right] = \sum_{c=3}^{n}\sum_{b=2}^{c-1}\sum_{a=1}^{b-1}\mathbf{E}\left[N_{a,b,c}\right].$$

We estimate this sum using the following standard result on integrals. If $s \le t$, $f : \mathbb{R} \to \mathbb{R}^+$ is nonincreasing and integrable on $[s-1, t+1]$ then

$$\int_s^{t+1} f(x)\,\mathrm{d}x \le \sum_{k=s}^{t} f(k) \le \int_{s-1}^{t+1} f(x)\,\mathrm{d}x$$

and hence if $s \ge 1$,

$$\sum_{k=s}^{t} f(k) = \int_{s-1}^{t+1} f(x)\,\mathrm{d}x + O(1).$$

Note that the order term only depends on $\int_{s-1}^{s} f(x)\,\mathrm{d}x$, which is bounded, and not on t. By applying this result, considering the maximum and minimum values of the error term separately, we obtain

$$\begin{aligned}\sum_{a=1}^{b-1} a^{-2/(2+\beta)}(1+O(1/a)) &= \sum_{a=2}^{b-1} a^{-2/(2+\beta)}(1+O(1/a)) + O(1)\\ &= \int_1^b a^{-2/(2+\beta)}(1+O(1/a))\,\mathrm{d}a + O(1)\\ &= \frac{2+\beta}{\beta}b^{\beta/(2+\beta)} + O(1).\end{aligned}$$

Then similarly we have

$$\sum_{b=2}^{c-1} b^{-1} \sum_{a=1}^{b-1} a^{-2/(2+\beta)}(1+O(1/a)) = \frac{(2+\beta)^2}{\beta^2} c^{\beta/(2+\beta)} + O(\log c).$$

Finally

$$\sum_{c=3}^{n} c^{-(2+2\beta)/(2+\beta)} \sum_{b=2}^{c-1} b^{-1} \sum_{a=1}^{b-1} a^{-2/(2+\beta)}(1+O(1/a)) = \frac{(2+\beta)^2}{\beta^2} \log n + O(1).$$

By substituting this expression into (2.12) we obtain the following result.

Proposition 2.13. *For $\beta > 0$, the expected number of triangles in $G^n_{m,\beta}$ is*

$$\left(m(m-1)\frac{(1+\beta)^2}{\beta^2} + m(m-1)^2 \frac{(1+\beta)^3}{\beta^2(2+\beta)} \right) \log n + O(1).$$

This result is very different from that obtained in [10] where it is shown that when $\beta = 0$ the expected number of triangles is $\Theta((\log n)^3)$.

2.4.2 Expectation of $\sum_{v \in V(G)} \binom{d(v)}{2}$

We begin by noting that if we regard each edge in the graph as consisting of two half-edges, with each half-edge retaining one endpoint of an edge then $\sum_{v \in V(G^n_{m,\beta})} \binom{d_n(v)}{2}$ is the number of pairs of half-edges with the same endpoint. We say such a pair of half-edges is *adjacent*. Suppose that e_1 and e_2 are half-edges with endpoint v. If e_1 and e_2 form respectively half of edges vu and vw with u, v, w pairwise distinct then we say that e_1 and e_2 form a *non-degenerate* pair of adjacent half-edges. Otherwise we say that they are *degenerate*.

Calculating the expected number of pairs of adjacent half-edges is slightly more complicated than calculating the expected number of triangles because there is less symmetry. We begin by counting the number of non-degenerate pairs of adjacent half-edges. Let $a < b < c$. We first calculate the expected number of pairs (v_b, v_a), (v_c, v_a) of adjacent half-edges in $G^n_{m,\beta}$ for $\beta > 0$. Just as in the previous section, there are two cases to consider, and similar

calculations, using Lemma 2.8, to those above show that the number of such pairs of adjacent half-edges is

$$\left(m\frac{1+\beta}{2+\beta}+m(m-1)\frac{(1+\beta)^2}{(2+\beta)^2}\right)\left(\frac{1}{a^2b^{1+\beta}c^{1+\beta}}\right)^{1/(2+\beta)}(1+O(1/a)).$$

By integrating, we see that the total number of pairs of adjacent half-edges in $G^n_{m,\beta}$ for which the common vertex has the smallest index is

$$\left(m\frac{2+\beta}{\beta}+m(m-1)\frac{1+\beta}{\beta}\right)n+O(n^{2/(2+\beta)}).$$

Now the expected number of pairs (v_b,v_a), (v_c,v_b) of adjacent half-edges is

$$m^2\frac{(1+\beta)^2}{(2+\beta)^2}\left(\frac{1}{ab^{2+\beta}c^{1+\beta}}\right)^{1/(2+\beta)}(1+O(1/a)).$$

Again we integrate to derive that the total number of pairs of adjacent half-edges in $G^n_{m,\beta}$ for which the common vertex has the middle index is $m^2n+O(n^{2/(2+\beta)})$. This is not surprising because it can be shown that very few vertices either have loops or do not have m distinct out-neighbours. Each loopless vertex with m distinct loopless out-neighbours, that each have m distinct out-neighbours, is the vertex with greatest index in m^2 pairs of adjacent half-edges of this form.

Finally the expected number of pairs (v_c,v_a), (v_c,v_b) of adjacent half-edges is

$$m(m-1)\frac{(1+\beta)^2}{(2+\beta)^2}\left(\frac{1}{abc^{2+2\beta}}\right)^{1/(2+\beta)}(1+O(1/a)).$$

So the total number of pairs of adjacent half-edges in $G^n_{m,\beta}$ for which the common vertex has the largest index is $\frac{m(m-1)}{2}n+O(n^{1/(2+\beta)})$. Again this is not surprising because each loopless vertex with m distinct out-neighbours is the vertex of greatest index in $\binom{m}{2}$ pairs of adjacent half-edges of this form.

We now determine the number of degenerate pairs of adjacent half-edges, that is the number of pairs of adjacent half-edges where the half-edges belong to parallel edges in $G^n_{m,\beta}$ or at least one half-edge belongs to a loop. First we calculate the number of pairs of adjacent half-edges which belong to parallel

edges in $G^n_{m,\beta}$.

Consider two vertices v_a and v_b with $a < b$ in $G^n_{m,\beta}$. Again we apply Lemma 2.8 to obtain the expected number of edges between v_a and v_b and carry out the same simplifications as above. There are $m\binom{m}{2}$ possibilities to form two parallel edges between v_a and v_b if the out-going edges in the underlying tree have the same parent vertex and $m(m-1)\binom{m}{2}$ if the out-going edges in the underlying tree have different parent vertices. Then the expected number of pairs of adjacent half-edges formed by parallel edges between v_a and v_b is given by the following term

$$2 \cdot \left(\frac{m^2(m-1)}{2}(2+\beta)(1+\beta) + \frac{m^2(m-1)^2}{2}(1+\beta)^2 \right) \cdot \frac{1}{m^2(2+\beta)^2} \frac{1}{(b^{2+2\beta}a^2)^{1/(2+\beta)}}(1+O(1/a)).$$

(The factor two is due to the fact that each pair of parallel edges between v_a and v_b in $G^n_{m,\beta}$ contributes a pair of parallel half-edges at both v_a and v_b.) The expected number of pairs of adjacent half-edges formed by parallel edges in $G^n_{m,\beta}$ is the sum over all $1 \leq a < b \leq n$ of this term. Again the sum can be estimated by an integral and we obtain in this manner that the expected number of pairs of adjacent half-edges formed by parallel edges in $G^n_{m,\beta}$ is $O(\log(n))$.

Now we consider how many pairs of adjacent half-edges are formed by loops. Since there are $\frac{m(m-1)}{2}$ possible edges in the underlying tree that correspond to a loop at v_a in $G^n_{m,\beta}$, by Lemma 2.8 the expected number of loops at vertex v_a is

$$(1+\beta)\frac{m(m-1)}{2}\frac{1}{m(2+\beta)a}(1+O(1/a)).$$

It follows by summing over all $1 \leq a \leq n$ that the expected number of pairs of adjacent half-edges where both half-edges belong to the same loop is $O(\log(n))$. There are at most $m-1$ loops at each vertex that has a loop, therefore the total number of adjacent half-edges where both half-edges belong to a loop is $O(\log(n))$.

In the last case we calculate the expected number of pairs of half-edges

for which one half-edge belongs to a loop and the other one belongs to an edge between two different vertices v_a, v_b in $G^n_{m,\beta}$.

First we calculate the expected number of pairs of adjacent half-edges where one half-edge belongs to a loop on v_a and the other one to edge (v_b, v_a) where $a < b$. By Lemma 2.8, the expected number of pairs of such adjacent half-edges is

$$\left(\frac{m^2(m-1)}{2}(2+\beta)(1+\beta) + \frac{m^2(m-1)^2}{2}(1+\beta)^2\right) \cdot \frac{1}{m^2(2+\beta)^2(a^{3+\beta}b^{1+\beta})^{1/(2+\beta)}}(1+O(1/a)).$$

By summing over all $1 \leq a < b \leq n$ we obtain that the expected number of pairs of such adjacent half-edges is $O(n^{\frac{1}{2+\beta}})$.

Now we calculate the expected number of pairs of adjacent half-edges where one half-edge belongs to a loop on v_b and the other one to edge (v_b, v_a) for $a < b$. We have shown above that the number of loops in $G^n_{m,\beta}$ is $O(\log(n))$. Since any vertex v_b in $G^n_{m,\beta}$ is connected to at most m vertices v_a with $a < b$ there are $O(\log(n))$ expected pairs of adjacent half-edges of this kind.

It follows that the number of degenerate pairs of adjacent half-edges is $O(n^{1/(2+\beta)})$.

Summing over all degenerate and non-degenerate pairs of adjacent half-edges we obtain the following result.

Proposition 2.14. *For $\beta > 0$, the expectation of $\sum_{v \in V(G)} \binom{d(v)}{2}$ in $G^n_{m,\beta}$ is*

$$\left(\frac{2+5\beta}{2\beta}m^2 + \frac{2-\beta}{2\beta}m\right)n + O(n^{2/(2+\beta)}).$$

Again the result is different from that obtained in [10] where it was shown that for the case $\beta = 0$ the expected number of pairs of adjacent edges is $\Theta(n \log n)$.

2.5 Concentration of $\sum_{v \in V(G)} \binom{d(v)}{2}$

In this section we show that the number of pairs of adjacent half-edges in $G^n_{m,\beta}$ is concentrated about its mean. This justifies obtaining the expected clustering coefficient by taking three times the quotient of the expected number of triangles and the expected number of pairs of adjacent half-edges. The main strategy is to apply a variant of the Azuma-Hoeffding inequality from [60], by making use of Móri's results [62] on the evolution of the maximum degree of $G^n_{m,\beta}$. (It is mentioned in [10] that martingale methods were used.) A key notion in the proof is to consider the mechanism by which edges incident with a fixed vertex are added.

Before we continue, we explain briefly why we follow this approach rather than the more elementary second moment method. It is possible to apply the second moment method to obtain some form of concentration. Certainly Lemma 2.4 may be applied to show the leading order terms cancel in the usual way. However the concentration result that may be obtained is not tight enough to obtain our final result without a considerable sharpening of the analysis in Section 2.6. It is far from clear whether this is possible. Furthermore the number of cases that need to be considered makes calculating the variance a gruesome proposition and therefore unlikely to be much shorter to describe than our approach.

Fix β and m. Let (H_t) be the graph process defined as follows. Run $(G^t_{1,\beta})$ and take H_n to be the graph formed from $G^n_{1,\beta}$ by merging groups of m consecutive vertices together until there are at most m left and finally merging the remaining unmerged vertices together. Note that H_n has $\lceil n/m \rceil$ vertices, which we denote by $v_1, \ldots, v_{\lceil n/m \rceil}$ in the obvious way, and $n-1$ edges. Furthermore, if $m|n$ and the graphs H_n and $G^{n/m}_{m,\beta}$ are formed from the same instance of the process $(G^t_{1,\beta})$, then H_n and $G^{n/m}_{m,\beta}$ are the same graph.

Let v_k be a vertex of H_s such that $km \leq s$. For $t \geq s$, we define a partition $\Pi_{k,s}(t)$ of the half-edges incident with v_k. The partition always has $d_s(v_k) + 1$ blocks. When $t = s$, each block of the partition except for one contains one of the $d_s(v_k)$ half-edges incident with v_k; with a slight abuse of

nomenclature the other block, which we call the *base* block, is initially empty. It follows that if v_k has a loop at time s then the two half-edges forming the loop are in separate blocks of $\Pi_{k,s}(s)$. As t increases and more edges are added to H, any newly added half-edge incident with v_k is added to the partition. If at time $t > s$ the parent vertex of the newly added edge is not v_k then $\Pi_{k,s}(t) = \Pi_{k,s}(t-1)$. Suppose that at time $t > s$ the parent vertex of the newly added edge f is v_k: if v_k is chosen preferentially by copying the half-edge $e \in A$, where A is a block of $\Pi_{k,s}(t-1)$, then we form $\Pi_{k,s}(t)$ from $\Pi_{k,s}(t-1)$ by adding the half-edge of f incident with v_k to A; if v_k is chosen uniformly then the half-edge of f incident with v_k is added to the base block.

Suppose that v_l is a vertex of H_s distinct from v_k such that $lm \leq s$. Suppose further that we choose two distinct blocks from $\Pi_{k,s}(t)$ and $\Pi_{l,s}(t)$, such that neither is a base block. The joint distribution of the sizes of the two blocks is the same for any choice of blocks, whether they are both chosen from $\Pi_{k,s}(t)$, $\Pi_{l,s}(t)$ or one from each. Furthermore if we choose either base block from $\Pi_{k,s}(t)$ or $\Pi_{l,s}(t)$ and one other block that is not a base block, then again the joint distribution of the sizes of the blocks does not depend on our choice.

Lemma 2.15. *Let v_j and v_k be distinct vertices of H_s such that $\max\{jm, km\} \leq s$. Let A (B) be respectively a block of $\Pi_{j,s}(t)$ ($\Pi_{k,s}(t)$) such that neither is a base block. Then*

$$\mathbf{E}\left[|A||B|\right] \leq \mathbf{E}\left[|A|\right]\mathbf{E}\left[|B|\right] \leq (t/s)^{2/(2+\beta)}(1+O(1/s)).$$

Proof. Let e_1, e_2 be half-edges so that at time s, e_1 is incident with v_k and e_2 is incident with v_l. Then let a_t denote the size, at time t, of the block of $\Pi_{k,s}(t)$ containing e_1 and let b_t be defined similarly with respect to $\Pi_{l,s}(t)$ and e_2. We first establish the second inequality. We have $\mathbf{E}\left[a_s\right] = 1$ and for $t \geq s$,

$$\mathbf{E}\left[a_{t+1}|a_t\right] = a_t\left(1 + \frac{1}{(2+\beta)t-2}\right). \tag{2.16}$$

Hence

$$\mathbf{E}\left[a_{t+1}\right] = \frac{t - 1/(2+\beta)}{t - 2/(2+\beta)}\mathbf{E}\left[a_t\right].$$

Solving this recurrence, we obtain

$$\mathbf{E}\left[a_t\right] = \frac{\Gamma\left(t - \frac{1}{2+\beta}\right)\Gamma\left(s - \frac{2}{2+\beta}\right)}{\Gamma\left(t - \frac{2}{2+\beta}\right)\Gamma\left(s - \frac{1}{2+\beta}\right)}.$$

A standard result on the ratio of gamma functions [57] states that if a, b are fixed members of $\mathbb{R}$ then for all $x > \max\{|a|, |b|\}$,

$$\frac{\Gamma(x+b)}{\Gamma(x+a)} = x^{b-a}(1 + O(1/x)).$$

Using this result, we obtain

$$\mathbf{E}\left[a_t\right] \leq (t/s)^{1/(2+\beta)}(1 + O(1/s)).$$

Since $|A|$ and $|B|$ are identically distributed, the second inequality in the lemma follows. We prove the first inequality by using induction on t. Observe that (a_{t+1}, b_{t+1}) can take the values $(a_t + 1, b_t)$, $(a_t, b_t + 1)$ and (a_t, b_t) with probabilities respectively $a_t/((2+\beta)t - 2)$, $b_t/((2+\beta)t - 2)$ and $1 - (a_t + b_t)/((2+\beta)t - 2)$. Therefore

$$\mathbf{E}\left[a_{t+1}b_{t+1}|a_t b_t\right] = a_t b_t + \frac{2a_t b_t}{(2+\beta)t - 2}$$

and from (2.16) we get

$$\mathbf{E}\left[a_{t+1}\right]\mathbf{E}\left[b_{t+1}\right] = \mathbf{E}\left[a_t\right]\mathbf{E}\left[b_t\right]\left(1 + \frac{1}{(2+\beta)t - 2}\right)^2.$$

So

$$\mathbf{E}\left[a_{t+1}b_{t+1}\right] - \mathbf{E}\left[a_{t+1}\right]\mathbf{E}\left[b_{t+1}\right] \leq \left(1 + \frac{2}{(2+\beta)t - 2}\right)\left(\mathbf{E}\left[a_t b_t\right] - \mathbf{E}\left[a_t\right]\mathbf{E}\left[b_t\right]\right)$$

and hence the result follows by induction. □

When the maximum degree of H_t becomes unusually large and the parent vertex is chosen to be a vertex of maximum degree, the number of pairs of adjacent half-edges increases by an unusually large amount. The next result enables us to show that the probability of this happening is extremely small. Let $\Delta(G)$ denote the maximum degree of G. The following is a very slight reformulation of what Móri proves in [62, Theorem 3.1].

Theorem 2.17. *For any positive integer k, there exists $\tilde{M}_k \in \mathbb{R}$, such that for all n,*

$$\mathbf{E}\left[\left(\frac{\Delta(G^n_{1,\beta})+\beta}{n^{1/(2+\beta)}}\right)^k\right] \leq \tilde{M}_k.$$

The following corollary is straightforward.

Corollary 2.18. *For any positive integers k, m, there exists $M_{k,m} \in \mathbb{R}$ such that for all positive integers $i_1, \ldots, i_k$,*

$$\mathbf{E}\left[\frac{\Delta(H_{mi_1})}{(mi_1)^{1/(2+\beta)}} \cdots \frac{\Delta(H_{mi_k})}{(mi_k)^{1/(2+\beta)}}\right] \leq M_{k,m}.$$

Proof. Since $\Delta(H_{mi_1}), \ldots, \Delta(H_{mi_k})$ are all positive we have

$$\frac{\Delta(H_{mi_1})}{(mi_1)^{1/(2+\beta)}} \cdots \frac{\Delta(H_{mi_k})}{(mi_k)^{1/(2+\beta)}} \leq \sum_{j=1}^{k} \left(\frac{\Delta(H_{mi_j})}{(mi_j)^{1/(2+\beta)}}\right)^k$$

and so

$$\mathbf{E}\left[\frac{\Delta(H_{mi_1})}{(mi_1)^{1/(2+\beta)}} \cdots \frac{\Delta(H_{mi_k})}{(mi_k)^{1/(2+\beta)}}\right] \leq \sum_{j=1}^{k} \mathbf{E}\left[\left(\frac{\Delta(H_{mi_j})}{(mi_j)^{1/(2+\beta)}}\right)^k\right].$$

Recall that H_{mi} is formed by merging together blocks of m consecutive vertices in an instance of $G^{mi}_{1,\beta}$. So we have $\mathbf{E}\left[(\Delta(H_{mi}))^k\right] \leq \mathbf{E}\left[(m\Delta(G^{mi}_{1,\beta}))^k\right]$. Hence

$$\sum_{j=1}^{k} \mathbf{E}\left[\left(\frac{\Delta(H_{mi_j})}{(mi_j)^{1/(2+\beta)}}\right)^k\right] \leq m^k \sum_{j=1}^{k} \mathbf{E}\left[\left(\frac{\Delta(G^{mi_j}_{1,\beta})}{(mi_j)^{1/(2+\beta)}}\right)^k\right] \leq km^k\tilde{M}_k.$$

The result follows by taking $M_{k,m} = km^k\tilde{M}_k$. □

Before we can state the large deviation result that we use, we need some more definitions. Recall that f_i is a random variable which determines the index of the parent vertex of v_i and that the values taken by $f_2, f_3, \ldots, f_t$ together determine H_t. Furthermore the set of values that f_i can take is denoted by Ω_i and $f_2, \ldots, f_t$ are independent. Let $\Omega = \prod_{i=2}^{t} \Omega_i$.

Let $\mathbf{X} = (f_2, \ldots, f_t)$. We let $H_t(\mathbf{X})$ be the instance of H_t determined by the random variables $f_2, \ldots, f_t$. We will also use this notation both for other random variables associated with H_t and when some or all of the f_i's are set to a particular value. The meaning should be clear from the context but we will generally use ω_i for a member of Ω_i and f_i for a random variable taking values in Ω_i.

Let $D(\mathbf{X}) = \sum_{v \in V(H_t(\mathbf{X}))} \binom{d_t(v)}{2}$ and let $F(\mathbf{X}) = D(\mathbf{X})t^{-2/(2+\beta)}$. Now let $g : \prod_{i=2}^{s} \Omega_i \to \mathbb{R}$ such that

$$(\omega_2, \ldots, \omega_s) \mapsto \mathbf{E}\left[F(\omega_2, \ldots, \omega_s, f_{s+1}, \ldots, f_t)\right]$$

and let $\mathrm{ran} : \prod_{i=2}^{s-1} \Omega_i \to \mathbb{R}$ such that

$$(\omega_2, \ldots, \omega_{s-1}) \mapsto \sup\left\{|g(\omega_2, \ldots, \omega_{s-1}, x) - g(\omega_2, \ldots, \omega_{s-1}, y)| : x, y \in \Omega_s\right\}.$$

So $\mathrm{ran}(\omega_2, \ldots, \omega_{s-1})$ measures the maximum amount that the expected value of $F(\mathbf{X})$ changes when the value of f_s is changed.

For $\boldsymbol{\omega} \in \Omega$, let

$$R^2(\boldsymbol{\omega}) = \sum_{k=2}^{t} \mathrm{ran}(\omega_2, \ldots, \omega_{k-1})^2.$$

Our aim is to bound $R^2(\boldsymbol{\omega})$ as $\boldsymbol{\omega}$ runs over all members of Ω with the possible exception of those belonging to some 'bad' subset $\mathcal{B}$ which we hope to have small probability. We specify $\mathcal{B}$ below but for the moment let $\mathcal{B}$ be any subset of Ω. Let

$$r^2 = \sup\{R^2(\boldsymbol{\omega}) : \boldsymbol{\omega} \in \Omega \setminus \mathcal{B}\}.$$

Then Theorem 3.7 in [60] yields the following inequality. For all $x > 0$,

$$\Pr(|F(\mathbf{X}) - \mathbf{E}\,[F(\mathbf{X})]\,| \geq x) \leq 2(e^{-2x^2/r^2} + \Pr(\mathbf{X} \in \mathcal{B})).$$

Fix $\delta > 0$. We let

$$\mathcal{B}_\delta = \left\{ \mathbf{X} \in \Omega : \sum_{i=1}^{n} \left(\frac{\Delta(H_{mi}(\mathbf{X}))}{(mi)^{2/(2+\beta)}} \right)^2 \geq n^{\frac{\beta}{2+\beta}+\delta} \right\}.$$

Then we have the following.

Lemma 2.19. *For any $\delta > 0$ and $\gamma > 0$, there exists L such that $\Pr(\mathcal{B}_\delta) \leq L\frac{1}{n^\gamma}$, where L is a constant depending on δ, γ, β, m but not on n.*

Proof. For any positive integer k, Markov's inequality gives

$$\Pr(\mathcal{B}_\delta) \leq \frac{\mathbf{E}\left[\left(\sum_{i=1}^{n} \left(\frac{\Delta(H_{mi}(\mathbf{X}))}{(mi)^{2/(2+\beta)}} \right)^2 \right)^k \right]}{n^{\frac{\beta k}{2+\beta}+k\delta}}.$$

The numerator of this fraction is equal to

$$\mathbf{E}\left[\sum_{i_1=1}^{n} \cdots \sum_{i_k=1}^{n} \left(\frac{\Delta(H_{mi_1}(\mathbf{X}))}{(mi_1)^{1/(2+\beta)}} \right)^2 \cdots \left(\frac{\Delta(H_{mi_k}(\mathbf{X}))}{(mi_k)^{1/(2+\beta)}} \right)^2 \frac{1}{(m^k i_1 \cdots i_k)^{2/(2+\beta)}} \right].$$

Using Corollary 2.18 this is at most

$$\begin{aligned} M_{2k,m} \sum_{i_1=1}^{n} \cdots \sum_{i_k=1}^{n} \left(\frac{1}{(m^k i_1 \cdots i_k)^{2/(2+\beta)}} \right) &= M_{2k,m} \left(\sum_{i=1}^{n} \frac{1}{(mi)^{\frac{2}{2+\beta}}} \right)^k \\ &\leq M_{2k,m} \left(\frac{2+\beta}{\beta} \frac{n^{\frac{\beta}{2+\beta}}}{m^{\frac{2}{2+\beta}}} \right)^k. \end{aligned}$$

Hence

$$\Pr(\mathcal{B}_\delta) \leq \frac{M_{2k,m} \left(\frac{2+\beta}{\beta} \frac{1}{m^{\frac{2}{2+\beta}}} \right)^k}{n^{k\delta}}$$

and so letting $k = \lceil \gamma/\delta \rceil$ gives the result. □

We can now state the main result of this section concerning the concentration of the number of pairs of adjacent half-edges around its expectation.

Theorem 2.20. *Let $\beta > 0$. For any $\epsilon > 0$, the number D of pairs of adjacent half-edges in $G^n_{m,\beta}$ is concentrated within $O(n^{(4+\beta)/(4+2\beta)+\epsilon})$ about its expected value. More precisely, for any $\epsilon > 0$ and $\gamma > 0$ there exists n^* such that for all $n \geq n^*$*

$$\Pr\left(|D - \mathbf{E}[D]| \geq n^{\frac{4+\beta}{4+2\beta}+\epsilon}\right) \leq \frac{1}{n^\gamma}.$$

Proof. Let $t = nm$, and fix $s \leq t$. Let $s' = m\lceil s/m \rceil$, so we have $s' \leq t$. Now let

$$\boldsymbol{\omega}_x = (\omega_2, \ldots, \omega_{s-1}, x, \omega_{s+1}, \ldots, \omega_t) \text{ and } \boldsymbol{\omega}_y = (\omega_2, \ldots, \omega_{s-1}, y, \omega_{s+1}, \ldots, \omega_t),$$

where $\omega_i \in \Omega_i$ and $x, y \in \Omega_s$. For $z \in \{x, y\}$, let $d^z_t(v)$ denote the total degree of v at time t in $H_t(\boldsymbol{\omega}_z)$ and let e denote the edge added at time s. Suppose that in $H_t(\boldsymbol{\omega}_x)$ the parent vertex of e is v_{k_1} and in $H_t(\boldsymbol{\omega}_y)$ the parent vertex of e is v_{k_2}. Note that at any time, for every vertex v other than v_{k_1} or v_{k_2}, the degree of v is the same in $H_t(\boldsymbol{\omega}_x)$ and $H_t(\boldsymbol{\omega}_y)$. Therefore $F(\boldsymbol{\omega}_x) - F(\boldsymbol{\omega}_y)$ depends only on the degrees of v_{k_1} and v_{k_2} and is given by

$$\begin{aligned} &F(\boldsymbol{\omega}_x) - F(\boldsymbol{\omega}_y) \\ &\quad = t^{-2/(2+\beta)} \left(\binom{d^x_t(v_{k_1})}{2} + \binom{d^x_t(v_{k_2})}{2} - \binom{d^y_t(v_{k_1})}{2} - \binom{d^y_t(v_{k_2})}{2} \right). \end{aligned} \tag{2.21}$$

From now on we will assume that $k_1 \neq k_2$, because otherwise $F(\boldsymbol{\omega}_x) - F(\boldsymbol{\omega}_y) = 0$. Consider the changes that occur to $H_{s'}$ if we replace $\boldsymbol{\omega}_y$ by $\boldsymbol{\omega}_x$. First the head of e is moved from v_{k_2} to v_{k_1}. Second it is possible that each of the at most $m-1$ edges that are added in the time interval $[s+1, s']$ also have an endpoint moved from v_{k_2} to v_{k_1}: this will happen if the parent vertex of an edge added in the interval $[s+1, s']$ is chosen by preferentially copying the head of an edge which has been moved from v_{k_2} to v_{k_1}, in particular if the parent vertex is chosen by preferentially copying the head of e. Consequently we have

$$d^y_{s'}(v_{k_1}) + 1 \leq d^x_{s'}(v_{k_1}) \leq d^y_{s'}(v_{k_1}) + m$$

and furthermore

$$d^x_{s'}(v_{k_1}) + d^x_{s'}(v_{k_2}) = d^y_{s'}(v_{k_1}) + d^y_{s'}(v_{k_2}).$$

Let $d = d^x_{s'}(v_{k_1}) - d^y_{s'}(v_{k_1})$, $d_1 = d^y_{s'}(v_{k_1})$ and $d_2 = d^x_{s'}(v_{k_2})$. Note that both d_1 and d_2 and consequently also $|d_1 - d_2|$ are at most $\Delta(H_{s-1}(\omega_2, \ldots, \omega_{s-1})) + m$.

Now let $A_0, A_1, \ldots, A_{d_1}$, $(B_0, B_1, \ldots, B_{d_2})$ denote the blocks of the partition $\Pi_{k_1,s'}(t)$ in $H_t(\boldsymbol{\omega}_y)$ ($\Pi_{k_2,s'}(t)$ in $H_t(\boldsymbol{\omega}_x)$) with A_0 (B_0) denoting the base block. The partition $\Pi_{k_1,s'}(t)$ in $H_t(\boldsymbol{\omega}_x)$ contains the blocks $A_0, \ldots, A_{d_1}$ but also d further blocks which we label $C_1, \ldots, C_d$. Then the partition $\Pi_{k_2,s'}(t)$ in $H_t(\boldsymbol{\omega}_y)$ contains the blocks $B_0, \ldots, B_{d_2}$, $C_1, \ldots, C_d$. So using (2.21), we have

$$F(\boldsymbol{\omega}_x) - F(\boldsymbol{\omega}_y) = t^{-2/(2+\beta)} \left(\sum_{i=0}^{d_1} \sum_{j=1}^{d} |A_i||C_j| - \sum_{i=0}^{d_2} \sum_{j=1}^{d} |B_i||C_j| \right). \quad (2.22)$$

Now let

$$\boldsymbol{\omega}_x = (\omega_2, \ldots, \omega_{s-1}, x, \omega_{s+1}, \ldots, \omega_{s'}, f_{s'+1}, \ldots, f_t)$$

and

$$\boldsymbol{\omega}_y = (\omega_2, \ldots, \omega_{s-1}, y, \omega_{s+1}, \ldots, \omega_{s'}, f_{s'+1}, \ldots, f_t).$$

So both $H_t(\boldsymbol{\omega}_x)$ and $H_t(\boldsymbol{\omega}_y)$ evolve deterministically until time s' but randomly thereafter.

Recall that $d \leq m$ and that $|d_1 - d_2|$ is at most $\Delta(H_{s-1}(\omega_2, \ldots, \omega_{s-1})) + m$. Hence from (2.22), Lemma 2.15 and the remarks immediately preceding the lemma, we see that

$$|\mathbf{E}\left[F(\boldsymbol{\omega}_x) - F(\boldsymbol{\omega}_y)\right]| \leq (\Delta(H_{s-1}(\omega_2, \ldots, \omega_{s-1})) + m) m (1/s')^{2/(2+\beta)} (1 + O(1/s')).$$

Notice that this expression does not depend on x or y and holds for all

$\omega_{s+1}, \ldots, \omega_{s'}$. Consequently

$$\mathrm{ran}(\omega_2, \ldots, \omega_{s-1}) \leq (\Delta(H_{s-1}(\omega_2, \ldots, \omega_{s-1})) + m)m(1/s')^{2/(2+\beta)}(1 + O(1/s')).$$

Now let $\boldsymbol{\omega} \in \Omega \setminus \mathcal{B}_\delta$. Then

$$\begin{aligned} R^2(\boldsymbol{\omega}) &= \sum_{s=2}^{nm} (\Delta(H_{s-1}(\omega_2, \ldots, \omega_{s-1})) + m)^2 m^2 (1/s')^{4/(2+\beta)}(1 + O(1/s')) \\ &\leq m^2 \sum_{s=2}^{nm} \left(\frac{2\Delta(H_{s'}(\omega_2, \ldots, \omega_{s'}))}{s'^{2/2+\beta}} \right)^2 (1 + O(1/s')) \\ &\leq 4m^3 \sum_{i=1}^{n} \left(\frac{\Delta(H_{mi}(\omega_2, \ldots, \omega_{mi}))}{(mi)^{2/2+\beta}} \right)^2 (1 + O(1/i')) \\ &\leq cn^{\frac{\beta}{2+\beta}+\delta}, \end{aligned}$$

where c is a constant.

Hence

$$\begin{aligned} \Pr\left(|D(\mathbf{X}) - \mathbf{E}\left[D(\mathbf{X})\right]| \geq n^{\frac{4+\beta}{4+2\beta}+\epsilon} \right) &= \Pr\left(|F(\mathbf{X}) - \mathbf{E}\left[F(\mathbf{X})\right]| \geq n^{\frac{\beta}{4+2\beta}+\epsilon} \right) \\ &\leq 2\exp\left(\frac{-2n^{\frac{\beta}{2+\beta}+2\epsilon}}{cn^{\frac{\beta}{2+\beta}+\delta}} \right) + 2\Pr(\mathcal{B}_\delta). \end{aligned}$$

If we choose $\delta = \epsilon$ then the first term is at most $\frac{1}{2n^\gamma}$ for any $\gamma > 0$ and sufficiently large n. Applying Lemma 2.19 with any $\gamma^* > \gamma$ we see that for sufficiently large n we also have $2\Pr(\mathcal{B}_\delta) \leq \frac{1}{2n^\gamma}$. Hence the result follows. □

2.6 Expected clustering coefficient

In this section we finally state and prove our main result.

Theorem 2.23. *For any $\beta > 0$, the expected clustering coefficient of $G^n_{m,\beta}$ is given by*

$$\mathbf{E}[C(G^n_{m,\beta})] = \frac{3c_1 \log n}{c_2 n} + O(1/n),$$

where

$$c_1 = m(m-1)\frac{(1+\beta)^2}{\beta^2} + m(m-1)^2\frac{(1+\beta)^3}{\beta^2(2+\beta)}$$

and

$$c_2 = \frac{2+5\beta}{2\beta}m^2 + \frac{2-\beta}{2\beta}m.$$

Proof. Recall that $N = N(G^n_{m,\beta})$, $D = D(G^n_{m,\beta})$ denote respectively the number of triangles and pair of adjacent half-edges in $G^n_{m,\beta}$. The expected clustering coefficient is given by $\mathbf{E}\left[C(G^n_{m,\beta})\right] = \mathbf{E}\left[3N/D\right]$.

Choose ϵ so that $0 < \epsilon < \frac{\beta}{4+2\beta}$ and let $\eta = \epsilon + \frac{4+\beta}{4+2\beta} < 1$. Let I denote the interval $[\mathbf{E}[D] - n^\eta, \mathbf{E}[D] + n^\eta]$. From Proposition 2.14 we have $\mathbf{E}[D] - n^\eta = c_2 n - (1 + o(1))n^\eta$ and $\mathbf{E}[D] + n^\eta = c_2 n + (1 + o(1))n^\eta$. Let $n \geq n^*$, the minimum value of n such that Theorem 2.20 may be applied with $\gamma = 4$. Since $C(G^n_{m,\beta}) \leq m$, an upper bound for $\mathbf{E}\left[C(G^n_{m,\beta})\right]$ may be obtained as follows.

$$\begin{aligned}
\mathbf{E}\left[C(G^n_{m,\beta})\right] &\leq \sum_{j=1}^{\infty}\sum_{i\in I}\frac{3j}{i}\Pr(N=j, D=i) + m\Pr(D \notin I) \\
&\leq \sum_{j=1}^{\infty}\frac{3j}{c_2 n - (1+o(1))n^\eta}\Pr(N=j) + m\Pr(D \notin I).
\end{aligned}$$

Applying Theorem 2.20 with $\gamma = 1$ and then Proposition 2.13, we obtain

$$\begin{aligned}
\mathbf{E}\left[C(G^n_{m,\beta})\right] &\leq \sum_{j=1}^{\infty}\frac{3j}{c_2 n - (1+o(1))n^\eta}\Pr(N=j) + \frac{m}{n} \\
&= \frac{3c_1 \log n}{c_2 n}(1 + (1/c_2 + o(1))n^{\eta-1}) + \frac{m}{n} \\
&= \frac{3c_1 \log n}{c_2 n} + O(1/n).
\end{aligned}$$

A lower bound for $\mathbf{E}\left[C(G^n_{m,\beta})\right]$ may be obtained as follows.

$$\begin{aligned}
\mathbf{E}\left[C(G^n_{m,\beta})\right] &\geq \sum_{j=1}^{\infty}\sum_{i\in I}\frac{3j}{i}\Pr(N=j,D=i)\\
&\geq \sum_{j=1}^{\infty}\sum_{i\in I}\frac{3j}{c_2n+(1+o(1))n^{\eta}}\Pr(N=j,D=i)\\
&= \frac{3\mathbf{E}\left[N\right]}{c_2n+(1+o(1))n^{\eta}}\\
&\quad -\sum_{j=1}^{\infty}\sum_{i\notin I}\frac{3j}{c_2n+(1+o(1))n^{\eta}}\Pr(N=j,D=i).
\end{aligned}$$

Now since there are at most n^3m^3 triangles in $G^n_{m,\beta}$

$$\sum_{j=1}^{\infty}\sum_{i\notin I}\frac{3j}{c_2n+(1+o(1))n^{\eta}}\Pr(N=j,D=i)\leq\frac{3n^3m^3}{c_2n+(1+o(1))n^{\eta}}\Pr(D\notin I).$$

Applying Theorem 2.20 with $\gamma=4$ shows that this is $O\left(1/n\right)$. Finally

$$\frac{3\mathbf{E}\left[N\right]}{c_2n+(1+o(1))n^{\eta}}=\frac{3c_1\log n}{c_2n}(1-(1/c_2+o(1))n^{\eta-1})=\frac{3c_1\log n}{c_2n}+O(1/n).$$

□

2.7 Conclusion

Our main result shows that for $\beta>0$ the expectation of the clustering coefficient of the Móri graph is asymptotically proportional to $\log n/n$ and consequently that the Móri graphs do not have the small-worlds property. Bollobás and Riordan showed for an almost identical model that when $\beta=0$, the expectation of the clustering coefficient is asymptotically proportional to $(\log n)^2/n$. An unexpected consequence, for which we do not yet have a good explanation, is that the clustering coefficient has a discontinuity at $\beta=0$. It is an open question whether the method for the calculation of the clustering coefficient can be extended to the Cooper-Frieze model which we discuss in the next chapter.

Chapter 3

Non-Searchability of Random Power-Law Graphs

3.1 Introduction

This chapter is joint work with Philippe Duchon, Nicolas Hanusse, both at the LaBRI of the Université Bordeaux, and Steven Noble, Brunel University. A preliminary version of this chapter appeared as a brief announcement in [25] and is published in [26]. In this chapter, we investigate the efficiency of local search algorithms for some of the networks discussed in the previous chapter. We are interested in seeking a vertex of a given identity starting from any vertex. The time complexity of searching is expressed as the number of vertices explored before reaching the target or a neighbour of the target. For an arbitrary graph and a worst-case choice of vertices, time complexity can be of the same order as the size of the graph itself. However, there is some hope that for large classes of graphs, the size of the exploration sequence would drastically decrease.

A *local distributed algorithm* is one which falls within the following framework. Each vertex knows the identities of all vertices and/or edges up to a fixed distance away. A search request contains the identities of the initial vertex, target vertex and possibly information about vertices previously encountered in the search and their neighbourhoods. When a search request arrives at a vertex v, the choice of the next vertex to explore must be based

on the information known to v. Normally the next vertex chosen by v must be one of its neighbours, although our results do not require this. The process stops as soon as the target is reached. This notion is similar to the decentralised algorithm of [42].

In the seminal paper [43], Kleinberg studies a local distributed search algorithm on a graph built from a 2D-grid augmented by one random extra edge per vertex. Kleinberg gives some condition on the design of the random extra edges to obtain graphs in which the expected number of search requests of this local distributed search algorithm is $O(\log^2 n)$ and proves that many augmentation processes yield graphs for which the expected number of search requests is $\Omega(n^{1/3})$. In both cases the expectation is over both the choice of random edges and the initial and target vertices.

We show that for random graphs built in different scale-free models, the expected number of requests made by any local distributed search algorithm for the worst case choice of target vertex is $\Omega(n^c)$ for explicit constants c. Our results apply to very general searching strategies.

For an arbitrary graph (for instance, a path), the time complexity of searching is clearly linear. Due to the common properties of scale-free graphs (low diameter and existence of large degree vertices), one might hope that the time complexity could be very small, sublinear and possibly polylogarithmic.

Such a claim can be found in different studies dealing with search strategies in scale-free networks (see [1, 41, 75]). In these papers, the authors start from different random graph models whose degree distributions follow power-laws and provide heuristics to reach a target vertex. Simulations and non-rigorous mean-field analysis are used to claim that on average, the target can be reached in $O(n^c)$ steps, for a constant $c < 1$, using simple greedy algorithms. As an input, Adamic *et al.* [1] takes random graphs in the random power-law model whose exponent ℓ is strictly between 2 and 3. They propose two strategies: a pure random walk and a search process based on high degree vertices. The last distributed algorithm works as follows: at each step, the next visited vertex is the highest degree neighbour of the set of visited vertices. Using a mean-field analysis of this greedy algorithm, the authors prove that the target is reached on average in $O(n^{2(1-2/\ell)})$ steps whereas for

a pure random walk, the time complexity becomes $O(n^{3(1-2/\ell)})$ steps. The article also provides simulations on random graphs and sampling confirming the tendency of the theoretical analysis. However, this article lacks a well-defined random graph model, which means that comparison with our work is difficult. Furthermore their searching model provides information about vertex degrees that is not available in our models.

In [41], the authors deal with a search process based on high degree vertices similar to that in [1], using the BA model for the underlying topology. The article contains simulations indicating that after visiting a sublinear, though polynomial, number of vertices, a path of logarithmic length can be found between the source and target vertices. Note that all of these articles deal with the average time complexity on all pairs of sources and targets. To our knowledge, it is still an open problem whether the searching time in these scale-free graphs can be of polylogarithmic order.

In this chapter, we answer this question negatively for two specific families of random scale-free graphs based upon preferential attachment processes, namely the Móri model [61, 62] and the Cooper-Frieze model [20].

In our setup we assume that each vertex knows its own identity and its degree and we study two models of local knowledge: each vertex either knows the labels of its neighbours (*strong model*) or it does not (*weak model*). The first model is the more realistic; the second model is mostly a technical tool to prove lower bounds, which are then extended to the strong model using known results on the maximum degree.

In both families, we prove polynomial lower bounds on the expected number of vertices that any local distributed algorithm visits before reaching the target in the weak model, for a worst-case choice of target vertex. For the Móri graph and the weak model, when the target is chosen uniformly at random, we can also prove a polynomial lower bound for the expected searching time, where the expectation is now taken over choice of target vertex. Furthermore we also establish a polynomial lower bound for the expected searching time for the worst case choice of target in the Móri graph operating under the strong model.

Our results are summarised in Table 3.1. In the random Móri tree for

Scale-free models	Knowledge model	Lower Bounds
Cooper-Frieze	Weak	$\Omega(n^{1/2})$
m-out Móri	Weak	$\Omega(n^{1/2})$ if $\beta > -1$
m-out Móri	Strong	$\Omega(n^{\frac{\beta}{4+2\beta}-\epsilon})$ if $\beta > 0, \epsilon > 0$
1-out Móri	Strong	$\Theta(n)$ if $\beta \geq 0$

Table 3.1: Lower bounds on expected search time.

$m = 1$ we have been able to prove a significantly higher $\Omega(n)$ lower bound on the complexity of finding vertex v_n when starting from v_1 in the strong model.

3.2 Modelling the searching process

Recall that we consider two models of local knowledge, which differ in the information gained when a vertex is visited for the first time.

At each time step, the searching process can try to discover a new vertex by making a request. Finally the process outputs the description of a path from the starting vertex to the target vertex and then stops; our *measure of its performance* is the number of requests it made prior to stopping. We will always assume that the starting vertex and the vertex to be found are different and are defined a priori that means before the searching process starts.

- In the *weak* model, we start by knowing at time 1 the identity of the starting vertex and the identities of its incident edges. At all further time steps a request is in the form of a pair (u, e), where u is an already discovered vertex, and e is an edge incident with u. The answer to the request is the identity v of the other endpoint of edge e, together with the list of all edges incident with v.

- In the *strong* model, we start by knowing at time 1 the identity of the starting vertex and its neighbouring vertices. At each further time step a request is in the form of a vertex u that is adjacent to an already

discovered vertex, and the answer consists of the list of vertices adjacent to u.

Let $\mathcal{A}$ denote some searching strategy in the weak model, that is, $\mathcal{A}$ defines, for each size n, starting vertex u and target vertex v, and each finite sequence of searching requests and possible answers to these requests, the next request that the searching process should make (or, if the searching process is itself randomised, a *probability distribution* for the next request). For each possible realisation G of the random graph, $\mathcal{A}$ thus defines a (finite or infinite) sequence $\mathcal{A}(G)$ of requests and answers. Thus, on the probability space Ω for the random graph, it defines a stochastic searching process $(R_t, A_t)_{t\geq 2}$ (here R_t is the request at time t, and A_t is the answer to it).

Our lower bound proofs rely on the weak model, results on the more realistic strong model come from known upper bounds on the degrees of the graphs studied.

Note that we do not impose restrictions on the sequence of requests. If modelling a search by a mobile agent, one would require at least that each request is about a vertex adjacent to the previous one, even if it meant re-issuing a previous request, so as to model the movement from one known vertex to another. We are also not making any assumptions on the computing power or memory requirements of the searching process. Since we are working on lower bounds, our results are valid under more restrictive assumptions.

3.3 Vertex equivalence

Our proofs of lower bounds are based on a probabilistic notion of vertex equivalence. Intuitively, vertices are equivalent if their identities can be exchanged without modifying the probability distribution on graphs. In this section, we make this notion precise, and show how it can be used to prove lower bounds on the complexity of searching processes.

Definition 3.1. Let G be some graph on the vertex set V, $U \subset V$, and $\sigma \in \mathcal{S}_U$ some permutation of U. We write $\sigma(G)$ for the graph on the same vertex set that is obtained by applying permutation σ to the vertices in U.

Definition 3.2. Let G be some random graph model on the vertex set V, and $U \subset V$, that is, we assume that a probability space Ω is given, where a graph-valued random variable is defined with the appropriate probability distribution.

We say that the vertices in U are (probabilistically) equivalent if, for any $\sigma \in \mathcal{S}_U$, the random graphs G and $\sigma(G)$ have the same probability distribution.

If $\mathcal{E}$ is some event, we say that the vertices in U are equivalent *conditioned on* $\mathcal{E}$ if, for all $\sigma \in \mathcal{S}_U$, G and $\sigma(G)$ have the same probability distribution, conditioned on $\mathcal{E}$.

We do not require $\mathcal{E}$ to be G-measurable, that is, when looking at a possible realisation G of the graph, or even at a possible realisation of the graph process $G_1, \dots, G_t$, it might not be possible to decide whether $\mathcal{E}$ occurred. We only have to find, in a suitable (faithful) model for the construction of our random graph, an event $\mathcal{E}$ (which will be measurable with respect to the construction process) with a large enough probability such that our equivalence property holds conditioned on $\mathcal{E}$.

The following lemma will be useful for the Cooper-Frieze model.

Lemma 3.3. *If $(\mathcal{E}_i)_{i\in I}$ is some (finite or countable) collection of pairwise disjoint events such that the vertices in U are equivalent conditioned on each event $\mathcal{E}_i$, then they are also equivalent conditioned on $\mathcal{E} = \cup_{i\in I}\mathcal{E}_i$.*

Proof. For any $\sigma \in \mathcal{S}_U$ and $i \in I$, we have

$$\begin{aligned}\Pr(G = g, \mathcal{E}_i) &= \Pr(\mathcal{E}_i)\Pr(G = g|\mathcal{E}_i)\\ &= \Pr(\mathcal{E}_i)\Pr(\sigma(G) = g|\mathcal{E}_i) = \Pr(\sigma(G) = g, \mathcal{E}_i).\end{aligned}$$

Summing over all i, and dividing by $\Pr(\mathcal{E})$, we get

$$\Pr(G = g|\mathcal{E}) = \Pr(\sigma(G) = g|\mathcal{E}).$$

□

The following lemma is the key to our lower bounds.

Lemma 3.4. *If a set U of vertices is equivalent conditioned on some event $\mathcal{E}$, then any search process operating in the weak model, starting from any vertex u and searching for any vertex $v \in U$ $(v \neq u)$, has an expected complexity of at least $|U| \Pr(\mathcal{E})/2$.*

Proof. The whole proof is a formalisation of the intuitively obvious fact that, conditioned on $\mathcal{E}$, the vertices in U are visited by the searching process in a uniform random order until the target is found, which implies that, in expectation, half of the vertices in U are visited before the target.

We are concerned with the length of this process, that is, the first index t for which the target vertex v is specified in A_t. If this length is not finite with probability 1, then its expectation is infinite and the lemma holds. Thus, in the rest of this proof, we assume that, with probability 1, the searching process does visit its target at some point.

For any $w \in U \backslash \{v, u\}$ let $\mathcal{E}_w$ be the event that vertex w is found before v during the search and let I_w be the indicator of the event $\mathcal{E}_w$. Conditional on $\mathcal{E}$, at any stage during the search before either w or v has been found the probability that the endpoint of a requested edge is v is the same as the probability that it is w. So $\Pr(\mathcal{E}_w|\mathcal{E}) = 1/2$. Let S be the number of requests needed to find the target vertex v. Then we obtain the following for the expected value of S.

$$\begin{aligned}
\mathbf{E}[S] &\geq 1 + \sum_{w \in U \backslash \{u,v\}} \mathbf{E}[I_w] \\
&= 1 + \sum_{w \in U \backslash \{u,v\}} \Pr(\mathcal{E}_w) \\
&\geq \left(1 + \sum_{w \in U \backslash \{u,v\}} \Pr(\mathcal{E}_w|\mathcal{E})\right) \Pr(\mathcal{E}) \\
&\geq \left(1 + \frac{|U| - 2}{2}\right) \Pr(\mathcal{E}). \qquad (3.5)
\end{aligned}$$

□

3.4 Lower bound on the searching time for the Móri graph

3.4.1 The Móri model for $\beta > -1$

In this section, we deviate slightly from the description of the Móri model [62] as given in Section 2.2. Throughout this section we define the random graph process $(G^n_{m,\beta})$ exactly as in Section 2.2 except that we impose the weaker condition that $\beta > -1$ (as in [62]) and we define the random variables f_i for $2 \leq i \leq n$ as denoting the index of the out-neighbour of v_i and taking probabilities as given by (2.3).

3.4.2 Conditional equivalence in the Móri tree

Definition 3.6. For integers a, b, M with $a < b$ let $\mathcal{E}_{a,b,M}$ be the event that the random graph $G^n_{1,\beta}$ has maximum degree at most M and $f_k \leq a$ for $a+1 \leq k \leq b$. Let $\mathcal{E}_{a,b}$ be the event that $f_k \leq a$ for $a+1 \leq k \leq b$.

Lemma 3.7. *Let $\beta > -1$ and a, b, M be integers with $a < b$. Suppose that $\mathcal{E}_{a,b,M}$ has positive probability. In $G^n_{1,\beta}$, the vertex set $U = \{v_{a+1}, \ldots, v_b\}$ is equivalent conditioned on the event $\mathcal{E}_{a,b,M}$ and also equivalent conditioned on $\mathcal{E}_{a,b}$.*

Proof. Let g_c be a possible realisation of $G^c_{1,\beta}$. For any c, the probability of $\{G^c_{1,\beta} = g_c\}$ can be expressed as a product:

$$\Pr(G^c_{1,\beta} = g_c) = \prod_{3 \leq k \leq c} \Pr(f_k = N_k | G^{k-1}_{1,\beta} = g_{k-1}),$$

where N_k is the index of the out-neighbour of v_k in g_c. So

$$\Pr(G^c_{1,\beta} = g_c) = \prod_{3 \leq k \leq c} \frac{d_{k-1}(v_{N_k}) + \beta}{2(k-2) + (k-1)\beta}. \tag{3.8}$$

We can rewrite the numerator in expression (3.8), grouping together fac-

tors for times where each vertex has its indegree increase:

$$\Pr(G^c_{1,\beta} = g_c) = \frac{\prod_{1\le k\le c}\prod_{1\le j<d_k}(j+\beta)}{(2+\beta)^{c-2}\frac{\Gamma(c-1+\beta/(2+\beta))}{\Gamma(1+\beta/(2+\beta))}}. \tag{3.9}$$

The interesting fact about (3.9) is that the expression *only depends on the degree sequence of* g_c. Thus, if σ is some permutation of the vertices, we have

$$\Pr(G^n_{1,\beta} = g_n) = \Pr(G^n_{1,\beta} = \sigma(g_n)),$$

provided both probabilities are nonzero, that is provided the graphs g_n and $\sigma(g_n)$ only have edges going from vertices to lower-numbered vertices. This depends on σ: the condition is that any vertices that are joined by an edge in g_n, do not have their relative order exchanged by σ.

Similarly we have for $\mathcal{E} = \mathcal{E}_{a,b}$ or $\mathcal{E} = \mathcal{E}_{a,b,M}$

$$\Pr(G^n_{1,\beta} = g_n|\mathcal{E}) \propto \frac{\prod_{1\le k\le c}\prod_{1\le j<d_k}(j+\beta)}{(2+\beta)^{c-2}\frac{\Gamma(c-1+\beta/(2+\beta))}{\Gamma(1+\beta/(2+\beta))}}$$

or

$$\Pr(G^n_{1,\beta} = g_n|\mathcal{E}) = 0$$

and so again

$$\Pr(G^n_{1,\beta} = g_n|\mathcal{E}) = \Pr(G^n_{1,\beta} = \sigma(g_n)|\mathcal{E})$$

providing both probabilities are non-zero, that is provided the graphs g_n and $\sigma(g_n)$ only have edges from higher to lower numbered vertices and if $\mathcal{E} = \mathcal{E}_{a,b,M}$ in addition the maximum degree is at most M. Any permutation σ on $\{v_{a+1}, \ldots, v_b\}$ fulfils this condition.

□

We are now ready to prove our main result on this model. All we need to prove is that, for suitable choices of a and b, $|U|$ and $\Pr(\mathcal{E}_{a,b})$ are both large enough.

Lemma 3.10. *Let* $b = a + \lfloor (a-1)^{1/2} \rfloor$. *Then* $\Pr(\mathcal{E}_{a,b}) \geq e^{-(1+\beta)}$.

Proof. Conditioned on $f_{a+j} \leq a$ for $1 \leq j \leq k$, the probability that f_{a+k+1} is larger than a is exactly $\frac{k(1+\beta)}{2(a+k-1)+(a+k)\beta}$ (there are exactly k vertices that are "bad" candidates, each with degree 1 and therefore a probability of being chosen equal to $(1+\beta)/(2(a+k-1)+(a+k)\beta)$). Taking the product over values of k from 0 to $b-a-1$, we get

$$\begin{aligned}
\Pr(\mathcal{E}_{a,b}) &= \prod_{k=0}^{b-a-1} \left(1 - \frac{k(1+\beta)}{2(a+k-1)+(a+k)\beta}\right) \\
&\geq \exp\left(-\frac{2(1+\beta)}{a-1} \sum_{k=0}^{b-a-1} k\right) \\
&\geq \exp\left(-\frac{(1+\beta)(b-a)^2}{a-1}\right),
\end{aligned}$$

where the second line uses the fact that $1 - x \geq \exp(-2x)$ for $0 \leq x \leq 1/2$.

Plugging $(b-a)^2 \leq a-1$ into the last inequality gives us the required lower bound

$$\Pr(\mathcal{E}_{a,b}) \geq e^{-(1+\beta)}.$$

□

3.4.3 Lower bound on searching time

We are now able to prove our main result on the searching time in the Móri graph.

Theorem 3.11. *Let* $N \geq n$. *For* $\beta > -1$ *and any* $m \geq 1$, *in the Móri graph* $G^N_{m,\beta}$, *no searching algorithm operating in the weak model can find a path to vertex* v_n *with an expected number of requests less than* $\Omega(n^{1/2})$.

For $\beta > 0, m \geq 1, \lambda \geq 1$ *in the Móri graph* $G^{\lceil \lambda n \rceil}_{m,\beta}$, *no searching algorithm operating in the strong model can find a path to vertex* v_n *with an expected number of requests less than* $\Omega(n^{\frac{\beta}{2\beta+4}-\epsilon})$, *for any* $\epsilon > 0$.

Proof. Combining Lemmas 3.7 and 3.10, we see that providing n is at least some very small constant, in $G^t_{1,\beta}$ for $t \geq n$ the vertices in $\{v_{n-\lfloor \sqrt{n}/2 \rfloor + 1}, \ldots, v_n\}$ are equivalent conditioned on an event of probability bounded away from 0.

Suppose that t is a multiple of m. Going from $G^t_{1,\beta}$ to $G^{t/m}_{m,\beta}$, we also get the same equivalence result for vertices in $\{v_{\lfloor (n-\sqrt{n}/2)/m \rfloor+1}, \ldots, v_{\lfloor n/m \rfloor}\}$ conditioned on the same event. (Permuting the vertices in $G^{t/m}_{m,\beta}$ corresponds to only permuting the vertices in $G^t_{1,\beta}$ in such a way as to keep together blocks of m consecutive vertices.) Lemma 3.4 then concludes the proof for the weak model.

We now turn to the proof of the strong model. Given any strong searching strategy $\mathcal{A}$, which starts at vertex v and searches for vertex u, we can define a weak searching strategy $\mathcal{B}$ which visits exactly the same vertices. At time $t = 1$ strategy $\mathcal{B}$ knows vertex v and its neighbouring edges. In the next time steps $\mathcal{B}$ requests all these neighbouring edges in an arbitrary order. So at time $t = 1 + d(v)$ it knows all neighbouring vertices of v. Suppose that with a certain probability at time $t = 2$ strategy $\mathcal{A}$ chooses neighbour x of v. Then with the same probability strategy $\mathcal{B}$ requests the not yet requested neighbouring edges of vertex x each in one time step from time $t = 2 + d(v)$ until all these edges are requested. In the following time steps strategy $\mathcal{B}$ requests the not yet requested neighbouring edges of the vertex which is chosen by strategy $\mathcal{A}$ at time 3 and so on.

So if strategy $\mathcal{A}$ needs in expectation time t^* to explore vertex v in a graph G then strategy $\mathcal{B}$ needs in expectation at most $\Delta(G) \cdot t^*$ time steps to explore the target v, where $\Delta(G)$ is the maximum degree of G.

We now turn to the Móri tree $G^{\lceil \lambda n \rceil}_{1,\beta}$ and show first that $\mathcal{E}_{n-\lfloor \sqrt{n}/2 \rfloor, n, (\lambda n)^{\frac{1}{2+\beta}+\epsilon}}$ has positive probability if the tree is sufficiently large. We have

$$\begin{aligned}\Pr\left(\mathcal{E}_{n-\lfloor \sqrt{n}/2 \rfloor, n, (\lambda n)^{\frac{1}{2+\beta}+\epsilon}}\right) &= \Pr\left(\{\Delta(G^{\lceil \lambda n \rceil}_{1,\beta}) \leq (\lambda n)^{\frac{1}{2+\beta}+\epsilon}\} \cap \mathcal{E}_{n-\lfloor \sqrt{n}/2 \rfloor, n}\right) \\ &\geq \Pr\left(\mathcal{E}_{n-\lfloor \sqrt{n}/2 \rfloor, n}\right) - \Pr\left(\{\Delta(G^{\lceil \lambda n \rceil}_{1,\beta}) > (\lambda n)^{\frac{1}{2+\beta}+\epsilon}\}\right).\end{aligned}$$

From Móri's result on the maximum degree $\Delta(G^t_{1,\beta})$ of the Móri graph $G^t_{1,\beta}$ [62] as stated in Theorem 2.17, we can deduce that for $\epsilon > 0$,

$$\lim_{n\to\infty} \Pr\left(\Delta(G^n_{1,\beta}) \geq n^{\frac{1}{2+\beta}+\epsilon}\right) = 0.$$

So by Lemma 3.10 for any $\delta > 0$ there is an n' so that for any $n > n'$

$$\Pr\left(\mathcal{E}_{n-\lfloor\sqrt{n}/2\rfloor,n,(\lambda n)^{\frac{1}{2+\beta}+\epsilon}}\right) \geq e^{-(1+\beta)} - \delta.$$

So if we let $M = (\lambda n)^{\frac{1}{2+\beta}+\epsilon}$ then

$$\Pr(\mathcal{E}_{n-\lfloor\sqrt{n}/2\rfloor,n,M}) \geq e^{-(1+\beta)} + o(1).$$

Let S_s be the number of requests needed by a strategy $\mathcal{A}$ operating in the strong model to find the target and let S_w denote the number of request given by the corresponding strategy $\mathcal{B}$ (defined above) operating in the weak model to find the target. We now establish the result for the strong model in the case when $m = 1$.

Conditional on $\mathcal{E}_{n-\lfloor\sqrt{n}/2\rfloor,n,M}$, the maximum degree in the Móri tree $G_{1,\beta}^{\lceil\lambda n\rceil}$ is at most M and by Lemma 3.7 the vertices $\{v_{n-\lfloor\sqrt{n}/2\rfloor+1}, \ldots, v_n\}$ are equivalent. Hence

$$\begin{aligned}
\mathbf{E}[S_s] &\geq \mathbf{E}\left[S_s|\mathcal{E}_{n-\lfloor\sqrt{n}/2\rfloor,n,M}\right]\Pr(\mathcal{E}_{n-\lfloor\sqrt{n}/2\rfloor,n,M}) \\
&\geq \frac{\mathbf{E}\left[S_w|\mathcal{E}_{n-\lfloor\sqrt{n}/2\rfloor,n,M}\right]}{M}\Pr(\mathcal{E}_{n-\lfloor\sqrt{n}/2\rfloor,n,M}) \\
&\geq \frac{\lfloor\sqrt{n/2}\rfloor - 1}{2M}(e^{-(1+\beta)} + o(1)) \\
&= \Omega(n^{\frac{\beta}{4+2\beta}-\epsilon}),
\end{aligned}$$

where we use the proof of Lemma 3.4 (a slight variation of inequality (3.5)) in the third inequality.

In $G_{1,\beta}^{\lceil\lambda n\rceil m}$ conditioned on

$$\mathcal{E}_{m(n-\lfloor\sqrt{n}/2\rfloor),mn,(\lambda nm)^{\frac{1}{2+\beta}+\epsilon}},$$

the vertices $v_{m(n-\lfloor\sqrt{n}/2\rfloor)+1}, \ldots, v_{mn}$ are equivalent. So we see that conditioned on the same event, the vertices $v_{n-\lfloor\sqrt{n}/2\rfloor+1}, \ldots, v_n$ are equivalent in $G_{m,\beta}^{\lceil\lambda n\rceil}$, and the result follows for all m.

□

Remark 1. Note that our result, at least for the basic BA model which corresponds to $\beta = 0$ in the Móri model may appear to be a consequence of Cooper *et al.*'s result [21] that the dominating number of the BA graph of size n is $\Omega(n)$, but it is not clear how one could rigorously argue that searching for the worst-case target requires (with positive probability) visiting a positive proportion of a dominating set. The search algorithm is provided with some information in the form of vertex labels indicating their age.

3.4.4 Average searching time in the weak model

Suppose the initial vertex v_i and the target vertex v_j are chosen uniformly at random. Then we obtain the expected number of requests needed to find the target vertex in $G^n_{m,\beta}$ in the weak model as follows. We let J denote the index of the target vertex and let S denote the number of requests needed to find the target vertex. By Theorem 3.11 we have $\mathbf{E}[S|J=j] = \Omega(\sqrt{j})$ for any vertex $j \leq n$ in $G^n_{m,\beta}$. Hence

$$\begin{aligned}\mathbf{E}[S] &= \sum_{1 \leq j \leq n} \mathbf{E}[S|J=j] \Pr(J=j) \\ &= \frac{1}{n}\sum_{j=1}^{n} \Omega(\sqrt{j}) = \Omega(\sqrt{n}).\end{aligned}$$

Thus the expected searching time to find the target vertex if initial and target vertex are chosen uniformly is $\Omega(\sqrt{n})$.

3.5 Lower bound on searching time for the Cooper-Frieze model

3.5.1 The Cooper-Frieze model

The Cooper-Frieze [20] model is a very general model of random evolving graphs, which is defined by several real parameters $\alpha, \beta, \gamma, \delta$ and two discrete probability distributions $\boldsymbol{p}, \boldsymbol{q}$. We first simply give a short informal

description of the evolution rules: at each time step, one randomly chooses whether to apply procedure **New** (with probability $1-\alpha$) or procedure **Old** (with probability α). Procedure **New** will add a new vertex v and a random number (governed by distribution $\boldsymbol{p}$) of edges oriented away from v, while procedure **Old** will add a random number (governed by distribution $\boldsymbol{q}$) of new edges from one randomly selected existing vertex to other existing vertices. Parameters β, γ and δ control probabilities that additional choices of vertices and endpoints are made preferentially or uniformly. We use the same model and notation as in [20].

The graph process G_t starts at time $t = 0$ with a single vertex 0. Thereafter the behaviour is governed by the parameters $\alpha, \beta, \gamma, \delta, \mathbf{p}, \mathbf{q}$ whose meaning is given below:

Choice of procedure at step t.

α**:** Probability that an OLD vertex generates edges.

$1-\alpha$**:** Probability that a NEW vertex is created.

Procedure NEW

$\mathbf{p} = (p_i : i \geq 1)$**:** Probability that the new vertex generates i new edges.

β**:** Probability that terminal vertices are selected uniformly from G_{t-1}.

$1-\beta$**:** Probability that terminal vertices are selected from G_{t-1} with probability proportional to vertex degree.

Procedure OLD

$\mathbf{q} = (q_i : i \geq 1)$**:** Probability that the old vertex generates i new edges.

δ**:** Probability that the initial vertex is selected uniformly from G_{t-1}.

$1-\delta$**:** Probability that the initial vertex is selected from G_{t-1} with probability proportional to vertex degree.

γ**:** Probability that terminal vertices are selected uniformly from G_{t-1}.

$1-\gamma$**:** Probability that terminal vertices are selected from G_{t-1} with probability proportional to vertex degree.

An exception to this is that at time $t = 1$ the endpoints of edges are always chosen uniformly. Cooper and Frieze assumed $\alpha < 1$, and that there are j_0, j_1 such that $p_j = 0$ if $j > j_0$ and $q_j = 0$ if $j > j_1$.

In [20] they showed that the proportion of vertices with a given degree in the evolving graph obeys a power-law.

The proof of the lower bound on the searching time for this model in the weak searching model relies on a conditional equivalence similar to Lemma 3.7. Because the total number of edges when vertex n appears is not deterministic in the Cooper-Frieze model, we do not get a nice product formula like (3.9), which forces us to use a more complex setup. Another complication is that the events which we use to prove conditional equivalence are not measurable with respect to the natural filtration of the graph process; we have to describe a suitable probability space for the construction of the Cooper-Frieze graph, and describe the conditioning events in this probability space. Actually, we will be describing this model for the so-called *restricted* process, which corresponds to conditioning with respect to the evolution of the numbers of vertices and edges.

Still, the general idea of the proof is the same as in the Móri case: it amounts to proving that a set of $\sqrt{n}$ consecutive vertices are equivalent conditional on an event of non-vanishing probability, because none of them has been used in preferential or uniform attachment choices by the time the latest of them appears.

3.5.2 A model for the construction of the Cooper-Frieze graph

In the unrestricted process, the number $\nu(t)$ of vertices in the graph at time t, and the number $\eta(t)/2$ of edges at time t, are both random.

Let $\mathcal{A}$ denote the event that, up to time $t = t_1$, these functions coincide with given functions $\nu_{\mathcal{A}}(t)$ and $\eta_{\mathcal{A}}(t)$. For $\mathcal{A}$ to have positive probabilities, the non-decreasing functions $\nu_{\mathcal{A}}$ and $\eta_{\mathcal{A}}$ must satisfy some additional conditions, namely, which $\nu(t)-\nu(t-1) \in \{0, 1\}$ and $(\eta(t)-\eta(t-1))/2$ must belong to the support of the probability distribution $\mathbf{p}$ or $\mathbf{q}$, depending on $\nu(t) - \nu(t-1)$.

With $\mathcal{A}$ fixed, we denote by $V(t) = V_{\mathcal{A}}(t) = \nu_{\mathcal{A}}(t) - \nu_{\mathcal{A}}(t-1)$ the number of new vertices appearing at time t, and $E(t) = (\eta_{\mathcal{A}}(t) - \eta_{\mathcal{A}}(t-1))/2$ the number of new edges appearing at time t. We call $\mathcal{A}$ a *partial history* up to time t_1, and note that $\mathcal{A}$ can equivalently be defined by the pair of functions $(V(t), E(t))_{t \le t_1}$ or (ν, η).

Now, conditional on $\mathcal{A}$, we can build a faithful model of the graph process $(G_t)_{1 \le t \le t_1}$, by using the following collection of independent random variables for each time $t \le t_1$: B_t, Γ_t, Δ_t, U_t, P_t, $U'_{k,t}$ $(1 \le k \le E(t))$, and $P'_{k,t}$ $(1 \le k \le E(t))$. B_t, Γ_t, and Δ_t are Bernoulli variables with respective parameters β, δ and γ, U_t and $U'_{k,t}$ are uniform on the set $\{1, \dots, \nu(t-1)\}$, and P_t and $P'_{k,t}$ are uniform on the set $\{1, \dots, \eta(t-1)\}$.

We define the endpoints of edges added at time t so that new edge number k is between endpoints $e_{\eta(t-1)+2k-1}$ and $e_{\eta(t-1)+2k}$. The edges added at time t are then defined as follows:

Vertex insertion If $V(t) = 1$ (procedure NEW), the $E(t)$ edges are added between the new vertex $\nu(t)$ and vertices chosen uniformly or preferentially, depending on B_t:

- $e_{\eta(t-1)+2k-1} = \nu(t)$;
- $e_{\eta(t-1)+2k} = U'_{k,t}$ if $B_t = 1$, or $e_{P'_{k,t}}$ if $B_t = 0$.

Edge insertion If $V(t) = 0$ (procedure OLD), the $E(t)$ edges are added between one old vertex (chosen uniformly or preferentially, depending on Γ_t) and other vertices (chosen uniformly or preferentially, depending on Δ_t):

- $e_{\eta(t-1)+2k-1} = U_t$ if $\Gamma_t = 1$, or e_{P_t} if $\Gamma_t = 0$;
- $e_{\eta(t-1)+2k} = U'_{k,t}$ if $\Delta_t = 1$, or $e_{P'_{k,t}}$ if $\Delta_t = 0$.

To obtain a model for the unrestricted process (up to time t_1), one would simply take the disjoint union of all restricted models (all choices of $(\nu_{\mathcal{A}}, \eta_{\mathcal{A}})$ for which $\mathcal{A}$ has positive probability), and select $\mathcal{A}$ with suitable probability, as described earlier.

3.5.3 Vertex equivalence in the Cooper-Frieze model

We are now ready to prove a conditional equivalence for vertices in the Cooper-Frieze model.

Definition 3.12. Let $\mathcal{A} = (\nu_{\mathcal{A}}, \eta_{\mathcal{A}})$ be some partial history up to time t_1, and σ some permutation of the integers $\{2, \ldots, \nu_{\mathcal{A}}(t_1)\}$.

We denote by $\mathcal{A}_\sigma$ the partial history (and event) determined by the suitably permuted functions (V', E') defined by

- if $V(t) = 0$ (that is, at time t, no new vertex is added in $\mathcal{A}$), then $V'(t) = 0$ and $E'(t) = E(t)$;
- if $V(t) = 1$ (the graph gets a new vertex at time t in $\mathcal{A}$), then $V'(t) = 1$ and $E'(t) = E(t')$, where t' is defined as the insertion time of vertex $\sigma^{-1}(\nu(t))$ in $\mathcal{A}$:

$$t' = \inf\{\tau : \nu_{\mathcal{A}}(\tau) = \sigma^{-1}(\nu_{\mathcal{A}}(t))\}.$$

Note that, since $\mathcal{A}_\sigma$, compared to $\mathcal{A}$, only corresponds to a reordering of the numbers of inserted edges at times at which vertices are added, we have $\Pr(\mathcal{A}) = \Pr(\mathcal{A}_\sigma)$.

We are now ready to prove our conditional equivalence result for the Cooper-Frieze model.

Lemma 3.13. *Let $t_0 < t_1$ be fixed times, and let $\mathcal{A}$ denote some partial history up to time t_1. Define the event*

$$\mathcal{E}_{\mathcal{A},t_0,t_1} = \mathcal{A} \cap \bigcap_{t_0 < t \le t_1} \{U_t \le \nu_{\mathcal{A}}(t_0), U'_{k,t} \le \nu_{\mathcal{A}}(t_0), P_t \le \eta_{\mathcal{A}}(t_0), P'_{k,t} \le \eta_{\mathcal{A}}(t_0)\}.$$

Then, for any permutation $\sigma \in \mathcal{S}_{\{\nu_{\mathcal{A}}(t_0)+1,\ldots,\nu_{\mathcal{A}}(t_1)\}}$, and any graph g

$$\Pr(G_{t_1} = g | \mathcal{E}_{\mathcal{A},t_0,t_1}) = \Pr(G_{t_1} = \sigma(g) | \mathcal{E}_{\mathcal{A}_\sigma,t_0,t_1}).$$

Proof. Conditional on $\mathcal{E}_{\mathcal{A},t_0,t_1}$, all U_t and $U'_{k,t}$ variables (for $t_0 < t \le t_1$) are uniform on $\{1, \ldots, \nu_{\mathcal{A}}(t_0)\}$, and all P_t and $P'_{k,t}$ variables (for $t_0 < t \le t_1$) are

uniform on $\{1, \ldots, \eta_{\mathcal{A}}(t_0)\}$. But since σ only exchanges times later than t_0, we have $\nu_{\mathcal{A}_\sigma}(t_0) = \nu_{\mathcal{A}}(t_0)$ and $\eta_{\mathcal{A}_\sigma}(t_0) = \eta_{\mathcal{A}}(t_0)$, so that the same holds for the event $\mathcal{E}_{\mathcal{A}_\sigma,t_0,t_1}$. Thus, from time t_0 to t_1, the two conditional graph processes (which are equal at time t_0) evolve by using identically distributed sequences of random variables, which implies that they have the same distribution at time t_1. □

Now consider the events

$$\begin{aligned} \mathcal{A}' &= \bigcup_{\sigma \in \mathcal{S}_{\{\nu_{\mathcal{A}}(t_0)+1,\ldots,\nu_{\mathcal{A}}(t_1)\}}} \mathcal{A}_\sigma \\ \mathcal{E}'_{\mathcal{A},t_0,t_1} &= \bigcup_{\sigma \in \mathcal{S}_{\{\nu_{\mathcal{A}}(t_0)+1,\ldots,\nu_{\mathcal{A}}(t_1)\}}} \mathcal{E}_{\mathcal{A}_\sigma,t_0,t_1}. \end{aligned}$$

Note that two $\mathcal{A}$ (respectively $\mathcal{E}$) events appearing in the union are either equal (if the action of permutations σ and σ' on the sequence $(V_{\mathcal{A}}(t), E_{\mathcal{A}}(t))_{t_0<t\le t_1}$ coincide), or are disjoint. Thus, applying Lemma 3.13 and (the proof of) Lemma 3.3 to each possible σ, we obtain the following result.

Lemma 3.14. *Conditional on $\mathcal{E}'_{\mathcal{A},t_0,t_1}$, the vertices in $\{\nu_{\mathcal{A}}(t_0)+1, \ldots, \nu_{\mathcal{A}}(t_1)\}$ are equivalent in G_{t_1}.*

Now we have some conditional equivalence, but the conditioning event has very small probability. We will use Lemma 3.3 to extend it to an event of much larger probability by taking a union over a large set of choices of $\mathcal{A}$, but first we will prove a lower bound on the probability of $\mathcal{E}'_{\mathcal{A},t_0,t_1}$ conditional on $\mathcal{A}'_{t_0,t_1}$.

Lemma 3.15.

$$\Pr(\mathcal{E}'_{\mathcal{A},t_0,t_1}|\mathcal{A}') \ge \left(\frac{\nu_{\mathcal{A}}(t_0)\eta_{\mathcal{A}}(t_0)}{\nu_{\mathcal{A}}(t_1)\eta_{\mathcal{A}}(t_1)}\right)^{t_1-t_0-(\nu(t_1)-\nu(t_0))+(\eta_{\mathcal{A}}(t_1)-\eta_{\mathcal{A}}(t_0))/2}$$

Proof. We prove the inequality for $\mathcal{E}_{\mathcal{A},t_0,t_1}$ and $\mathcal{A}$, and extend it by convexity and disjoint unions.

To see that the inequality is true for $\Pr(\mathcal{E}_{\mathcal{A},t_0,t_1}|\mathcal{A})$, note that we are conditioning on a total of $t_1 - t_0 - (\nu(t_1) - \nu(t_0))$ variables of type U_t (respectively,

P_t) and $(\eta(t_1) - \eta(t_0))/2$ variables of type $U'_{k,t}$ (respectively $P'_{k,t}$), each of which is uniform on a set of size at most $\nu(t_1)$ (respectively $\eta(t_1)$), and is conditioned to take a value in a set of size $\nu(t_0)$ (respectively $\eta(t_0)$). The lower bound immediately follows. □

3.5.4 Lower bound on searching time

We are now able to prove our main result on the searching time in the Cooper-Frieze graph. One small issue with this model is that we do not know how many vertices are present at each time. We assume that if a vertex that does not exist is requested then the process returns this information immediately.

Theorem 3.16. *In $G_{\lfloor \frac{2n}{1-\alpha} \rfloor}$ with $0 < \alpha < 1$, no searching algorithm can find a path to vertex n, using only local information in the weak model, with an expected number of requests less than $\Omega(n^{1/2})$.*

Proof. Let $\mu_{\mathbf{p}}$ and $v_{\mathbf{p}}$ be the mean and variance, respectively of the number of edges generated by a new vertex and let $\mu_{\mathbf{q}}$ and $v_{\mathbf{q}}$ denote the mean and variance, respectively, of the number of edges generated by an old vertex.

Note that, in the unrestricted process, $\nu(t)$ follows the binomial distribution with parameters $1 - \alpha$ and t, and $\eta(t)/2$ is the sum of t independent random variables with expectation $m = (1 - \alpha)\mu_{\mathbf{p}} + \alpha\mu_{\mathbf{q}}$ and variance

$$v = \alpha(1 - \alpha)\left(\mu_{\mathbf{p}} - \mu_{\mathbf{q}}\right)^2 + (1 - \alpha)v_{\mathbf{p}} + \alpha v_{\mathbf{q}}.$$

Thus, by the standard Central Limit Theorem both $(\eta(t)/2 - mt)/\sqrt{vt}$ and $(\nu(t) - (1 - \alpha)t)/\sqrt{\alpha(1 - \alpha)t}$ converge to the standard normal distribution. We now have to define a set of partial histories with large enough probability, each of which makes a given set of vertices conditionally equivalent. For a given (large enough) n, this set of vertices will be vertices $\{n, \ldots, \lfloor n + \sqrt{n} \rfloor\}$. As a consequence, for any $0 < \epsilon < 1$, there exists a $\lambda = \lambda(\epsilon) > 0$ such that, for n large enough, the probability is at least ϵ that, setting $t_0 = \lfloor \frac{n}{1-\alpha} - \lambda\sqrt{n} \rfloor$

and $t_1 = \lfloor \frac{n}{1-\alpha} + \left(\frac{1}{1-\alpha} + \lambda\right)\sqrt{n} \rfloor$, we have

$$\begin{aligned}
\nu(t_0) &\leq n && (3.17)\\
\nu(t_0) &\geq n - 2(1-\alpha)\lambda\sqrt{n} && (3.18)\\
\nu(t_1) &\geq n + \sqrt{n} && (3.19)\\
\nu(t_1) &\leq n + (1 + 2(1-\alpha)\lambda)\sqrt{n} && (3.20)\\
\eta(t_0)/2 &\geq mt_0 - \lambda\sqrt{n} && (3.21)\\
\eta(t_1)/2 &\leq mt_1 + \lambda\sqrt{n} && (3.22)
\end{aligned}$$

Taking as $\mathcal{E}$ the event that (3.17-3.22) are satisfied, we see that $\mathcal{E}$ is an event of probability at least ϵ, which is the disjoint reunion of partial histories (up to time t_1). For each $\mathcal{A}$ in this set of partial histories, Lemma 3.13 tells us that a set of vertices including $\{n, \ldots, \lfloor n + \sqrt{n} \rfloor\}$ (because of (3.17) and (3.19)) is equivalent conditional on $\mathcal{E}'_{\mathcal{A},t_0,t_1}$, and Lemma 3.15, together with inequalities (3.21) and (3.22), gives us a lower bound for $\Pr(\mathcal{E}'_{\mathcal{A},t_0,t_1}|\mathcal{A}')$ of the form

$$\left(\frac{n - 2(1-\alpha)\lambda\sqrt{n}}{n + (1 + 2(1-\alpha)\lambda)\sqrt{n}} \frac{mt_0 - \lambda\sqrt{n}}{mt_1 + \lambda\sqrt{n}}\right)^{c' n^{1/2}} \geq \left(1 - \frac{c''}{\sqrt{n}}\right)^{c'\sqrt{n}} \geq c$$

for some positive constants c, c', c''.

Taking the disjoint reunion of all these $\mathcal{E}'_{\mathcal{A},t_0,t_1}$, we find an event of probability at least $c\Pr(\mathcal{E}) \geq \epsilon c$, conditional on which the vertices $\{n, \ldots, n + \lfloor\sqrt{n}\rfloor\}$ are equivalent, proving Theorem 3.16. □

3.6 Optimal search algorithm for the Móri tree

For $\beta > 0$ we proved, in Section 3.4, a general lower bound of $\Omega(n^{\frac{\beta}{2\beta+4}-\epsilon})$ for searching for vertex v_n in the merged Móri graphs, valid for any finite $m \geq 1$. However, for $\beta \geq 0$ and $m = 1$, when the search starts from vertex v_1, we can prove a much higher lower bound of $\Omega(n)$ by exhibiting a very simple algorithm, operating under the strong model, that is optimal in a very strong

sense.

Any searching process, be it fully deterministic or randomised, can be described as a *searching strategy*, which, for each time k, maps each possible execution of the searching process up to time k (each alternating sequence of possible requests and answers to these requests) to one of the legal choices for the next request, or, in the case of a randomised search, each possible execution is mapped to a probability distribution on the set of legal choices for the next request. Note that we discuss searching *processes* instead of *algorithms* because we do not need any assumption on the computability of the mapping corresponding to a strategy: our optimal algorithm is deterministic and obviously computable with a very low complexity. We let $t(S, T)$ denote the number of requests made by strategy S on tree T. We assume that the searching strategy and the tree construction process work independently and do not "share coin flips".

One possible strategy S_0 is the following: at each time step, always request the lowest possible index among vertices which have not been requested yet, and for which a neighbour has been visited, until the target is found. Since each vertex other than the root v_1 has a neighbour (its single out-neighbour) with a strictly lower label, this strategy, starting from vertex v_1, will obviously result in requesting vertex v_k at time k, until the target v_n is discovered at time f_n. (Recall from the previous chapter that f_n denotes the index of the out-neighbour of v_n.) The expected number of requests made by S_0 is thus $\mathbf{E}[f_n]$, which is clearly $\Omega(n)$, because with probability $\frac{n-2}{(2+\beta)(n-1)-2}$ the out-neighbour of v_n is chosen preferentially by copying the tail of the out-going edge of the vertices $v_1, \ldots, v_{n-1}$.

The somewhat surprising result is that S_0 is the best possible algorithm, not just in expectation, but in a much stronger sense.

Theorem 3.23. *For any possible (deterministic or randomised) strategy S and any value of $\beta > 0$, the number of requests made by S on the random tree $G^n_{1,\beta}$ stochastically dominates the number of requests made by S_0 on the same random graph model.*

Recall that a random variable X *stochastically dominates* a random variable Y (which we will write $Y \preceq X$) if, for any value x, the event $\{X \geq x\}$

has probability no less than the probability of the event $\{Y \geq x\}$.

What Theorem 3.23 means is not that S_0 performs better than any other possible strategy on any possible tree (obviously, for any possible tree T, the deterministic strategy that "magically" requests only the vertices on the path between v_1 and v_n does better than S_0 on this particular graph); it is only that no other possible strategy, when applied to a random Móri tree (and thus restricted to only using whatever information it has "learned" on the realisation T of the random tree model), will have a better chance than S_0 of terminating within a given time limit. It is still a very strong result, since it means that there is never a good reason not to follow strategy S_0.

Proof. In order to prove the inequality $t(S_0, G^n_{1,\beta}) \preceq t(S, G^n_{1,\beta})$, we construct, for any strategy S that has positive probability of not agreeing with S_0 at some point, two new strategies S' and S'' with the following properties. (Strictly speaking S' does not satisfy our definition of a strategy since it may keep making requests even when the target has appeared in the answer to a request.)

- $t(S', G^n_{1,\beta}) \overset{(d)}{=} t(S, G^n_{1,\beta})$.
- For any possible tree T, the inequality $t(S'', T) \leq t(S', T)$ holds with probability 1.
- On any possible tree T, S and S' agree until S has positive probability of disagreeing with S_0, at which time S' and S'' deterministically agree with S_0.

Before describing the construction of S' and S'', let us first see why they imply the stochastic dominance claim. For any possible strategy S, let $\tau_n(S)$ denote the first possible time where S can (with positive probability) disagree with S_0 when dealing with a tree of size at most n. Since S_0 will always find its target in at most n requests, any strategy S for which $\tau_n(S) \geq n$ will deterministically agree with S_0 until termination on any tree of size at most n.

Notice that the first two conditions on S' and S'' imply $t(S'', G^n_{1,\beta}) \preceq t(S, G^n_{1,\beta})$, and that the third implies that $\tau_n(S'') \geq 1 + \tau_n(S)$ unless S

agrees deterministically with S_0 on all trees up to size n. Thus, applying the above construction n times to any S, we get a strategy $S^{(n)}$ with $t(S^{(n)}, G^n_{1,\beta}) \preceq t(S, G^n_{1,\beta})$ that agrees with S_0 on all trees of size up to n, implying $t(S_0, G^n_{1,\beta}) = t(S^{(n)}, G^n_{1,\beta})$. This completes the stochastic dominance proof.

We now turn to the construction of S' and S''. The idea is this: on any tree T, S' acts exactly as S until it terminates or it has positive probability of disagreeing with S_0, whichever comes first. At this point, S' follows S_0 for one more step, after which it determines a random tree T', with the same probability distribution as $G^n_{1,\beta}$ conditional on the common past execution of S and S', and starts simulating the behaviour of S *on* T' *instead of* T, making the requests in T that S would make if presented with the tree T', until the simulated version of S finds the target and terminates. The construction ensures at each time-step, the vertices that are discovered by the simulated S in T' are amongst those that have been discovered by S' in T. The strategy S'' is the same as S', except that it very sensibly terminates as soon as it has found v_n, instead of waiting until the simulated S does.

All that remains is to describe how this simulation can be done and to verify that the three properties hold. Recall the description of the Móri tree in terms of the random variables $f_2, \ldots, f_n$ from Section 2.2. For ease of description we will use N_i and N'_i to denote the indices of the out-neighbour of v_i in T and T' respectively.

Suppose that on T the first time there is positive probability of a disagreement between S_0 and S comes after k steps, so that, by this point, both S and S_0 know every vertex whose out-going edge goes to a vertex of index at most k. In this situation, S_0 selects v_{k+1} for its next request and S requests some vertex $v_{k'}$.

At this point, S' requests vertex v_{k+1}, thus satisfying the third condition. As a result, it learns the indices of all vertices v_j in the "real" graph T such that $N_j = k+1$. Now S' constructs a new tree T' which, conditional on what is known to both S and S' at time k, has the same probability of occurring as T. If the strategy S is probabilistic then it is possible that $k' = k+1$, in which case $T' = T$, and the strategies S' and S are the same from this

point. Otherwise $k' > k+1$ and we focus on this case from now on. To see how the construction works it may be helpful to recall the partition of half-edges around a vertex used in the proof of the concentration result in the previous chapter. Assume for the moment that S' has access to the values of the random variables governing the construction of T, in particular whether choices of neighbour are made uniformly or preferentially. The incoming edges at v_{k+1} are partitioned into two sets A and B. An edge is placed in A if the other endpoint of the edge is v_j for $k+1 < j \leq k'$ or if v_{k+1} is chosen as the out-neighbour of a new vertex by preferentially copying an edge in A. All the other incoming edges at v_{k+1} are placed in B. The tree T' is now formed by moving all the edges in B from v_{k+1} to $v_{k'}$ and moving all the incoming edges at $v_{k'}$ to v_{k+1}. All other edges remain unchanged from T.

Let $\Omega' \subset \Omega$ denote the set of construction processes of $G^n_{1,\beta}$ that result in a tree such that the neighbours of $v_1, \ldots, v_k$ are as in T. Then the construction of T' from T can be described in terms of an involution ϕ on Ω'. If an edge is moved then the corresponding value of f_i may need to be altered accordingly. For instance any edge that is moved for which $f_i = (k+1, v)$, indicating that v_{k+1} was chosen uniformly as an out-neighbour, is replaced by (k', v) (and vice versa). Similarly any edge that is moved for which $f_i = (k+1, t)$, indicating that the head of the edge directed away from v_{k+1} was selected preferentially, is replaced by (k', t) (and vice versa). Clearly $\Pr(\omega) = \Pr(\phi(\omega))$ and so T and T' have the same probability distribution.

Of course, S' does not have access to the construction process, only to the resulting tree, so T' cannot be constructed in exactly this way. The strategy S' has to resimulate the construction process. It is not necessary to simulate the whole process, merely to determine which edges should be switched from v_{k+1} to $v_{k'}$. Suppose the vertices sending their out-going edges to v_{k+1} in T are $v_{r_1}, \ldots, v_{r_l}$ where $r_1 < \cdots < r_l$. For each i, $1 \leq i \leq l$, in increasing order, S' decides whether to place the edge $v_{k+1}v_{r_i}$ into A or B. If $r_i \leq k'$ then the edge $v_{k+1}v_{r_i}$ is placed into A. Otherwise let j_i denote the size of A just before $v_{k+1}v_{r_i}$ has been placed. Then place $v_{k+1}v_{r_i}$ into A with probability $j_i/(i+\beta)$ and into B with the complementary probability $(i-j_i+\beta)/(i+\beta)$. So given a tree T, the probability distribution of the set of edges that are

moved from v_{k+1} to $v_{k'}$ is the same when S' simulates the construction process as when S' had access to the values of the random variables determining the construction. Consequently the probability distributions of T and T' are the same.

Now S' simulates in T' how S would behave after having made requests for the neighbours of $v_1, \ldots, v_k$ at times $1, \ldots, k$ and then $v_{k'}$ at time $k+1$. At each time after $k+1$, S' makes whatever request S would select if confronted with tree T' except that if S requests v_{k+1} then S' requests $v_{k'}$ in T. When the simulated S finds the target, S' terminates. If S is probabilistic, then S' has to use the same probability distribution for its random choices. Notice that at each time t the set of vertices discovered up to time t by the strategy S in T' is a subset of the set of vertices discovered by S' in T so each request made by S' is valid. Furthermore when S' terminates, it has found the target. (At time $k+1$, the strategy S' may discover more vertices, including possibly the target v_n, than the simulated strategy S operating in T', depending on whether or not the set A is empty.) Since T and T' have the same distribution and the time taken by S' on T is exactly the same as the time taken by the simulated copy of S on T' we have $t(S', G^n_{1,\beta}) \stackrel{(d)}{=} t(S, G^n_{1,\beta})$, satisfying the first property required.

For the definition of S'', follow the same rules as for S', except that S'' always terminates once it has found the target vertex. Thus, with probability 1, S'' will terminate no later than S', which is exactly the second condition we imposed. This completes the description of S' and S'', and the proof of the theorem.

□

3.7 Conclusion and open problems

We have proved, in our weak model of local information a polynomially high lower bound on the time required to find a given vertex in the model of scale-free graphs considered. This is in contrast with all proved upper bound results on the *diameter* of such graphs, which are at most logarithmic in expectation and with high probability. That is, these random graph models

cannot be searched in time polynomial in their diameter. Thus, these graph models do not have the "small world" easy searchability property as exhibited by several authors in different models of random graphs. The technique we used seems broad enough to be adapted to different models of growing random graphs.

For the evolving graph model, our results carry over somehow weakly to the (arguably more realistic) strong model of local information, whenever the maximum degree can be proved to be significantly smaller than $n^{1/2}$.

It is still open to know whether the right lower bound is always linear for the Móri graph in the strong model. The case $m = 1$ is special since between two vertices, there is a single path. The multiplicities of short paths for $m > 1$ could drastically change the bounds.

Chapter 4

The complexity of two graph orientation problems

4.1 Introduction

In this chapter, which is joint work with Steven Noble, Brunel University, we turn to a very different topic. An extended abstract based on this chapter is published in [28]. We consider two problems concerned with orienting the edges of an undirected graph in order to minimise two global measures of distance in the resulting directed graph. Our work is motivated by an application involving the design of urban light rail networks. In such an application a number of stations are to be linked with unidirectional track in order to minimise some function of the travel times between stations and subject to constraints on cost, engineering and planning. In practice these constraints mean that the choice of which stations to link may be forced upon us and the only control we have is over the choice of direction of each piece of track. Since the stations that are linked tend to be those that are close to each other, we make the simplifying assumption that the travel time along each single piece of track or link is the same. Consequently the network can be viewed as an (unweighted) graph in which the vertices represent stations and the edges represent track that is to be built. Furthermore planning constraints tend to rule out the possibility of tracks crossing so the graph is usually planar. The aim is to orient the resulting graph to minimise travel

time. We assume that each journey in the oriented network progresses along the shortest directed path from the vertex representing the starting station to the vertex representing the destination.

Unless stated otherwise, all our graphs are simple that is they have no loops or parallel edges. When the underlying graph is obvious, we use n and m to denote its numbers of vertices and edges, respectively. We use (G, ω) to denote the directed graph obtained by applying the orientation ω to the edges of G. Sometimes we abbreviate this to $\vec{G}$. We let $d(x, y)$ denote the distance from vertex x to vertex y in a directed graph. The two measures of the quality of an orientation are its *diameter* $\mathrm{diam}(\vec{G})$, given by

$$\mathrm{diam}(\vec{G}) = \max_{x \neq y} d(x, y)$$

and the *Wiener Index* $Z(\vec{G})$, given by

$$Z(\vec{G}) = \sum_{x \neq y} d(x, y).$$

The name Wiener Index is perhaps not widely used but is more common in applications in chemistry. The networks arising in the application tend to be planar and have small degree. Our original aim was to determine the complexity of minimising $\mathrm{diam}(\vec{G})$ and $Z(\vec{G})$ for planar graphs of bounded degree. We have two partial results in this direction. In the next section we give a brief overview of computational complexity. Following that, we show that the problem where we are given a graph G and an integer k and must determine whether G can be oriented so that its Wiener Index is at most k, is NP-complete. We believe that it should be possible to sharpen this result so that the input graph is restricted to having degree at most three but our idea for a proof became extremely complicated, so we have not pursued this. We do not know whether the input may be restricted to planar graphs. Our hardness result depends on a result of Chvátal and Thomassen [19] who showed that determining whether a graph may be oriented so that its diameter is at most two is NP-complete. A result of Bollobás and Scott [12]

shows that an oriented graph with diameter two and n vertices must have at least $(1 + o(1))n \log_2 n$ edges. Since a planar graph with n vertices has at most $3n - 6$ edges, this implies that there is a constant upper bound on the number of vertices in a planar oriented graph with diameter two. So there is a constant time algorithm to determine whether a planar graph can be oriented so that its diameter is at most two. However there are arbitrarily large planar graphs that can be oriented so that their diameter is three or four, for example, a set of triangles sharing two common vertices or a common vertex, respectively. The rest of the chapter is devoted to the contrasting result that for any fixed constant l, there is a polynomial time algorithm that will take a planar graph and determine whether it may be oriented so that its diameter is at most l. The algorithm relies on graph minor theory [56, 72, 74]. In Section 4.4 we discuss necessary concepts from tree-width. In Section 4.5 we describe an algorithm that attempts to find a suitable orientation when the input graph has bounded tree-width. Section 4.6 contains our main result for planar graphs.

After completing this work we discovered that the techniques that we employ here have been used before, perhaps most notably to find a k-dominating set in a planar graph [2, 32]. A general setting for these techniques is introduced in [22].

4.2 Complexity

In this section we give a short introduction to the theory of NP-completeness, which concerns the complexity of decision problems, that is, problems for which the answer is always *yes* or *no*. We briefly summarise the explanation of the theory of NP-completeness in [35, Chapter 2] and mainly use the same notation given there. For more information see also [63].

Let Σ be a finite set of symbols, then Σ^* is the set of all finite strings of symbols from Σ. Then L is a *language* over the alphabet Σ if $L \subseteq \Sigma^*$. Suppose a program M of a Turing machine halts for all $x \in \Sigma^*$ with possible halt-states q_N and q_Y. The language L_M is defined to be set of elements in Σ^* for which the output of M is q_Y. The time complexity of a program

M is a function $T_M(n)$ that maps each positive integer n onto the maximal time it can take to obtain the output for an input of length n. If $T_M(n)$ can be bounded above by a polynomial then M is called a *polynomial time deterministic Turing machine program*. Then P is formally defined to be the class of all languages L for which there is a polynomial time Turing machine program M. Essentially any computation, which can be performed on a computer, can also be done on a Turing machine. Therefore an informal definition of P is the class of decision problems which can be solved in time bounded by a polynomial in the size of the input.

A nondeterministic algorithm consists of a guessing stage and a checking stage. Let I be an instance of a decision problem. First a structure S is guessed. Then I and S are the inputs of the checking stage which is a deterministic process and halts with answer *yes* or *no* or never halts. A nondeterministic algorithm is said to "solve" a decision problem if for all instances of the problem the following holds:

- If the correct answer to instance I of the decision problem is yes then there exists a structure S, a possible guess, which will yield the answer *yes* in the checking stage with input S and I.
- If the correct answer to instance I of the decision problem is no then there exists no structure S which will yield the answer *yes* in the checking stage with input S and I.

A nondeterministic algorithm is said to solve a decision problem in polynomial time if for every instance I, that has a yes-answer, there is a guessed structure S so that the checking stage yields a yes-answer to the input I and S in time bounded by a polynomial in the size of I. The class NP is informally defined to be the class of decision problems that can be solved in polynomial time by a nondeterministic algorithm. So the class NP consists of those decision problems for which any solution can be verified in time bounded by a polynomial in the size of the input. We omit the formal definition of the class NP which depends on the so-called nondeterministic Turing machine and yields essentially the same thing.

Let Π_1 and Π_2 be two decision problems. A *polynomial transformation* is a function f that maps the set of instances of Π_1 into the set of instances of Π_2, is computable in polynomial time and an instance I is a yes-instance of Π_1 if and only if $f(I)$ is a yes-instance of Π_2. In this case we write $\Pi_1 \propto \Pi_2$. Note that if $\Pi_2 \in \mathrm{P}$ then $\Pi_1 \in \mathrm{P}$. It can easily be shown that if $\Pi_1 \propto \Pi_2$ and $\Pi_2 \propto \Pi_3$, then $\Pi_1 \propto \Pi_3$.

The class of NP-complete problems is defined to be the class of decision problems Π which are in NP and for all other $\Pi' \in \mathrm{NP}$, $\Pi' \propto \Pi$. So the NP-complete problems are the hardest problems in NP.

Suppose a problem Π is known to be NP-complete and a problem Π' is known to be NP. If it can be shown that $\Pi \propto \Pi'$ then it follows from the transitivity of $\propto$ that Π' is NP-complete.

One can easily verify that $\mathrm{P} \subseteq \mathrm{NP}$. However it is still an open problem to determine whether $\mathrm{P} = \mathrm{NP}$. Note that, providing $\mathrm{NP} \neq \mathrm{P}$, there are problems in NP that are neither in P nor NP-complete [54].

The class of NP-hard problems is not just restricted to decision problems. A problem Π is NP-hard if the existence of a polynomial time algorithm for Π would imply the existence of a polynomial time algorithm for some NP-complete problem. Clearly all NP-complete problems are NP-hard.

4.3 Complexity of the Wiener Index

Imagine we are given a graph and an integer k and we would like to know whether the graph can be oriented in such a way that the Wiener Index is less than k. In this section we investigate the complexity of this problem.

Chvátal and Thomassen [19] showed that the following problem is NP-complete.

Problem 4.1.
Instance: A graph G.
Question: Is it possible to orient G to ensure that $\mathrm{diam}(\vec{G}) \leq 2$*?*

From this result we can easily conclude that the following problem concerning the Wiener Index is NP-complete.

Problem 4.2.
Instance: A graph G, integer k.
Question: Is it possible to orient G to ensure that the Wiener Index of $\vec{G}$ is at most k?

Theorem 4.3. *Problem 4.2 is NP-complete.*

Proof. The problem is clearly in NP. Suppose that G has m edges. Let $k = 2(n^2 - n) - m$. We count the number of pairs of vertices joined by paths of small lengths. There are m pairs of vertices joined by a path of length 1 in $\vec{G}$. If $\mathrm{diam}(\vec{G}) \leq 2$ then all the remaining pairs of vertices are joined by paths of length two. So $Z(\vec{G}) = 2(n^2 - n) - m = k$. Conversely if $\mathrm{diam}(\vec{G}) > 2$, there are $n^2 - n - m$ pairs of vertices joined by paths of length at least two including at least one path of length at least three, so $Z(\vec{G}) > 2(n^2 - n) - m = k$. Consequently there is an orientation of G with $\mathrm{diam}(\vec{G}) \leq 2$ if and only if there is an orientation of G with $Z(\vec{G}) \leq k$. □

We have been unable to determine the complexity of the following problem.

Problem 4.4.
Instance: Planar graph G and integer k.
Question: Can we orient the edges of G so that $Z(\vec{G}) \leq k$?

The rest of the paper is dedicated to the investigation of the complexity of the following problem for any fixed integer l.

Problem 4.5.
Instance: Planar graph G.
Question: Can we orient the edges of G so that $\mathrm{diam}(\vec{G}) \leq l$*?*

There is a considerable amount of work on this topic, see for instance the survey of Koh and Tan [48]. Much of the focus has been on very specific classes of graphs. For instance the optimal orientation of a complete graph [64, 58, 66, 8], a complete bipartite graph [8, 65, 78, 37], the square lattice [68, 69, 70, 71] and the hypercube [65, 78, 59] are known. Partial results have been obtained for complete multipartite graphs [65, 37, 38, 39, 44, 45]

and the toroidal lattice [46, 47, 51]. It is known that for $k \geq 4$, it is NP-complete to determine whether a chordal graph has an orientation of diameter at most k [31].

4.4 Tree-decompositions

The notion of a tree-decomposition was developed by Robertson and Seymour in [72]. Good introductions to the theory of tree-decompositions can be found, for example, in [5], [56] and [77]. The definition of a tree-decomposition is as follows.

Definition 4.6. A tree-decomposition $\mathcal{T}$ of a graph G is a pair $(T, \mathcal{W})$ where T is a tree and $\mathcal{W} = (W_t : t \in V(T))$ is a family of subsets of $V(G)$ such that:

- $\bigcup_{t \in V(T)} W_t = V(G)$ and every edge in G has both endpoints in W_t for some t;
- if $t, t', t'' \in V(T)$ and t' lies on the path from t to t'' in T then $W_t \cap W_{t''} \subseteq W_{t'}$.

The width of $(T, \mathcal{W})$ is defined to be

$$\max\{|W_t| - 1 : t \in V(T)\}.$$

The tree-width $tw(G)$ of G is the minimum width among all possible tree-decompositions of G.

One reason for the importance of tree-width is that many NP-hard problems can be solved in polynomial or even linear time when restricted to graphs of bounded tree-width [77], [5]. For the algorithm we give in Section 4.5, the following features of tree-decompositions will be useful. We call a tree-decomposition $\mathcal{T} = (T, \mathcal{W})$ which fulfils the following properties a *reduced tree-decomposition of width k*.

- T is a rooted tree with root r.

- For all $i \in V(T)$, $|W_i| = k + 1$.
- For all $i \in V(T)$, there is a leaf j of T such that $W_j = W_i$.
- For all $i \in V(T)$, either i is a leaf of T or i has exactly two children.
- $|V(T)| \leq 2n$.

Note that this definition deviates from the way in which a reduced tree-decomposition of width k is usually defined in the literature, namely as a tree-decomposition in which $W_i \not\subseteq W_j$ for all $i \neq j$.

In 1996 Bodlaender [6] gave a linear-time algorithm for finding tree decompositions of small width. Let

$$f(k) = k^5(2k+1)^{2k-1}((4k+5)^{4k+5}(\frac{8}{3} \cdot 2^{2k+2})^{4k+5})^{4k+1}.$$

Theorem 4.7. *Let k be a constant. Then there is an algorithm running in time $O(f(k)n)$ that inputs a graph G and determines whether the tree-width of G is at most k, and if so finds a tree-decomposition of G with tree-width at most k.*

It follows from the theorem that any tree-decomposition $\mathcal{T} = (T, \mathcal{W})$ obtained by Bodlaender's algorithm satisfies $\sum_{i \in T} |W_i| = O(f(k)n)$.

We will assume that if i, j are neighbouring vertices of T, we have $W_i \not\subseteq W_j$, because otherwise we may contract the edge in T and remove W_i from $\mathcal{W}$ as seen in [6]. Ensuring that this condition holds requires time $O(\sum_{i \in T} |W_i|) = O(f(k)n)$. (We assume that checking the membership of a set requires unit time.)

Some parts of the proof of the following lemma can also be found in [6].

Lemma 4.8. *Any tree-decomposition $\mathcal{T}' = (T', \mathcal{W}')$ of a graph G with width k, obtained by Bodlaender's algorithm, can be modified in time $O(kn)$ into a reduced tree-decomposition $\mathcal{T} = (T, \mathcal{W})$.*

Proof. First let $(T, \mathcal{W}) = (T', \mathcal{W}')$. Choose vertex $r \in V(T)$ arbitrarily to be the root of T and regard T as a rooted tree.

Now for each $i \neq r \in V(T)$ suppose that j is its parent in T. Mark i with an element of $W_i \backslash W_j$. Clearly such an element exists because $W_i \not\subseteq W_j$. Mark r with any element of W_r. Since no two marks are the same it follows that the tree contains at most n vertices, as also seen by Bodlaender [6].

Note that there is at least one vertex $i \in V(T)$ such that $|W_i| = k+1$. Suppose that there is a vertex $u \in V(T)$ with $|W_u| < k+1$ and such that u has a neighbour $v \in V(T)$ with $|W_v| = k+1$. Then we may add vertices of $W_v \backslash W_u$ to W_u so that $|W_u| = k+1$ and $(T, \mathcal{W})$ is still a tree-decomposition of G of width $k+1$. We can repeat this procedure until $|W_i| = k+1$ for all $i \in V(T)$. So we will assume that $|W_i| = k+1$ for all $i \in V(T)$. This feature was also proved earlier by Bodlaender [6].

For each vertex $i \in V(T)$ that is not a leaf, add a new vertex i^* joined only to i and a set $W_{i^*} = W_i$ to $\mathcal{W}$. We now have at most n leaves. Note that $(T, \mathcal{W})$ now satisfies the first three properties of a reduced tree-decomposition. Furthermore no $v \in V(T)$ has exactly one child. Suppose that $v \in T$ has $p \geq 3$ children, $v_1, \ldots, v_p$. Remove the edges $vv_2, \ldots, vv_p$ from T, add a new vertex v^* to $V(T)$ and add edges $vv^*, v^*v_2, \ldots, v^*v_p$. Finally add $W_{v^*} = W_v$ to $\mathcal{W}$. After a finite number of repetitions of this procedure, a reduced tree-decomposition $(T, \mathcal{W})$ of width k is produced. No new leaves are added in this procedure.

Since, in a binary tree, twice the number of leaf vertices is greater than the number of vertices in the tree we obtain $V(T) \leq 2n$. It follows that the overall time to obtain T is $O(kn)$, as there are at most $2n$ vertices in T. $\square$

Due to the third property of a reduced tree-decomposition $(T, \mathcal{W})$, we may arbitrarily associate any edge $vw \in E(G)$ with exactly one leaf i of T such that $\{v, w\} \subseteq W_i$. In this way we obtain a partition of the edges of G. Let E_i be the set of edges associated with a leaf i. For any non-leaf vertex v of T, let E_v be the set of all edges associated with a leaf i that is a descendant of v in T.

4.5 Algorithm DIAMETER

In this section we describe an algorithm that for fixed k, takes as input a graph G and a reduced tree-decomposition with width k and determines whether there is an orientation of G with diameter at most l running in time $O(cn)$, where c is a constant depending on k and l which we will determine.

Given a directed graph $\vec{G}$ we let $M'(\vec{G})$ be its shortest path matrix, that is the matrix whose rows and columns are both indexed by $V(\vec{G})$ with zeros on the diagonal and otherwise (x, y)-entry equal to $d(x, y)$. For the description of the algorithm we will need to introduce the *truncated distance* $d_l^*(x, y)$ from x to y given by $d_l^*(x, y) = \min\{l+1, d(x, y)\}$ if $d(x, y) < \infty$ and $d_l^*(x, y) = l+1$ if $d(x, y) = \infty$. Then the truncated distance matrix $M_l(\vec{G})$ is defined by replacing each distance $d(x, y)$ in $M'(\vec{G})$ by $d_l^*(x, y)$. Since l will always be the constant referred to in Problem 4.5, we will drop the subscript.

Now we describe the main idea behind the algorithm. Let $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, be graphs such that $V_1 \cap V_2 = X$, $E_1 \cap E_2 = \emptyset$ and let $G = G_1 \cup G_2$. So X is a vertex cut of G. Suppose we want to find whether we can orient G so that the diameter is less than or equal to the fixed constant l. It is possible to find the answer using certain information about all orientations of G_1 and G_2.

Suppose we know the following for $i = 1, 2$ and each orientation ω_i of G_i:

- the truncated distance from every vertex in $V_i \backslash X$ to each vertex in X and also the truncated distance in the reverse direction;
- the submatrix of the truncated distance matrix corresponding to vertices in X;
- which pairs of vertices in $V_i \backslash X$ are not joined to each other by a path of length at most l.

Then by considering every pair of orientations ω_1 of G_1 and ω_2 of G_2 we can determine whether G can be oriented as required. However we do not need this much information about G_1 and G_2. For instance, suppose that the truncated distances from (to) $v \in V_i \backslash X$ to (from) each member of X is specified by a vector $\overrightarrow{d}_v$ ($\overleftarrow{d}_v$). It turns out that we do not need the identity

of v, merely that there exists a vertex of V_i with these vectors of truncated distances to and from the members of X. As we will see, this means that the amount of information that must be stored by the algorithm at each stage is a function of k and l but not of n.

Given a graph G, $X \subset V(G)$ and ω an orientation of $E(G)$, we define the *characteristic* $c(G, X, \omega)$ as follows. Suppose without loss of generality that $X = \{v_1, \ldots, v_{k+1}\}$. Let $v \in V(G)\backslash X$. First we define the *distance vector of v to X* and the *distance vector of v from X*. Let $\overrightarrow{d}(G, X, \omega, v) = (\overrightarrow{d}_1, \ldots, \overrightarrow{d}_{k+1})$ and $\overleftarrow{d}(G, X, \omega, v) = (\overleftarrow{d}_1, \ldots, \overleftarrow{d}_{k+1})$ where for $1 \leq j \leq k+1$, $\overrightarrow{d}_j = d^*(v, v_j)$ and $\overleftarrow{d}_j = d^*(v_j, v)$. (In general $X \neq \{v_1, \ldots, v_{k+1}\}$ so we need to store information that make it possible to tell which entry of $\overrightarrow{d}(G, X, \omega, v)$ and $\overleftarrow{d}(G, X, \omega, v)$ corresponds to which vertex of X, but for simplicity we avoid discussing this.) Then $c(G, X, \omega)$ is a 4-tuple $(\overrightarrow{S}, \overleftarrow{S}, M, F)$ defined as follows:

- $\overrightarrow{S}$ and $\overleftarrow{S}$ are subsets of $[l+1]^{k+1}$ with $\overrightarrow{S} = \{\overrightarrow{d}(G, X, \omega, v) | v \in V(G)\backslash X\}$ and $\overleftarrow{S} = \{\overleftarrow{d}(G, X, \omega, v) | v \in V(G)\backslash X\}$.

- M is the submatrix of the truncated distance matrix of (G, ω) corresponding to the vertices of X.

- $F \subseteq \overrightarrow{S} \times \overleftarrow{S}$ such that $(s, t) \in F$ if and only if there are vertices $v, w \in V(G)\backslash X$ such that the distance vector of v to X is s and the distance vector of w from X is t and $d(v, w) > l$.

So F keeps track of pairs of vertices in $V(G)\backslash X$ that are not yet joined by a short enough path. We will see that it is not necessary to store the identities of these vertices, since the information in F is enough.

Let $\Omega(G)$ be the set of all possible orientations of G. Returning to the situation where G is the union of two edge-disjoint graphs G_1 and G_2 with $V(G_1) \cap V(G_2) = X$, the main idea behind the algorithm, corresponding to Lemma 4.12, below, is that the set $\{c(G, X, \omega) | \omega \in \Omega(G)\}$ can be computed quickly from $\{c(G_1, X, \omega) | \omega \in \Omega(G_1)\}$ and $\{c(G_2, X, \omega) | \omega \in \Omega(G_2)\}$.

Suppose we are given a reduced tree-decomposition $(T, \mathcal{W})$. For any $v \in T$, let $Y_v = \bigcup\{W_u | u = v$ or u is a descendant of $v\}$. Let G_v be the subgraph with vertex set Y_v and edge-set E_v (defined at the end of the previous section).

Define the characteristic of $v \in V(T)$ to be $c(v) = \{c(G_v, W_v, \omega) | \omega \in \Omega(G_v)\}$. Because the members of $c(v)$ are distinct characteristics, each one may correspond to many orientations of G_v. We will assume that we have a computational model that enables us to represent sets in an efficient way, in particular, it enables us to remove redundant repetition of elements.

The algorithm works upwards from the leaves of T computing $c(v)$ for each v until finally $c(r)$ is computed.

Let $h(l,k) = (l+1)^{k(k+1)} 2^{[2(l+1)^{k+1} + (l+1)^{2k+2}]}$. We will prove the following statements in order to show that the algorithm runs in linear time:

- $c(v)$ can be computed for a leaf node v in time $O(k^3 \cdot 2^{\frac{k^2+k}{2}})$;
- for any non-leaf v with children v_1 and v_2, $c(v)$ can be computed in time $O(k^2(l+1)^{2k+2} h(l,k)^2)$ from $c(v_1)$ and $c(v_2)$;
- from $c(r)$ we can determine in time $O(k(l+1)^{k+1} h(l,k))$ whether the graph can be oriented as required.

Before we turn to the proofs of these statements we give the following result on the size of the set $\{c(G, X, \omega) | \omega \in \Omega(G)\}$, which will be needed for the proofs below.

Lemma 4.9. *If* $|X| = k+1$ *then*

$$|\{c(G, X, \omega) | \omega \in \Omega(G)\}| = O(h(l,k)).$$

Proof. Recall that $c(G, X, \omega)$ is a 4-tuple $(\overrightarrow{S}, \overleftarrow{S}, M, F)$ where $\overrightarrow{S}$ and $\overleftarrow{S}$ are both subsets of $[l+1]^{k+1}$, M is a $(k+1) \times (k+1)$ matrix with off-diagonal entries from $[l+1]$ and F is a set of elements from $\overrightarrow{S} \times \overleftarrow{S}$. So the number of choices for both $\overrightarrow{S}$ and $\overleftarrow{S}$ is $2^{(l+1)^{k+1}}$, the number of choices for M is $(l+1)^{k(k+1)}$ and the number of choices for F is at most $2^{(l+1)^{2k+2}}$, so the total number of choices for $c(G, X, \omega)$ is at most $(l+1)^{k(k+1)} 2^{[2(l+1)^{k+1} + (l+1)^{2k+2}]}$. Hence the total number of elements of $c(v)$ is at most $(l+1)^{k(k+1)} 2^{[2(l+1)^{k+1} + (l+1)^{2k+2}]}$.

□

We are now able to prove that the algorithm runs in linear time by the following lemmas.

Lemma 4.10. *If v is a leaf of T, Then $c(v)$ can be computed in time $O(k^3 \cdot 2^{\frac{k^2+k}{2}})$.*

Proof. $|W_v| = k+1$. Therefore there can be at most $\frac{k^2+k}{2}$ edges in the subgraph (W_v, E_v). It follows that there are at most $2^{\frac{k^2+k}{2}}$ different orientations of the subgraph (W_v, E_v). We now show that for any orientation ω of E_v, we can compute $c((W_v, E_v), W_v, \omega) = (\overrightarrow{S}, \overleftarrow{S}, M, F)$ in time $O(k^3)$.

For any orientation ω of E_v we have $\overrightarrow{S} = \overleftarrow{S} = \emptyset$. The shortest path matrix can be determined by the Floyd-Warshall algorithm which has running time $O(k^3)$. Then we obtain the truncated shortest path matrix M by replacing each entry of the shortest path matrix that is greater than $l+1$ by $l+1$ in time $O(k^2)$. Since $\overrightarrow{S} = \overleftarrow{S} = \emptyset$ we also have $F = \emptyset$ by definition of F.

□

Lemma 4.11. *Let $v \in V(T)$ and let u be a child of v. Let G^* be the graph $(W_v \cup Y_u, E_u)$. We can compute $\{c(G^*, W_v, \omega) | \omega \in \Omega(G^*)\}$ from $c(u)$ in time*

$$O(k(l+1)^{2k+2}h(l,k)).$$

Proof. Let $\omega \in \Omega(G^*)$. Since G^* is obtained from G_u by merely adding the elements of $W_v \backslash Y_u$ as isolated vertices we have $\omega \in \Omega(G_u)$. Moreover distances in G_u are preserved in G^* and the truncated distance to or from any element of $W_v \backslash Y_u$ is $l+1$.

Claim: $M(G^*, W_v, \omega)$ can be obtained from $M(G_u, W_u, \omega)$ in time $O(k^2)$.

Proof. To obtain $M(G^*, W_v, \omega)$ from $M(G_u, W_u, \omega)$ we replace the off-diagonal entries in rows and columns indexed by elements of $W_u \backslash W_v$, with $l+1$ and change the row and columns labels to elements of $W_v \backslash W_u$. Since $|W_u| = |W_v| = k+1$ the claim follows. □

Claim: $\overrightarrow{S}(G^*, W_v, \omega)$ ($\overleftarrow{S}(G^*, W_v, \omega)$) can be obtained from $\overrightarrow{S}(G_u, W_u, \omega)$ ($\overleftarrow{S}(G_u, W_u, \omega)$) and $M(G_u, W_u, \omega)$ in time $O(k(l+1)^{k+1})$.

Proof. Let $\overrightarrow{d} \in \overrightarrow{S}(G_u, W_u, \omega)$. Then $\overrightarrow{d}$ is the distance vector of one or more vertices in $Y_u \backslash W_u$ to W_u. If we replace the entries of $\overrightarrow{d}$ indexed by members of $W_u \backslash W_v$ with $l+1$ and index them by members of $W_v \backslash W_u$ then we obtain

the distance vector of the same vertices in $Y_u \backslash W_u$ to W_v. Doing this for every $\overrightarrow{d} \in \overrightarrow{S}(G_u, W_u, \omega)$ requires time $O(k(l+1)^{k+1})$ and gives all the distance vectors from vertices in $Y_u \backslash W_u$ to W_v. The other elements of $\overrightarrow{S}(G^*, W_v, \omega)$ are the distance vectors of vertices in $W_u \backslash W_v$ to W_v. These are easily obtained from $M(G_u, W_u, \omega)$ in time $O(k^2)$. So $\overrightarrow{S}(G^*, W_v, \omega)$ can be found from $\overrightarrow{S}(G_u, W_u, \omega)$ and $M(G_u, W_u, \omega)$ in time $O(k(l+1)^{k+1})$. $\overleftarrow{S}(G^*, W_v, \omega)$ can be found in a similar way. □

Claim: $F(G^*, W_v, \omega)$ can be obtained from $\overrightarrow{S}(G_u, W_u, \omega)$, $\overleftarrow{S}(G_u, W_u, \omega)$, $M(G_u, W_u, \omega)$ and $F(G_u, W_u, \omega)$ in time $O(k(l+1)^{2k+2})$.

Proof. Each element of $F(G_u, W_u, \omega)$ corresponds to one or more pairs of vertices $Y_u \backslash W_u$ at distance at least $l+1$ in G_u. These pairs of vertices are still at distance $l+1$ in G^* so $F(G^*, W_v, \omega)$ must contain an element corresponding to them. If $f \in F(G_u, W_u, \omega)$ then all the entries of f indexed by vertices of $W_u \backslash W_v$ must be replaced by $l+1$ in the same way that the entries of elements of $\overrightarrow{S}(G_u, W_u, \omega)$ and $\overleftarrow{S}(G_u, W_u, \omega)$ were modified above. Amending the elements of $F(G_u, W_u, \omega)$ in this way requires time $O(k(l+1)^{2k+2})$ and gives the set of all pairs $(\overrightarrow{d}(G^*, W_v, \omega, x), \overleftarrow{d}(G^*, W_v, \omega, y))$ where $x, y \in Y_u \backslash W_u$ and $d^*(x, y) = l+1$.

It remains to compute the set of all pairs $(\overrightarrow{d}(G^*, W_v, \omega, x), \overleftarrow{d}(G^*, W_v, \omega, y))$ where at least one of x, y is in $W_u \backslash W_v$ and $d^*(x, y) = l+1$. If $x, y \in W_u \backslash W_v$, then it is easy to check if $d^*(x, y) = l+1$ by inspecting $M(G_u, W_u, \omega)$. Furthermore $(\overrightarrow{d}(G^*, W_v, \omega, x), \overleftarrow{d}(G^*, W_v, \omega, y))$ can easily be computed from $M(G_u, W_u, \omega)$. This requires time $O(k^3)$.

Now fix $x \in W_u \backslash W_v$. We show how to compute $\{(\overrightarrow{d}(G^*, W_v, \omega, x), \overleftarrow{d}(G^*, W_v, \omega, y)) | y \in Y_u \backslash W_u \text{ and } d^*(x, y) = l+1\}$. By inspecting the entries indexed by x of all the elements of $\overleftarrow{S}(G_u, W_u, \omega)$ and selecting those elements corresponding to vertices of $Y_u \backslash W_u$ at distance at least $l+1$ from x, it is possible to compute $\{\overleftarrow{d}(G_u, W_u, \omega, y) | y \in Y_u \backslash W_u \text{ and } d^*(x, y) = l+1\}$. The vectors $\overrightarrow{d}(G^*, W_v, \omega, x)$ and $\overleftarrow{d}(G^*, W_v, \omega, y)$ for $y \in Y_u \backslash W_u$ are computed as above. Doing this for all $x \in W_u \backslash W_v$ requires time $O(k^2(l+1)^{k+1})$. The symmetric case where $x \in Y_u \backslash W_u$ and $y \in W_u \backslash W_v$ is very similar.

□

The discussion above show that it is possible to compute $c(G^*, W_v, \omega)$ from $c(G_u, W_u, \omega)$ alone, that is without using any additional information about ω. So to compute $\{c(G^*, W_v, \omega) | \omega \in \Omega(G^*)\}$, we run through the distinct elements of $c(u)$ and carry out the computations described above. This requires time $O(k(l+1)^{2k+2}h(l,k))$. □

Lemma 4.12. *Let v be a non-leaf vertex of T with children v_1 and v_2. Furthermore let G_1^* be the graph $(W_v \cup Y_{v_1}, E_{v_1})$ and G_2^* be the graph $(W_v \cup Y_{v_2}, E_{v_2})$. We can compute $c(v)$ from $\{c(G_1^*, W_v, \omega) | \omega \in \Omega(G_1^*)\}$ and $\{c(G_2^*, W_v, \omega) | \omega \in \Omega(G_2^*)\}$ in time*

$$O(k^2(l+1)^{2k+2}h(l,k)^2).$$

Proof. Let $\omega \in \Omega(G_v)$. The sets E_{v_1} and E_{v_2} are disjoint so we can write $\omega = \omega_1 \cup \omega_2$ where $\omega_1 \in \Omega(G_1^*)$ and $\omega_2 \in \Omega(G_2^*)$. In order to compute $c(v)$, it turns out not to be necessary to consider every $\omega_1 \in \Omega(G_1^*)$ and $\omega_2 \in \Omega(G_2^*)$ separately because we will show that $c(G_v, W_v, \omega)$ depends only on $c(G^*, W_v, \omega_1)$ and $c(G_2^*, W_v, \omega_2)$ and not on the actual choices of ω_1 and ω_2.

Suppose that $c(G_1^*, W_v, \omega_1) = (\vec{S}_1, \overleftarrow{S}_1, M_1, F_1)$ and $c(G_2^*, W_v, \omega_2) = (\vec{S}_2, \overleftarrow{S}_2, M_2, F_2)$. We now show how to compute $c(G_v, W_v, \omega)$ from $(\vec{S}_1, \overleftarrow{S}_1, M_1, F_1)$ and $(\vec{S}_2, \overleftarrow{S}_2, M_2, F_2)$. For notional convenience we will assume that $W_v = \{x_1, \ldots, x_{k+1}\}$.

Claim: $M(G_v, W_v, \omega)$ can be determined from M_1 and M_2 in time $O(k^3)$.

Proof. Let $m_1(i,j)$ and $m_2(i,j)$ denote the entries of M_1 and M_2 respectively in the row indexed by x_i and column indexed by x_j. Let $\hat{m}(i,j) = \min\{m_1(i,j), m_2(i,j)\}$ and let $\hat{M} = (\hat{m}(i,j))$. Then each entry of $\hat{M}$ is the truncated length of the shortest path between two vertices of W_v such that either all of the edges in the path are in E_{v_1} or all the edges in the path are in E_{v_2}. Any shortest path in G_v between two vertices in W_v can be broken up into subpaths beginning and ending in W_v and in which each subpath consists either of edges just from E_{v_1} or of edges just from E_{v_2}. This is because there is no edge between a vertex in $Y_{v_1} \backslash W_v$ and a vertex in $Y_{v_2} \backslash W_v$. Consequently the truncated lengths of the shortest paths between vertices

in W_v can be found by applying the Floyd-Warshall algorithm to $\hat{M}$. This requires time $O(k^3)$. □

Claim: $\overrightarrow{S}(G_v, W_v, \omega)$ ($\overleftarrow{S}(G_v, W_v, \omega)$) can be obtained from $M_1, M_2, \overrightarrow{S}_1$ ($\overleftarrow{S}_1$) and $\overrightarrow{S}_2$ ($\overleftarrow{S}_2$) in time $O(k^2(l+1)^{k+1})$.

Proof. An element of $\overrightarrow{S}_1$ is the distance vector of one or more vertices in $Y_{v_1} \backslash W_v$ to W_v in the graph G_{v_1}. It is possible that these truncated distances are smaller in G_v because shorter paths including edges from E_{v_2} may exist. Let $\overrightarrow{d} \in \overrightarrow{S}_1$ and suppose that $\overrightarrow{d}$ is the distance vector of $y \in Y_{v_1} \backslash W_v$ to W_v. (The identity of y is immaterial in the following argument.)

In (G_v, ω), for each i, the shortest path from y to x_i contains an initial subpath from y to some vertex $x_j \in W_v$, consisting of only edges from E_{v_1} followed by a possibly empty subpath from x_j to x_i. For all i, let $\overrightarrow{d}_i$ denote the truncated distance from y to x_i in (G_1^*, ω_1). The distance $\overrightarrow{d}_i$ is given by one of the entries of $\overrightarrow{d}$. The truncated distance from y to x_i in (G_v, ω) is

$$\min\{\min_j\{\overrightarrow{d}_j + M(j,i)\}, l+1\},$$

where $M(j,i)$ is the entry of $M(G_v, W_v, \omega)$ in the row indexed by x_j and column indexed by x_i. To carry out this computation on $\overrightarrow{d}$ requires time $O(k^2)$. Every member of $\overrightarrow{S}(G_v, W_v, \omega)$ is obtained from a member of either $\overrightarrow{S}_1$ or $\overrightarrow{S}_2$ in this way. The total time to determine $\overrightarrow{S}(G_v, W_v, \omega)$ is therefore $O(k^2(|\overrightarrow{S}_1| + |\overrightarrow{S}_2|)) = O(k^2(l+1)^{k+1})$.

$\overleftarrow{S}(G_v, W_v, \omega)$ can be obtained in an analogous way.

□

Claim: $F(G_v, W_v, \omega)$ can be obtained from $\overleftarrow{S}_1, \overrightarrow{S}_1, \overleftarrow{S}_2, \overrightarrow{S}_2, M_1, M_2, F_1$ and F_2 in time $O(k^2(l+1)^{2k+2})$.

Proof. Each element of $F(G_v, W_v, \omega)$ corresponds to one or more pairs (x, y) of vertices of $Y_v \backslash W_v$ such that $d^*(x, y) = l+1$. Each of x, y belongs to either $Y_{v_1} \backslash W_v$ or $Y_{v_2} \backslash W_v$ and there are two types of pairs to consider depending on whether x, y both belong to the same set or not. (Of course a particular entry of $F(G_v, W_v, \omega)$ may correspond to pairs of more than one type. When

this happens we will try to add it to $F(G_v, W_v, \omega)$ more than once, but this creates no difficulties.)

We first show how to find the set

$$\hat{F}_1 = \{(\overrightarrow{d}(G_v, W_v, \omega, x), \overleftarrow{d}(G_v, W_v, \omega, y)) | d^*(x, y) = l + 1, x, y \in Y_{v_1} \backslash W_v\}.$$

For any element $(\overrightarrow{d}(G_v, W_v, \omega, x), \overleftarrow{d}(G_v, W_v, \omega, y))$ of $\hat{F}_1$ there is a corresponding element $(\overrightarrow{d}(G^*_{v_1}, W_v, \omega, x), \overleftarrow{d}(G^*_{v_1}, W_v, \omega, y)) \in F_1$. However the converse is not true because there may be a short path from x to y in G_v including edges from E_{v_2}. Suppose $(\overrightarrow{d}(G^*_{v_1}, W_v, \omega, x), \overleftarrow{d}(G^*_{v_1}, W_v, \omega, y)) \in F_1$. Let $\overrightarrow{d}_i$ be the truncated length of the shortest path from x to x_i in $G^*_{v_1}$ and let $\overleftarrow{d}_i$ be the truncated length of the shortest path from x_i to y in $G^*_{v_1}$. Any path in G_v from x to y that has length at most l must include at least one vertex from W_v. This path must contain an initial subpath beginning at x and ending in W_v with all edges in E_{v_1} and a terminal subpath beginning in W_v and ending at y with all edges in E_{v_1}. The minimum truncated length of such a path is

$$\min\{\min_{i,j}\{\overrightarrow{d}_i + m(i, j) + \overleftarrow{d}_j\}, l + 1\},$$

where $m(i, j)$ is the entry of $M(G_v, W_v, \omega)$ in the row indexed by x_i and column indexed by x_j. For all i, $\overrightarrow{d}_i$ and $\overleftarrow{d}_i$ are given by the entries of $(\overrightarrow{d}(G^*_1, W_v, \omega_1, x), \overleftarrow{d}(G^*_1, W_v, \omega_1, y))$. Consequently, by running through all elements of F_1 and excluding those corresponding to pairs for which there is now a path of length at most l, the set

$$\{(\overrightarrow{d}(G^*_1, W_v, \omega, x), \overleftarrow{d}(G^*_1, W_v, \omega, y)) | d^*(x, y) = l + 1, x, y \in Y_{v_1} \backslash W_v\}$$

can be computed from F_1, M_1 and M_2 in time $O(k^2(l + 1)^{2k+2})$. From this set it is easy to compute $\hat{F}_1$ by computations similar to those used to form $\overrightarrow{S}(G_v, W_v, \omega)$ and $\overleftarrow{S}(G_v, W_v, \omega)$. If the correspondence between members of $\overrightarrow{S}_1$ and $\overleftarrow{S}_1$, $\overrightarrow{S}_2$ and $\overleftarrow{S}_2$, and $\overrightarrow{S}(G_v, W_v, \omega)$ and $\overleftarrow{S}(G_v, W_v, \omega)$ is stored, then it is not necessary to repeat the computations and so $\hat{F}_1$, can be computed from F_1 in time $O(k^2(l + 1)^{2k+2})$. $\hat{F}_2$ is defined in the obvious way and can

be found in a similar fashion. We now find the set

$$\hat{F}_{12} = \{(\overrightarrow{d}(G_v, W_v, \omega, x), \overleftarrow{d}(G_v, W_v, \omega, y)) | d^*(x,y) = l+1, \\ x \in Y_{v_1} \backslash W_v, y \in Y_{v_2} \backslash W_v\}.$$

Let $\overrightarrow{d}_i$ be the truncated length of the shortest path from x to x_i in $G^*_{v_1}$ but now let $\overleftarrow{d}_i$ be the truncated length of the shortest path from x_i to y in $G^*_{v_2}$. Any path from x to y must pass through a vertex of W_v. So such a path must contain an initial subpath starting at x, ending in W_v and containing only edges of E_{v_1} and a terminal subpath starting in W_v, ending at y and containing only edges of E_{v_2}. The minimum length of such a path is

$$\min\{\min_{i,j}\{\overrightarrow{d}_i + m(i,j) + \overleftarrow{d}_j\}, l+1\}.$$

So by running through pairs of elements of $\overrightarrow{S}_1$ and $\overleftarrow{S}_2$ and selecting those at distance at least $l+1$, the set

$$\{(\overrightarrow{d}(G^*_1, W_v, \omega, x), \overleftarrow{d}(G^*_2, W_v, \omega, y)) | d^*(x,y) = l+1, x \in Y_{v_1} \backslash W_v, y \in Y_{v_2} \backslash W_v\}$$

can be computed from $M_1, M_2, \overrightarrow{S}_1$ and $\overleftarrow{S}_2$ in time $O(k^2(l+1)^{2k+2})$. It is now easy to compute $\hat{F}_{12}$ by computations similar to those used to form $\overrightarrow{S}(G_v, W_v, \omega)$ and $\overleftarrow{S}(G_v, W_v, \omega)$. Finally $\hat{F}_{21}$, defined in the obvious way, can also be found in time $O(k^2(l+1)^{2k+2})$. Now $F(G_v, W_v, \omega) = \hat{F}_1 \cup \hat{F}_2 \cup \hat{F}_{12} \cup \hat{F}_{21}$. □

The computations described above show that it is possible to compute $c(G_v, W_v, \omega)$ from $c(G^*_1, W_v, \omega_1)$ and $c(G^*_2, W_v, \omega_2)$ alone, that is without using any additional information about ω_1 and ω_2. So to compute $c(v)$, we run through every distinct pair of elements from $c(G^*_1, W_v, \omega_1)$ and $c(G^*_2, W_v, \omega_2)$ and carry out the computations described above. This requires time $O(k^2(l+1)^{2k+2}h(l,k)^2)$. □

Note that Lemma 4.11 and 4.12 imply that we can compute $c(v)$ from

$c(v_1)$ and $c(v_2)$ in time

$$O(k^2(l+1)^{2k+2}h(l,k)^2).$$

Lemma 4.13. *We can determine whether G can be oriented with diameter at most l from $c(r)$ in time*

$$O(k(l+1)^{k+1}h(l,k)).$$

Proof. Consider one of the 4-tuples $(\overrightarrow{S}, \overleftarrow{S}, M, F) \in c(r)$ corresponding to a set of orientations of G. Let ω be any element of this set of orientations. If there is a distance vector in $\overrightarrow{S}$ or $\overleftarrow{S}$ having an entry equal to $l+1$ then there is a vertex in $V(G)\backslash W_r$ having distance strictly greater than l to or from a vertex of W_r in (G,ω). Thus in this case any such orientation ω yields (G,ω) with diameter strictly greater than l. Checking the entries of all truncated distance vectors in $\overrightarrow{S}$ and $\overleftarrow{S}$ in this way requires time $O(k(l+1)^{k+1})$.

If there is an entry equal to $l+1$ in M then there is a pair of vertices in W_r having distance strictly greater than l in (G,ω). This can be checked in time $O(k^2)$.

If $F \neq \emptyset$ there is a pair of vertices in $V(G)\backslash W_r$ such that the distance between them in one direction or the other is strictly greater than l. It can be checked in time $O(1)$ whether $F \neq \emptyset$.

On the other hand any 4-tuple $(\overrightarrow{S}, \overleftarrow{S}, M, F) \in c(r)$ for which no member of $\overrightarrow{S}$ or $\overleftarrow{S}$ has an entry equal to $l+1$, no element of M is equal to $l+1$ and F is empty corresponds to at least one orientation ω such that (G,ω) has diameter at most l.

It follows that to check if any orientation ω corresponding to $(\overrightarrow{S}, \overleftarrow{S}, M, F)$ yields (G,ω) with diameter at most l requires time $O(k(l+1)^{k+1} + k^2) = O(k(l+1)^{k+1})$.

Since there are at most $O(h(l,k))$ 4-tuples in $c(r)$, the time to determine from $c(r)$ whether G can be oriented with diameter at most l is $O(k(l+1)^{k+1}h(l,k))$. □

The following theorem is a consequence of Lemmas 4.10, 4.11, 4.12 and 4.13.

Theorem 4.14. *The overall running time of algorithm* DIAMETER is

$$O(nk^2(l+1)^{2k+2}h(l,k)^2).$$

Proof. There are $O(n)$ leaf vertices in the tree-decomposition. For each leaf vertex v it takes time $O(2^{\frac{k^2+k}{2}} \cdot k^3)$ to determine $c(v)$ by Lemma 4.10.

There are $O(n)$ non-leaf vertices in the tree-decomposition. For each non-leaf vertex v with children v_1 and v_2 for which both $c(v_1)$ and $c(v_2)$ are already determined $c(v)$ can be calculated in time $O(k^2(l+1)^{2k+2}h(l,k)^2)$ by Lemmas 4.11 and 4.12. It follows that, given a reduced tree-decomposition, $c(r)$ can be calculated in time $O(n(2^{\frac{k^2+k}{2}} \cdot k^3 + k^2(l+1)^{2k+2}h(l,k)^2))$. Thus by Lemma 4.13 the overall running time to determine whether G can be oriented so that the diameter is at most l is $O(n(2^{\frac{k^2+k}{2}} \cdot k^3 + k^2(l+1)^{2k+2}h(l,k)^2)) = O(nk^2(l+1)^{2k+2}h(l,k)^2)$. □

4.6 Minimising the diameter of any planar graph

We now show how to use the result of the previous section to solve Problem 4.5.

We say that a graph G has an *H-minor* if it is possible to delete vertices and edges, and contract edges from G to obtain a graph isomorphic to H. Since we are only working with simple graphs, we delete any loops or parallel edges that are formed after a contraction. If H is obtained from G by deleting a set A of edges and contracting a set B of edges then in whatever order these deletions and contractions are done, we always obtain H.

Lemma 4.15. *Any planar graph with a $(2l+1) \times (2l+1)$-grid-minor has diameter at least l.*

Proof. Suppose that G is a planar graph having a $(2l+1)\times(2l+1)$-grid-minor. We may assume that G is connected because otherwise $\mathrm{diam}(G) = \infty$. So a $(2l+1) \times (2l+1)$-grid may be obtained from G by a series of contractions and deletions of edges. Without loss of generality we can assume that first

all the contractions are done and afterwards all the deletions. For any graph G' and for any edge $e \in E(G')$ the following inequality holds:

$$\mathrm{diam}(G') \geq \mathrm{diam}(G'/e).$$

Let K be the graph obtained from G after all the contractions of edges. Then the above inequality yields

$$\mathrm{diam}(G) \geq \mathrm{diam}(K).$$

For $l \geq 1$, the graph obtained by contracting one of the edges adjacent to each vertex of degree two in the $(2l+1) \times (2l+1)$-grid is 3-connected. A famous result of Whitney [80] implies that it has a unique embedding (apart from mirror image) in the plane once the outside face has been selected. Subdividing edges has no effect on an embedding in the plane so the grid also has a unique embedding up to selection of the outside face. Imagine that we have embedded the graph in the plane in the standard way so that every face except the outside face has four edges.

K is a simple graph of which the $(2l+1) \times (2l+1)$-grid is a subgraph. The only edges of K which do not join two vertices of the outside face and which are not in the grid join opposite corners of an internal face of the grid. Consequently $\mathrm{diam}(K) \geq l$ and so $\mathrm{diam}(G) \geq l$.

□

The following result which we will need for the proof of our main statement, is from [74].

Theorem 4.16. *Any planar graph with no $g \times g$-grid-minor has tree-width at most $6g - 5$.*

We are now ready to prove the main statement of this chapter which is the following result on the complexity of deciding whether a graph can be oriented in such a way that its diameter is at most l.

Theorem 4.17. *For every l, Problem 4.5 is solvable in time $O(c_l n)$, where c_l is a constant depending on l.*

Proof. By Theorem 4.7, there is an algorithm running in time $O(f(12l+13)n)$ that determines whether the tree-width of G is at most $12l + 13$.

First let us consider the case that the tree-width is at most $12l + 13$. Then the algorithm also finds a tree-decomposition of G with tree-width at most $12l + 13$. By Lemma 4.8 this tree-decomposition can be modified in time $O(f(12l + 13)n)$ so that it is a reduced tree-decomposition. Then by Theorem 4.14, the algorithm DIAMETER described in Section 4.5 can determine in time $O(n(12l + 13)^2(l + 1)^{2(12l+13)+2}h(l, 12l + 13)^2)$, whether it is possible to orient the edges of G so that the diameter is at most l.

Now we consider the case that the tree-width of G is at least $12l + 14$. Then it follows from Theorem 4.16, that G has a $(2l+3)\times(2l+3)$-grid-minor. By Lemma 4.15, G must have diameter at least $l+1$. It follows that it is not possible to orient G so that the diameter is at most l. □

4.7 Conclusion

We have shown that Problem 4.1 which is NP-complete for arbitrary graphs becomes polynomial time for planar graphs even if the constant two is replaced by any larger constant. It remains to determine the complexity of minimising the Wiener index for planar graphs. Furthermore it would be interesting to try to find a more efficient algorithm for our main problem, not depending on graph minor theory, and also to determine the complexity when l is part of the input.

Chapter 5

k-$L(2,1)$-Labelling for Planar Graphs is NP-Complete for $k \geq 4$

In this chapter, which is joint work with Frédéric Havet, INRIA in Nice, and Steven Noble, Brunel University, we turn to a topic which is very different from the problems that we have already discussed in this thesis. The Frequency Assignment Problem requires the assignment of frequencies to radio transmitters in a broadcasting network with the aim of avoiding undesired interference and minimising bandwidth. One of the longstanding graph theoretical models of this problem is the notion of distance constrained labelling of graphs. An $L(2,1)$-*labelling* of a graph G is a mapping from the vertex set of G into the nonnegative integers such that the labels assigned to adjacent vertices differ by at least 2, and labels assigned to vertices at distance 2 are different. The *span* of such a labelling is the maximum label used. In this model, the vertices of G represent the transmitters and the edges of G express which pairs of transmitters are too close to each other so that undesired interference may occur, even if the frequencies assigned to them differ by 1. This model was introduced by Roberts [67] and since then the concept has been intensively studied (see the survey of Yeh [82]).

In their seminal paper, Griggs and Yeh [36] proved that determining the minimum span of a graph G, denoted $\lambda_{2,1}(G)$, is an NP-hard problem.

Fiala *et al.* [30] proved that deciding $\lambda_{2,1}(G) \leq k$ is NP-complete for every fixed $k \geq 4$ and later Havet and Thomassé [40] proved that for any $k \geq 4$, it remains NP-complete when restricted to bipartite graphs (and even a restricted family of bipartite graphs, i.e *incidence graphs* or *first division of graphs*). When the span k is part of the input, the problem is nontrivial for trees but a polynomial time algorithm based on bipartite matching was presented in [15]. The problem is still solvable in polynomial time if the input graph is outerplanar [49, 50].

Moreover, somewhat surprisingly, the problem becomes NP-complete for series-parallel graphs [29], and thus the $L(2,1)$-labelling problem belongs to a handful of problems known to separate graphs of tree-width 1 and 2 by P/NP-completeness dichotomy.

In this chapter we consider the following problem.

Problem 5.1 (Planar k-$L(2,1)$-Labelling)**.**
Let $k \geq 4$ be fixed.
Instance: A planar graph G.
Question: Is there an $L(2,1)$-labelling with span k?

Bodlaender *et al.* [7] showed that this problem is NP-complete if we require $k \geq 8$ and k even. In the survey paper [14], it is suggested that the problem is NP-complete for all $k \geq 8$ due to [33]. However this does not seem to be the case. In [33] there is a proof showing that the corresponding problem where k is specified as part of the input is NP-complete. This proof shows that the problem is NP-complete for certain fixed values of k. However it is far from clear for which values of k this is true. The same authors also show in [34] that the problem is NP-complete for $k = 8$.

In this chapter we first prove that Planar Cubic Two-Colourable Perfect Matching, which we define in the next section, is NP-Complete. This result was first stated by Schaefer [76] but without proof. In the second part of this chapter we use this result in order to show that Problem 5.1 is NP-complete.

5.1 Preliminary results

The starting problem for our reductions is Not-All-Equal 3SAT, which is defined as follows [76].

Definition 5.2 (Not-All-Equal 3SAT)**.**
Instance: A set of clauses each having three literals.
Question: Can the literals be assigned value true or false so that each clause has at least one true and at least one false literal?

In [76], it is shown that this problem is NP-complete.

Our reduction involves an intermediate problem concerning a special form of two-colouring. In this section we define the intermediate problem and show that it is NP-complete. When $k = 4$ or $k = 5$, the final stage of our reduction is similar to the reduction in [30]. However we cannot use induction for higher values of k in contrast with the situation in [30] and the problem from which the reduction starts in [30] is not known to be NP-complete for planar graphs. So considerably more work is required.

The following problem is also discussed in [76].

Problem 5.3 (Two-Colourable Perfect Matching)**.**
Instance: A graph G.
Question: Is there a colouring of the vertices of G with colours black and white in which every vertex has exactly one neighbour of the same colour?

In [76] it was shown that Two-Colourable Perfect Matching is NP-complete. We are more interested in the case where the input is restricted to being a planar cubic graph. We call this variant, Planar Cubic Two-Colourable Perfect Matching defined formally as follows [76].

Problem 5.4 (Planar Cubic Two-Colourable Perfect Matching)**.**
Instance: A planar cubic graph G.
Question: Is there a colouring of the vertices of G with colours black and white in which every vertex has exactly one neighbour of the same colour?

Schaefer [76] states that this problem is NP-complete but does not give the details of the proof. We call a colouring as required in Problem 5.4 a *two-coloured perfect matching*. This section is devoted to the proof of this result, using a reduction from Not-All-Equal 3SAT [76]. As far as we know, no proof of this has ever been published.

We say that a colouring of the vertices of a graph with colours black and white is an *almost two-coloured perfect matching* if every vertex of degree at least two is adjacent to exactly one vertex of the same colour. We say an edge is *monochromatic* if both endpoints have the same colour and *dichromatic* if its endpoints have different colours.

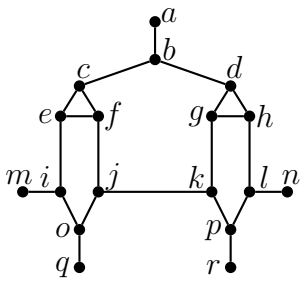

Figure 5.1: Planar graph H.

Let H be the planar graph (see Fig. 5.1) with

$$V(H) = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r\}$$

and

$$E(H) = \{ab, bc, bd, ce, cf, dg, dh, ef, gh, ei, fj, \\ gk, hl, im, jk, ln, io, jo, oq, kp, pl, pr\}.$$

H plays a key role in showing that Problem 5.4 is NP-complete. We need the following lemma.

Lemma 5.5. *Any almost two-coloured perfect matching of H has the following properties.*

- *Exactly one of the edges ab, mi, ln is monochromatic.*

- *Vertices b, i, l receive the same colour.*
- *Vertices o, p, q, r receive the other colour to b, i, l.*

Proof. Consider the triangles on the vertices c, e, f and g, h, d. In order to obtain an almost two-coloured perfect matching exactly one of the edges ce, ef and cf must be monochromatic. The same is true for the triangle on the vertices g, h, d. Now consider the subgraph of H induced by the vertices $c, e, f, i, j, m, k, o, q$. In Fig. 5.2 three of the six almost two-coloured perfect matchings of this subgraph are depicted with monochromatic edges shown by heavy lines. The other three two-coloured perfect matchings are obtained by interchanging the colours. It follows that oq and pr must be monochromatic.

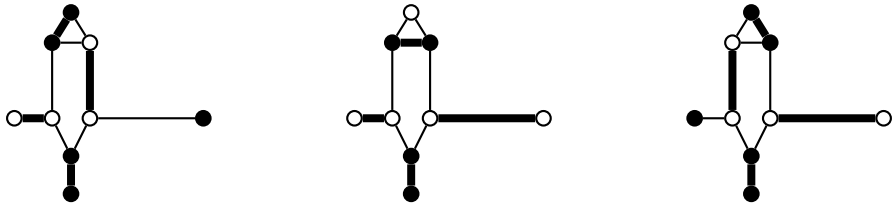

Figure 5.2: Three almost two-coloured perfect matchings of the subgraph of H induced by the vertices $c, e, f, m, i, j, k, o, q$.

By symmetry the same applies to the subgraph of H induced by the vertices $d, g, h, j, k, l, n, p, r$. Considering which pairs of these almost two-coloured perfect matchings are compatible and extend to an almost two-coloured perfect matching of H shows that there are only six possibilities. In Fig. 5.3 three possible almost two-coloured perfect matchings are depicted. The only other possible almost two-coloured perfect matchings are obtained by interchanging the two colours. Clearly these all have the properties described in the lemma.

□

We define what we call the *clause gadget graph* K as follows, see Fig. 5.4. Take three copies of H, namely H_1, H_2 and H_3. We label the vertices by adding the subscript $i \in \{1, 2, 3\}$ to the corresponding label of H. Now identify a_1, a_2, a_3 into a single vertex a, remove vertices $m_1, m_2, m_3, n_1, n_2, n_3$

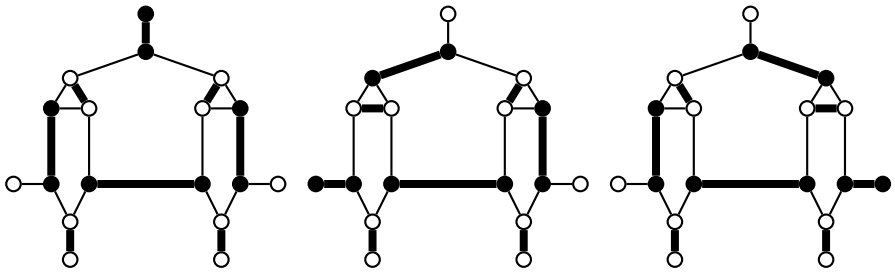

Figure 5.3: Almost two-coloured perfect matchings of H.

and their incident edges and replace them with edges l_1i_2, l_2i_3, l_3i_1. Notice that K is planar and every vertex has degree three, except for q_1, q_2, q_3 and r_1, r_2, r_3.

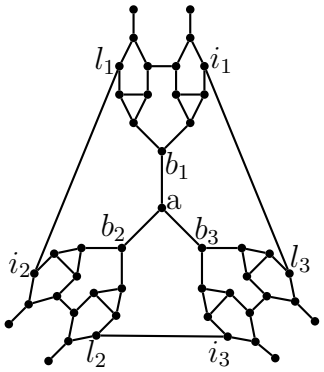

Figure 5.4: Planar clause gadget K.

Lemma 5.6. *A two-colouring of* $\bigcup_{t=1}^{3}\{o_t, q_t, p_t, r_t\} \cup \{a\}$ *may be extended to an almost two-coloured perfect matching of* K *if and only if*

- *For each* $t = 1, 2, 3$, o_tq_t, p_tr_t *are monochromatic and* o_t, p_t, q_t, r_t *all receive the same colour.*
- *For exactly two values of* $t = 1, 2, 3$, *the vertices* o_t, p_t, q_t, r_t *receive the same colour as* a.

Proof. We first show that any almost two-coloured perfect matching of K must have the two properties in the lemma.

The first property is an immediate consequence of Lemma 5.5.

To show that the second property holds, recall that exactly one neighbour of a must receive the same colour as a. Let b_{t_1} for $1 \leq t_1 \leq 3$ be this neighbour. Then from Lemma 5.5 we know that b_{t_1} must have the opposite colour to $o_{t_1}, p_{t_1}, q_{t_1}, r_{t_1}$. Since the other neighbours of a, namely b_{t_2} and b_{t_3} for $t_2, t_3 \in \{1,2,3\}\backslash\{t_1\}$, receive the opposite colour to a, the vertices $o_{t_2}, p_{t_2}, q_{t_2}, r_{t_2}, o_{t_3}, p_{t_3}, q_{t_3}, r_{t_3}$ must receive the same colour as a.

Now we show that any two-colouring of $\bigcup_{t=1}^{3}\{o_t, q_t, p_t, r_t, b_t\} \cup \{a\}$ satisfying the conditions of the lemma may be extended to an almost two-coloured perfect matching of K. Suppose without loss of generality, a is coloured black and o_1, p_1, q_1, r_1 are coloured white. Then colour l_1, i_1 black and l_2, i_2, l_3, i_3 white. This colouring may be extended to an almost two-coloured perfect matching using the colourings of Fig. 5.3 and the colourings obtained from those in Fig. 5.3 by interchanging the colours. □

We now move a step towards the main result of this section with the following proposition.

Proposition 5.7. *Problem 5.3 is NP-complete if the input is restricted to cubic graphs.*

Proof. Given an instance of Not-All-Equal 3SAT with clauses $C_1, \ldots, C_m$, construct a graph as follows. For every clause C take a copy of the clause gadget graph K and do the following. Suppose without loss of generality C has literals x_1, x_2, x_3. Label the two vertices of degree one of the subgraph H_i of $K(C)$ and their neighbours in H_i with x_i.

Now for each literal x do the following. Suppose that literal x appears in clauses $C_{i_1}, \ldots, C_{i_k}$. (If x appears twice or three times in a clause C_r then add C_r twice or three times to this list.) For every $j = 1, \ldots, k-1$ remove either one of the vertices of degree one labelled x from $K(C_{i_j})$ and from $K(C_{i_{j+1}})$ leaving two half-edges. Now identify these two half edges to form an edge joining $K(C_{i_j})$ and $K(C_{i_{j+1}})$. Finally do the same thing with the remaining two edges labelled x in C_{i_k} and C_{i_1}. We call the graph obtained G.

Now suppose that there is a solution S of the instance of Not-All-Equal 3SAT given. For all literals x in Not-All-Equal 3SAT, colour all vertices in G that are labelled x with colour white if $S(x)$ is true and black if $S(x)$ is false. We now show that this colouring can be extended to a two-coloured perfect matching of G. First note that all edges joining copies of K are monochromatic since their endpoints are labelled with the same literal. In the next step colour the vertex a in each copy of K so that it has the same colour as the vertices o_t, p_t for exactly two values of $t = 1, 2, 3$. This is possible because for $t = 1, 2, 3$, o_t, p_t are labelled with literals x_1, x_2, x_3 which cannot have all the same value as they belong to one clause. By Lemma 5.6 we can extend the colouring of each copy of K to an almost two-coloured perfect matching of K which yields a two-coloured perfect matching of G.

Now suppose there is a two-coloured perfect matching of G. By Lemma 5.6 the edges joining the copies of K must be monochromatic. All vertices in G labelled with the same literal therefore must have the same colour and in each copy of K, for exactly two values of $t = 1, 2, 3$, the vertices o_t, p_t receive the same colour. It follows that if we assign to each literal x the value true if it is the label of white vertices and false it is the label of black vertices then we obtain a solution to Not-All-Equal 3SAT. □

We call the edges joining copies of K *identifying edges*.

In order to prove that Problem 5.4 is NP-complete we still need to deal with edges that cross. For this reason we define the *uncrossing gadget* U as follows, see Fig. 5.5.

$$V(U) = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, z_1, z_2, z_3, z_4\}$$

and

$$\begin{aligned} E(U) = \{&ab, bc, bg, ce, cd, ef, df, dh, gh, hi, gj, ij, fk, kl, km, ln, mn, mp, \\ &io, op, or, pq, rq, ev, lv, wv, js, rt, st, sz_1, tz_1, z_1z_2, nu, qx, ux, \\ &uz_3, xz_3, z_3z_4\}. \end{aligned}$$

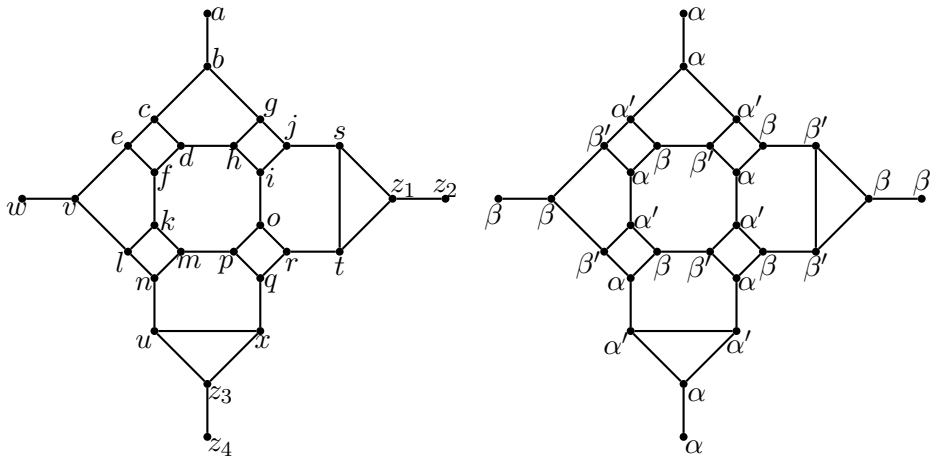

Figure 5.5: Uncrossing gadget U and its possible almost two-coloured perfect matchings where $\alpha, \beta \in \{\text{black}, \text{white}\}$ and α', β' are the opposite colours to α and β, respectively.

Lemma 5.8. *There exists an almost two-coloured perfect matching of U if and only if w, v, z_1, z_2 have the same colour and a, b, z_3, z_4 have the same colour.*

Proof. Consider the four-cycle on the vertices c, e, f, d in U. In a two-coloured perfect matching exactly two of the vertices c, e, f, d must receive the colour black and the other two must receive the colour white. There are two different ways of colouring them. The first way is that exactly two of the edges in the four-cycle are monochromatic, namely ce and df, or cd and ef. Then none of the edges cb, dh, ev, fk can be monochromatic. The other way is that none of the edges in the four-cycle is monochromatic and all of the edges cb, dh, ev, fk are monochromatic. The analogous thing is true for any four-cycle in U.

We now prove that in any almost two-coloured perfect matching the edges ab, wv, z_1z_2 and z_3z_4 are monochromatic. Suppose ab is dichromatic. Then precisely one of bc and bg must be monochromatic. Without loss of generality assume bc is monochromatic. Then the four-cycle on c, e, f, d cannot have a monochromatic edge. It follows dh must be monochromatic. But then bg must be monochromatic which is a contradiction. Thus ab and by symmetry wv must be monochromatic. Now suppose z_1z_2 is dichromatic. It follows that precisely one of sz_1 and tz_1 is monochromatic. Without loss of gen-

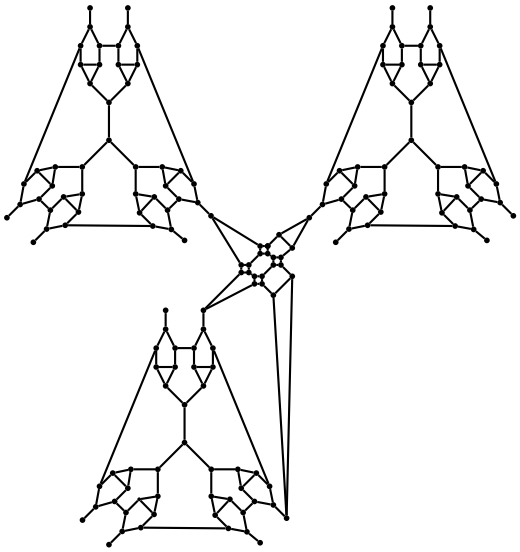

Figure 5.6: Sketch of the modification from the Not-All-Equal 3SAT problem to a planar cubic graph .

erality assume tz_1 is monochromatic. Then js must be monochromatic. It follows that io must be monochromatic and so must rt. This is not possible. Thus z_1z_2 and due to symmetry z_3z_4 must be monochromatic. Hence each of the four-cycles $(c, d, f, e), (g, h, i, j), (k, l, n, m), (o, p, q, r)$ contains exactly two monochromatic edges. So vertices that are opposite of each other in these four-cycles receive opposite colours. It is now easy to see that the only possible colourings are as shown in Fig. 5.5 where $\alpha, \beta \in \{\text{black}, \text{white}\}$ and α' denotes the opposite colour to α and β' denotes the opposite colour to β. The result then follows. □

Our next result was originally stated without proof in [76]. To the best of our knowledge no proof of this result has ever been published. In [76] a reduction from Not-All-Equal 3SAT is used to show that the version of Problem 5.4 where the input may be any graph is NP-complete. We use a reduction from Not-All-Equal 3SAT in a similar way but with much more complicated gadgets.

Theorem 5.9. *Problem 5.4 is NP-complete.*

Proof. Given an instance of Not-All-Equal 3SAT, construct the graph G as

in the proof of Proposition 5.7. This graph can be drawn in the plane so that the only edges that cross are the identifying edges, each pair of identifying edges crosses at most once and at most two edges cross at any point.

Now we replace the crossings one by one by replacing a pair of crossing edges by the uncrossing gadget, see Fig. 5.6. Suppose γ and δ are two edges that cross. We delete γ and δ and replace them with a copy of the uncrossing gadget attaching w and z_2 to the endpoints of γ, and a and z_4 to the endpoints of δ. We will also call the four pendant edges wv, ab, z_1z_2, z_3z_4 in the uncrossing graph identifying edges. After each replacement we can draw the graph so that only identifying edges cross and such that there is one fewer crossing. We continue until there are no more crossing edges. The final graph can be constructed in polynomial time and is planar and cubic. Each original identifying edge in G now corresponds to one or more identifying edges with each consecutive pair being on opposite sides of a copy of the uncrossing gadget. Lemma 5.8 shows that in a two-coloured perfect matching all of these edges must be monochromatic and all the endpoints of these edges have the same colour.

Now the argument in Proposition 5.7 shows that the final graph has a two-coloured perfect matching if and only if the instance of Not-All-Equal 3SAT is satisfiable. □

5.2 k-$L(2,1)$-labelling for $k \geq 4$ fixed

Let G be a planar cubic graph. In order to establish our main result we will reduce Planar Cubic Two-Colourable Perfect Matching to Planar k-$L(2,1)$-Labelling for planar graphs for each $k \geq 4$. From any planar cubic graph G forming an instance of Planar Cubic Two-Colourable Perfect Matching, we construct a graph K. As we see in the next section, the basic form of K does not depend on k but is formed by constructing an auxiliary graph H and then replacing each edge of H by a gadget which does depend on k. In this section we define these gadgets and analyse certain $L(2,1)$-labellings of them. Each gadget has two distinguished vertices, which will always be labelled u and v, corresponding to the endpoints of the edge that is replaced

in the auxiliary graph defined in the next section. These two vertices have degree $k-1$ in K. Any vertex of degree $k-1$ must receive either label 0 or k in a k-$L(2,1)$-labelling because these are the only possible labels for which there are $k-1$ labels remaining to label the neighbours of that vertex, so we will analyse the k-$L(2,1)$-labellings of these gadgets in which u, v receive label 0 or k.

5.2.1 $\lambda_{2,1}(G) = 4$

In this subsection the gadget G_4 is simply a path of length three. More precisely the gadget G_4 is given by

$$V(G_4) = \{u, a_u, a_v, v\}$$

and

$$E(G_4) = \{ua_u, a_u a_v, a_v v\}.$$

This gadget is used in [30], from where we get the following lemma.

Lemma 5.10. *There is a 4-$L(2,1)$-labelling L of G_4 with $L(u), L(v) \in \{0,4\}$ if and only if the following conditions are satisfied.*

1. *If $(L(u), L(v)) = (0,0)$, then $(L(a_u), L(a_v)) \in \{(2,4),(4,2)\}$.*
2. *If $(L(u), L(v)) = (4,4)$, then $(L(a_u), L(a_v)) \in \{(0,2),(2,0)\}$.*
3. *If $(L(u), L(v)) = (4,0)$, then $(L(a_u), L(a_v)) = (1,3)$.*
4. *If $(L(u), L(v)) = (0,4)$, then $(L(a_u), L(a_v)) = (3,1)$.*

5.2.2 $\lambda_{2,1}(G) = 5$

Let G_5 be the graph depicted in Fig. 5.7.

$$V(G_5) = \{u, a_u, a_v, v, b_u, b_v, c, d, e_1, e_2, e_3, f, g_1, g_2\}$$

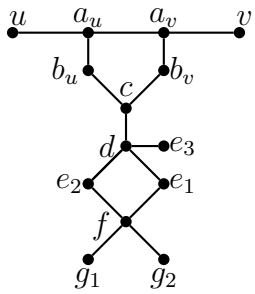

Figure 5.7: The edge gadget G_5.

and

$$E(G_5) = \{ua_u, a_ua_v, a_vv, a_ub_u, a_vb_v, b_uc, b_vc, cd, de_1, de_2, de_3, e_1f, e_2f, fg_1, fg_2\}.$$

Lemma 5.11. *There is a* 5*-*$L(2,1)$*-labelling* L *of* G_5 *with* $L(u), L(v) \in \{0, 5\}$ *if and only if the following conditions are satisfied.*

1. *If* $(L(u), L(v)) = (0, 0)$*, then* $(L(a_u), L(a_v)) \in \{(2, 5), (5, 2), (3, 5), (5, 3)\}$.
2. *If* $(L(u), L(v)) = (5, 5)$*, then* $(L(a_u), L(a_v)) \in \{(3, 0), (0, 3), (2, 0), (0, 2)\}$.
3. *If* $(L(u), L(v)) = (0, 5)$*, then* $(L(a_u), L(a_v)) = (4, 1)$.
4. *If* $(L(u), L(v)) = (5, 0)$*, then* $(L(a_u), L(a_v)) = (1, 4)$.

Proof.

1. By the definition of $L(2,1)$-labelling, $L(a_u)$ and $L(a_v)$ are both in $\{2, 3, 4, 5\}$. As $|L(a_u) - L(a_v)| \geq 2$, $(L(a_u), L(a_v)) \in \{(2, 4), (4, 2), (2, 5), (5, 2), (3, 5), (5, 3)\}$.

   Suppose for a contradiction that $(L(a_u), L(a_v)) \in \{(2, 4), (4, 2)\}$. By symmetry, we may assume that $(L(a_u), L(a_v)) = (2, 4)$. Then $L(b_u) = 5$ and $L(b_v) = 1$. The vertices d and f have degree four and thus must receive labels from $\{0, 5\}$. Because $d(b_u, d) = 2$ we must have $L(d) = 0$ and because $d(d, f) = 2$ we must have $L(f) = 5$. This implies that $\{L(e_1), L(e_2)\} = \{2, 3\}$. But then vertex c cannot be labelled, giving a contradiction. An $L(2,1)$-labelling is obtained if $(L(a_u), L(a_v)) = (5, 2)$

and $L(b_u) = 3$, $L(b_v) = 4$, $L(c) = 0$, $L(d) = 5$, $L(e_3) = 1$, $L(e_2) = 2$, $L(e_1) = 3$, $L(f) = 0$ $L(g_1) = 5$ and $L(g_2) = 4$ or if $(L(a_u), L(a_v)) = (5, 3)$ and $L(b_u) = 2$, $L(b_v) = 1$, $L(c) = 4$, $L(d) = 0$, $L(e_3) = 5$, $L(e_2) = 2$, $L(e_1) = 3$, $L(f) = 5$ $L(g_1) = 0$ and $L(g_2) = 1$. The other cases follow by symmetry.

2. Analogous to (i).

3. By the definition of $L(2,1)$-labelling, $L(a_u) \in \{2, 3, 4\}$ and $L(a_v) \in \{1, 2, 3\}$. As $|L(a_u) - L(a_v)| \geq 2$, $(L(a_u), L(a_v)) \in \{(4, 1), (4, 2), (3, 1)\}$. Suppose for a contradiction that $(L(a_u), L(a_v)) \neq (4, 1)$. By the label symmetry $x \mapsto 5 - x$, we may assume that $(L(a_u), L(a_v)) = (4, 2)$. Hence $L(b_u) = 1$ and $L(b_v) = 0$. Now the vertices d and f have degree four and thus $L(d) = 0, L(f) = 5$ or $L(d) = 5, L(f) = 0$. As $d(b_v, d) = 2$, $L(d) = 5$ and as $d(d, f) = 2$, $L(f) = 0$. Now $\{L(e_1), L(e_2)\} = \{2, 3\}$. But then vertex c cannot be labelled, a contradiction. An $L(2,1)$-labelling is obtained if $(L(a_u), L(a_v)) = (4, 1)$ and $L(b_u) = 2$, $L(b_v) = 3$, $L(c) = 0$, $L(d) = 5$, $L(e_3) = 1$, $L(e_2) = 2$, $L(e_1) = 3$, $L(f) = 0$ $L(g_1) = 5$ and $L(g_2) = 4$.

4. Analogous to (iii).

□

5.2.3 $\lambda_{2,1}(G) \geq 6$

In this subsection we introduce for any $k \geq 6$ the gadget G_k and consider some of its $L(2,1)$-labellings. The gadgets G_6 and G_7 are depicted in Fig. 5.8 and in Fig. 5.9, respectively.

Let H' be defined as follows, see Fig. 5.10.

$$V(H') = \{c, d, e, f_1, \ldots, f_{k-3}, g, h, i\}$$

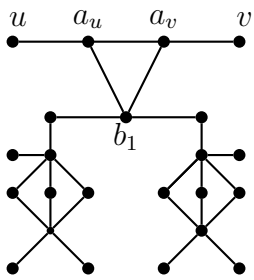

Figure 5.8: The edge gadget G_6.

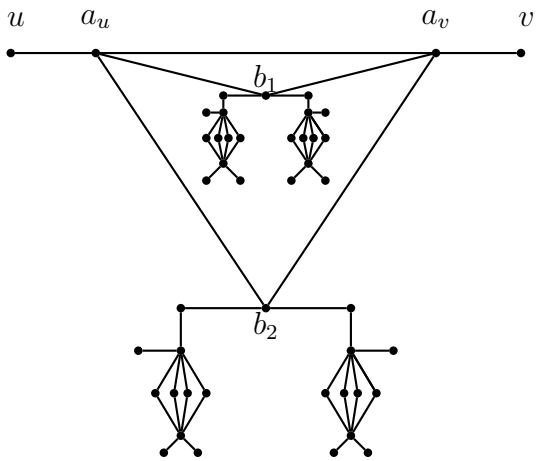

Figure 5.9: The edge gadget G_7.

and

$$E(H') = \{cd, de, hg, gi, gf_1, \ldots, gf_{k-3}, df_1, \ldots, df_{k-3}\}.$$

Lemma 5.12. *For any $k \geq 6$ the graph H' is planar and there exists an $L(2,1)$-labelling L of H' with span k if and only if*

$$(L(c), L(d)) \in \{(0,k), (1,k), (k-1,0), (k,0)\}.$$

Proof. As seen from Fig. 5.10, H' is planar for $k = 6$. For any higher k we need to connect $k-6$ paths of length two at d and g to the graph H' in

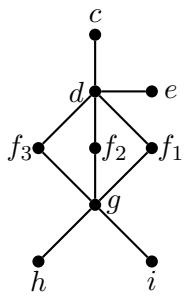

Figure 5.10: The graph H' for $k = 6$.

Fig. 5.10. So H' is planar for any $k \geq 6$.

If L is a k-$L(2,1)$-labelling of H' then $\{L(g), L(d)\} = \{0, k\}$ because g and d have degree $k-1$ and $d(g,d) = 2$. It follows that $\{f_1, \ldots, f_{k-3}\} = \{2, \ldots, k-2\}$. Suppose that $L(d) = 0$ and $L(g) = k$. Then $\{L(h), L(i)\} = \{1, 0\}$, $\{L(c), L(e)\} = \{k-1, k\}$. Similarly if $L(d) = k$ and $L(g) = 0$, then $\{L(h), L(i)\} = \{k-1, k\}$, $\{L(c), L(e)\} = \{0, 1\}$. □

We now define the graph G_k for $k \geq 6$. Take a path of length three with vertices u, u_a, a_v, v and edges ua_u, a_ua_v, a_vv. Add vertices $b_1, \ldots, b_{k-5}$, with each joined to a_u and a_v. Now for each $i = 1, \ldots, k-5$, add two copies of H' with the vertex labelled c in each copy joined to b_i. So each of $b_1, \ldots, b_{k-5}$ has degree four. To refer to the vertices in the two copies of H' attached to b_i, we add a subscript of (l, i) to the name of the vertices in one copy of H' and (r, i) to the name of the vertices in the other copy of H'. Notice that G_k is planar.

Lemma 5.13. *There exists a k-$L(2,1)$-labelling L of G_k with $L(u) = L(v) = 0$ if and only if $(L(a_v), L(a_u)) \in \{(2,k), (k,2), (k-2,k), (k,k-2)\}$.*

Proof. First notice that we need $k-5$ different colours to colour the vertices $b_1, \ldots, b_{k-5}$ as they are all at distance two from each other. We first show that there exists a k-$L(2,1)$-labelling L of G_k with $L(u) = L(v) = 0$ and $(L(a_u), L(a_v)) \in \{(2,k), (k,2), (k-2,k), (k,k-2)\}$. The first case is when $L(a_u) = 2$ and $L(a_v) = k$. Take $L(b_j) = j+3$ for $j = 1, \ldots, k-5$. A k-$L(2,1)$-labelling is obtained by setting $L(d_{l,j}) = L(d_{r,j}) = k$ and $L(c_{l,j}) = 0$

and $L(c_{r,j}) = 1$ for $1 \leq j \leq k-5$ and then using Lemma 5.12 to give a valid labelling of the rest of the graph.

A similar argument shows that we may take $(L(a_u), L(a_v)) = (k, 2)$.

The second case is $L(a_u) = k-2$ and $L(a_v) = k$. Take $L(b_j) = j+1$ for $j = 1, \ldots, k-5$. A k-$L(2,1)$-labelling is obtained by setting $L(d_{l,j}) = 0$, $L(d_{r,j}) = k$ and $L(c_{l,j}) = k-1$ and $L(c_{r,j}) = 0$ for $1 \leq j \leq k-5$ and then using Lemma 5.12 to give a valid labelling of the rest of the graph.

A similar argument shows that we may take $(L(a_u), L(a_v)) = (k, k-2)$.

Next we show that there is no k-$L(2,1)$-labelling L of G_k with $L(u) = L(v) = 0$ and $(L(a_v), L(a_u)) \notin \{(2,k), (k,2), (k-2,k), (k,k-2)\}$. Assume without loss of generality that $L(a_u) < L(a_v)$. Suppose first that $3 \leq L(a_u) \leq k-3$ and $L(a_v) = k$. Note that for any j by considering the proximity of b_j to u and a_v, we see that b_j cannot be labelled with $0, k-1$ or k. Furthermore we cannot have $b_j = 1$ because then $\{L(c_{l,j}), L(c_{r,j})\} = \{k, k-1\}$. But they are both at distance two from a_v which is labelled k, so this is not possible. So $b_1, \ldots, b_{k-5}$ must receive distinct labels from $\{2, \ldots, k-2\} \backslash \{L(a_u)-1, L(a_u), L(a_u)+1\}$, but this only gives $k-6$ labels which is not enough.

Now suppose $L(a_v) \neq k$ and $2 \leq L(a_u) \leq L(a_v) - 2 \leq k-3$. Note that for any j, b_i cannot be labelled with 0. Furthermore b_j cannot be labelled k as then $\{L(c_{l,j}), L(c_{r,j})\} = \{0, 1\}$ and $L(d_{l,j}) = L(d_{r,j}) = k$ but this is invalid. So $b_1, \ldots, b_{k-5}$ must receive distinct labels from $\{1, \ldots, k-1\} \backslash \{L(a_u)-1, L(a_u), L(a_u)+1, L(a_v)-1, L(a_v), L(a_v)+1\}$. This is only possible if $L(a_u) = k-3$ and $L(a_v) = k-1$. Then $\{L(b_1), \ldots, L(b_{k-5})\} = \{1, \ldots, k-5\}$. However if $L(b_j) = 1$ then $\{L(c_{l,j}), L(c_{r,j})\} = \{k-1, k\}$. But this is invalid as $L(a_v) = k-1$. □

Analogously we obtain the following lemma.

Lemma 5.14. *There exists a k-$L(2,1)$-labelling L of G_k with $L(u) = L(v) = k$ if and only if $(L(a_v), L(a_u)) \in \{(2,0), (0,2), (k-2,0), (0,k-2)\}$.*

Lemma 5.15. *There exists a k-$L(2,1)$-labelling L of G_k with $L(u) = k$ and $L(v) = 0$ if and only if $(L(a_v), L(a_u)) = (k-1, 1)$.*

Proof. Let $l_1 = \min\{L(a_u), L(a_v)\}$ and $l_2 = \max\{L(a_u), L(a_v)\}$. The vertices $b_1, \ldots, b_{k-5}$ must be labelled with distinct labels from $S = \{1, \ldots, k-1\} \backslash \{l_1 - 1, l_1, l_1 + 1, l_2 - 1, l_2, l_2 + 1\}$. The only way that this set can contain $k-5$ elements is if $(l_1, l_2) = (1, k-1)$, $(l_1, l_2) = (1, 3)$ or $(l_1, l_2) = (k-3, k-1)$.

We first show that there exists a k-$L(2,1)$-labelling L of G_k with $L(u) = k, L(v) = 0$ and $(L(a_v), L(a_u)) = (k-1, 1)$.

We need $\{L(b_1), \ldots, L(b_{k-5})\} = \{3, \ldots, k-3\}$. Then let $L(c_{l,j}) = 0$, $L(d_{l,j}) = k$, $L(c_{r,j}) = k$ and $L(d_{l,j}) = 0$ for $1 \leq j \leq k-5$. So by Lemma 5.12 we obtain a valid labelling.

Next we show that there is no k-$L(2,1)$-labelling of G_k with $L(u) = k$ and $L(v) = 0$ and $(L(a_v), L(a_u)) \neq (k-1, 1)$.

Assume that $l_1 = 1$ and $l_2 = 3$ so $L(a_v) = 3$ and $L(a_u) = 1$. Then the vertices $b_1, \ldots, b_{k-5}$ must take distinct labels from $\{5, \ldots, k-1\}$. However if b_j is labelled $k-1$ the only label $c_{l,j}$ and $c_{r,j}$ can be labelled with is 0 but $c_{l,j}$ and $c_{r,j}$ must have distinct labels. Therefore this labelling is not possible.

Now suppose $l_1 = k-3$ and $l_2 = k-1$ then $L(a_v) = k-1$ and $L(a_u) = k-3$. Then the vertices $b_1, \ldots, b_{k-5}$ must take distinct labels from $\{1, \ldots, k-5\}$. However if b_j is labelled 1 the only label $c_{l,j}$ and $c_{r,j}$ can be labelled with is k but $c_{l,j}$ and $c_{r,j}$ must have distinct labels. Therefore this labelling is not possible. □

5.2.4 Summary

The following theorem summarises the results of this section and follows immediately from Lemmas 5.10, 5.11, 5.13, 5.14 and 5.15.

Theorem 5.16. *Let $k \geq 4$ be fixed. There is a k-$L(2,1)$-labelling L of G_k with $L(u), L(v) \in \{0, k\}$ if and only if the following conditions are satisfied.*

1. *If $(L(u), L(v)) = (0, 0)$, then $(L(a_u), L(a_v)) \in \{(2, k), (k, 2), (k-2, k), (k, k-2)\}$.*

2. *If $(L(u), L(v)) = (k, k)$, then $(L(a_u), L(a_v)) \in \{(2, 0), (0, 2), (k-2, 0), (0, k-2)\}$.*

3. *If $(L(u), L(v)) = (k, 0)$, then $(L(a_u), L(a_v)) = (1, k-1)$.*

4. If $(L(u), L(v)) = (0, k)$*, then* $(L(a_u), L(a_v)) = (k-1, 1)$.

5.3 k-$L(2,1)$-labelling for planar graphs is NP-complete for $k \geq 4$

We reduce Planar Cubic Two-Colourable Perfect Matching to Planar k-$L(2,1)$-Labelling. Suppose we are given a cubic planar graph G corresponding to an instance of Planar Cubic Two-Colourable Perfect Matching. From G we construct a graph K which has the property that K has a k-$L(2,1)$-labelling if and only if G has a two-coloured perfect matching.

In order to show this we also construct an auxiliary graph H and define what we call a *coloured orientation*. Then we show that G has a two-coloured perfect matching if and only if H has a coloured orientation and finally that H has a coloured orientation if and only if K has a k-$L(2,1)$-labelling.

H is obtained by replacing every edge of G with the gadget as depicted in Fig. 5.11, where the endpoints of the edge being replaced are u, v.

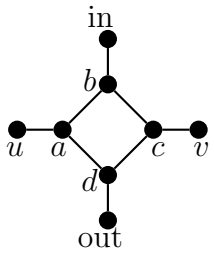

Figure 5.11: An auxiliary edge.

The gadget has two special vertices labelled *in* and *out*, which we call the *invertex* and the *outvertex* and we explain in a moment. The edges incident with them are called the *inedge* and *outedge*, respectively. We use the phrase *auxiliary edge* to refer to a subgraph of H that has replaced an edge of G, that is, any of the copies of the gadget from Fig. 5.11. A coloured orientation of an auxiliary graph H is a colouring of the vertices of H with black and white and an orientation of some of the edges satisfying certain properties. The indegree and outdegree of a vertex v are the number of edges oriented towards v and the number of edges oriented away from v, respectively. Unoriented edges are

not counted towards indegree and outdegree. A coloured orientation must satisfy the following properties.

- Every vertex is adjacent to at most one vertex of the opposite colour.
- An edge is oriented if and only if it is monochromatic.
- Every vertex except those labelled *out* has outdegree at most two and indegree at most one.
- Every vertex labelled *out* has indegree zero.

We say a coloured orientation is *good* if every vertex of degree three is adjacent to precisely one vertex of the opposite colour and has indegree and outdegree one.

Lemma 5.17. *Let G be a cubic planar graph and let H be the corresponding auxiliary graph. If G has a two-coloured perfect matching then H has a good coloured orientation.*

Proof. First colour the vertices of H that were present in G with the same colour that they receive in G.

We next colour the vertices of auxiliary edges where both endpoints of the corresponding edge in G receive the same colour. The in- and outvertex receive the same colour as the endpoints of the corresponding edge in G and the vertices on the four-cycle receive the opposite colour. We orient this cycle to form a directed cycle, see Fig. 5.12.

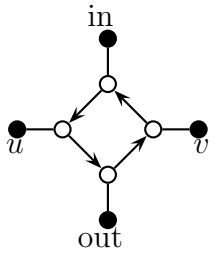

Figure 5.12: Good coloured orientation of an auxiliary edge if u and v receive the same colour.

Now we colour all the other vertices and orient edges as follows. Vertices remaining uncoloured all belong to auxiliary edges for which the endpoints

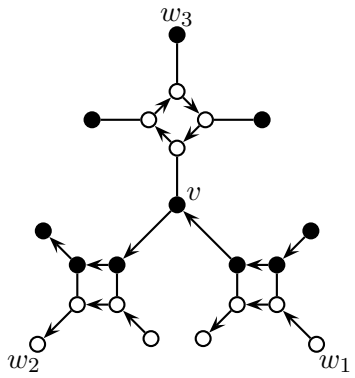

Figure 5.13: Assignment of a good coloured orientation on H.

of the corresponding edge in G are coloured differently in G. We start by choosing an auxiliary edge e between a black vertex v and a white vertex w_1 of G. In G, v has two white neighbours and one black neighbour. Call the other white neighbour w_2. Colour the outvertex of e black and orient the edge incident with it away from the outvertex. Now follow the shortest path from the outvertex, through v and to the invertex of the auxiliary edge vw_2. Colour every uncoloured vertex on this path black and orient every edge consistently with the path. At each stage of this colouring/orientation process we will colour and orient a path like this from an outvertex of an auxiliary edge, through a vertex present in G to an invertex of a neighbouring auxiliary edge, see Fig. 5.13. It only remains to describe how to choose the outvertex and invertex pair forming the endpoints of each path. The first pair is chosen as above. Otherwise, if at some stage, we colour an invertex of an auxiliary edge f with colour c and the outvertex f is still uncoloured then at the next stage we form a path from the outvertex of f colouring the vertices on it with the opposite colour to c. If the outvertex of f has already been coloured then we choose another auxiliary edge for which the outvertex is uncoloured. This method ensures that at each stage there is at most one auxiliary edge with the outvertex coloured and the invertex uncoloured and at most one auxiliary edge with the outvertex not coloured but the invertex coloured. Such an uncoloured outvertex is always the next one to be coloured.

Due to the construction process, every vertex of H which is also present in G is adjacent to exactly one vertex of the opposite colour and has indegree and outdegree one. Clearly the same is true for all vertices of degree three of auxiliary edges where both endpoints of the corresponding edge in G receive the same colour. Now consider an auxiliary edge which corresponds to a dichromatic edge e in G. Due to the colouring/orientation process the invertex and outvertex must receive opposite colours and the shortest path from each of them to the endpoints of e with the same colour is monochromatic. It follows that all vertices of degree three on the auxiliary edge must be adjacent to exactly one vertex of the opposite colour and have indegree and outdegree one, see Fig. 5.13. Therefore the method yields a good coloured orientation. □

Lemma 5.18. *Let G be a cubic planar graph and let H be the corresponding auxiliary graph. If H has a coloured orientation, then it has a good coloured orientation.*

Proof. Consider the possible coloured orientations of an auxiliary edge. Because each vertex is adjacent to at most one of the opposite colour the only ways in which the four-cycle of an auxiliary edge may be coloured are with all four vertices receiving the same colour or with a pair of adjacent vertices receiving one colour and the other pair receiving the opposite colour. In the first case we may change the colours of the invertex and outvertex (if necessary) to be the opposite colour to that of the vertices in the four-cycle. (We also remove the orientation of the inedge and outedge if necessary.) In this way both the inedge and the outedge are dichromatic. In the second case the fact that each vertex is adjacent to at most one vertex of the opposite colour forces both the invertex and the outvertex to have the same colour as their neighbour.

From now on we will assume we have a coloured orientation with each auxiliary edge being coloured in this way. We will show that if H has a coloured orientation then it has a good coloured orientation. Let H' be formed from H by deleting all the dichromatic, or equivalently unoriented edges and consider a connected component C of H'. In H' every vertex has

outdegree at most two and indegree at most one. So C is either an isolated vertex, a directed circuit with a number of trees rooted on the circuit and directed away from the circuit or a directed rooted tree in which all edges are oriented away from the root. Bearing in mind the constraints on the in- and outdegree of the vertices, we see that every vertex of degree three in the auxiliary graph has total degree at least two in H'. Leaves of H' correspond to invertices and roots with degree one correspond to either invertices or outvertices. Notice that the isolated vertices of H' can only be invertices or outvertices and by the remarks at the beginning of the proof exactly half of the isolated vertices are invertices. Hence the numbers of invertices and outvertices appearing in H' that are not isolated are equal. So the number of leaves of H' is at most the number of roots of tree components. Consequently each connected component of H', that is not just an isolated vertex, is either a path beginning at an ouvertex and ending at an invertex or a directed circuit. So every vertex of degree three in the auxiliary graph has one out-neighbour, one in-neighbour and one incident unoriented edge. Therefore the coloured orientation is good. □

Lemma 5.19. *Let G be a cubic planar graph and let H be the corresponding auxiliary graph. If H has a good coloured orientation then G has a two-coloured perfect matching.*

Proof. Consider a vertex v of G and let w_1, w_2 and w_3 be its neighbours in G. Suppose without loss of generality that v is coloured black in the good coloured orientation of H. We will show that in H, two of the vertices w_1, w_2, w_3 are coloured white and one is coloured black. Then we only need to assign to any vertex in G the colour it receives in the good coloured orientation in H to obtain a two-coloured perfect matching of G.

Vertex v has two black neighbours and one white neighbour in H. In the proof of Lemma 5.18 we showed that in a good coloured orientation a terminal vertex of an auxiliary edge receives the opposite colour to the unique neighbour in the auxiliary edge of the other terminal vertex of the auxiliary edge. Thus two of the vertices w_1, w_2, w_3 must be coloured white and the other one black. □

Now given an instance G of Planar Cubic Two-Colourable Perfect Matching, we define an instance K of k-$L(2,1)$-labelling. First form the auxiliary graph H. For every vertex v of H add sufficient vertices of degree one with edges joining them to v to ensure that v has degree $k-1$. Now replace each edge that was originally present in H by the gadget G_k identifying the vertices u, v of G_k with the two endpoints of edges of H being replaced. Finally for each outvertex v choose a neighbour w of v with degree one and add $k-2$ vertices of degree one joined to w. To illustrate this, suppose that in G, v is adjacent to w_1, w_2, w_3. In Fig. 5.14 we show how the neighbourhood of v is modified in K. Note that K can be constructed from G in time $O(n)$.

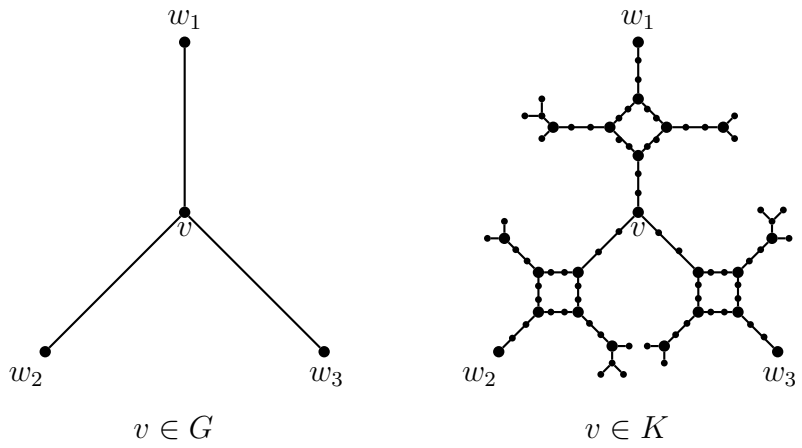

Figure 5.14: Construction of graph K from G for $k = 4$.

Lemma 5.20. *Let G be a cubic planar graph and let H be the corresponding auxiliary graph. Let K be the instance of k-$L(2,1)$-labelling constructed from G as described above. Then H has a good coloured orientation if and only if K has a k-$L(2,1)$-labelling.*

Proof. Suppose that K has a k-$L(2,1)$-labelling L. We now describe how to obtain a coloured orientation of H from L. Any vertex in H corresponds to a vertex of degree $k-1$ in K and so must be coloured 0 or k. Colour a vertex of H white if it corresponds to a vertex labelled 0 in K and black if it corresponds to vertex labelled k in K.

We orient some of the edges of H as follows. If uv is an edge of H then there is a path u, a_u, a_v, v between u, v in K where u is adjacent to a_u and v is adjacent to a_v. Orient the edge uv from u to v if and only if $a_v \in \{0, k\}$ and orient it from v to u if and only if $a_u \in \{0, k\}$. From Theorem 5.16 it follows that in our colouring of H, each vertex of H is adjacent to at most one vertex of the opposite colour, and an edge is oriented if and only if it joins two vertices of the same colour. Consider a vertex $v \in H$. All neighbours of v in K must receive different colours, so in H, v has at most one incoming edge, and at most two out-going edges. Finally let u be an outvertex of H. Then u is part of exactly one copy of the gadget G_k and has a neighbour w of degree $k-1$ that is not part of this copy of G_k. We have $\{L(u), L(w)\} = \{0, k\}$ which means that no other neighbour of u is labelled 0 or k and hence u has indegree 0. Therefore H has a coloured orientation and by Lemma 5.18, H has a good coloured orientation.

Now suppose that H has a good coloured orientation. We will show how to construct a k-$L(2,1)$-labelling L of K. First label all vertices v in K that appear in H, so that $L(v) = 0$ if v is coloured white in H and otherwise $L(v) = k$. Next give labels to all the remaining vertices that appear in a copy of the gadget G_k. Let uv be an edge of H and suppose without loss of generality that $L(u) = 0$. Let u, a_u, a_v, v be the path of length three from u to v in K. If uv is not oriented, let $L(a_u) = k-1$ and $L(a_v) = 1$. If uv is oriented from u to v then let $L(a_u) = 2, L(a_v) = k$ and if uv is oriented from v to u then let $L(a_u) = k, L(a_v) = 2$. Furthermore if $w \in V(H)$ then, because we start from a good coloured orientation of H, its three neighbours in K receive different labels. Then Theorem 5.16 shows that L may be extended so that any vertex appearing in a copy of G_k receives a label. For each outvertex, its neighbour of degree $k-1$ must be labelled. This can be done because each edge in H adjacent to an outvertex x is oriented away from x, so one of the labels $0, k$ is always available. Finally the vertices of degree one form an independent set and are all adjacent to vertices of degree $k-1$ that have received label 0 or k. So they may be labelled. Hence K has a k-$L(2,1)$-labelling. □

We now return to the main problem of this chapter. The following the-

orem which is the main statement of this chapter follows immediately from Theorem 5.9 and Lemmas 5.17, 5.18, 5.19 and 5.20.

Theorem 5.21. *Problem 5.1 is NP-complete.*

Chapter 6

Conclusion and open problems

In Chapter 2 we calculated the expected clustering coefficient of the Móri graph for $\beta > 0$. Bollobás and Riordan essentially calculated the expected clustering coefficient of the Móri graph for $\beta = 0$. However determining the expected clustering coefficient of the Móri graph for $-1 < \beta < 0$ remains an open problem.

Furthermore it remains an open question whether the proof in Chapter 3.6 can be generalised to show that the search algorithm S_0 is also optimal for the Móri graph for any $m \geq 1$. In this case it would follow that a lower bound for the expected searching time in the Móri graph is $\Omega(n)$ in the strong model.

In Chapter 4 we showed that, given a graph G and an integer k, it is NP-complete to determine whether G can be oriented so that the Wiener Index is at most k. Furthermore we gave a linear time algorithm that for a fixed l decides for a planar graph G whether there is an orientation of G so that its diameter is at most l. Recall Problem 4.4
Instance: A planar graph G and integer k.
Question: Is there an orientation of G so that the Wiener Index of $\vec{G}$ is at most k?
and the following problem
Instance: A planar graph G with maximum degree three, integer k.
Question: Is there an orientation of G so that the Wiener Index of $\vec{G}$ is at most k?
Our first two open questions are to determine their complexity.

In Chapter 5 we proved the NP-completeness of Planar k-$L(2,1)$-labelling by reduction of Planar Cubic Two-Colourable Perfect Matching to Planar k-$L(2,1)$-labelling. However the proof could be significantly simplified if it could be proven that the following problem is NP-complete.

Problem 6.1.
Instance: A planar cubic graph G.
Question: Is there a black and white colouring of G so that every vertex is adjacent to exactly two vertices of the same colour and one of the opposite colour?

This problem with cubic graphs as an input was shown to be NP-complete in [52]. It then was used in [30] to show the NP-completeness of 4-$L(2,1)$-labelling for general graphs. If Problem 6.1 was proven to be NP-complete the NP-completeness of Planar 4-$L(2,1)$-labelling could be proven with the same methods as the NP-completeness of 4-$L(2,1)$-labelling was proven for general graphs in [30]. For any $k \geq 5$ the NP-completeness of Planar k-$L(2,1)$-labelling then could be proven by replacing the gadget G_4 in the proof for the NP-completeness of Planar 4-$L(2,1)$-labelling by G_k and applying a very much simplified version of the orientation argument used in Chapter 5.3.

Bibliography

[1] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman. Search in power law networks. *Physical Reviews E*, 64:1–8, 2001.

[2] J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier. Fixed parameter algorithms for dominating set and related problems on planar graphs. *Algorithmica*, 33:461–493, 2002.

[3] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[4] A.-L. Barabási and R. Albert. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.

[5] H. L. Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11:1–21, 1993.

[6] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25:1305–1317, 1996.

[7] H. L. Bodlaender, T. Kloks, R. B. Tan, and J. van Leeuwen. Approximations for λ-coloring of graphs. *The Computer Journal*, 47:193–204, 2004.

[8] F. Boesch and R. Tindell. Robbins' theorem for mixed multigraphs. *American Mathematical Monthly*, 87:716–719, 1980.

[9] B. Bollobás and O. Riordan. The diameter of a scale-free random graph. *Combinatorica*, 24:5–34, 2004.

[10] B. Bollobás and O. M. Riordan. Mathematical results on scale-free random graphs. In S. Bornholdt and H. G. Schuster, editors, *Handbook of Graphs and Networks: From the Genome to the Internet*, pages 1–34. Wiley-VCH, Berlin, 2003.

[11] B. Bollobás, O. M. Riordan, J. Spencer, and G. Tusnády. The degree sequence of a scale-free random graph process. *Random Structures and Algorithms*, 18:279–290, 2001.

[12] B. Bollobás and A. Scott. Separating systems and oriented graphs of diameter two. *Journal of Combinatorial Theory, Series B*, 97:193–203, 2007.

[13] P. G. Buckley and D. Osthus. Popularity based random graph models leading to a scale-free degree sequence. *Discrete Mathematics*, 282:53–63, 2004.

[14] T. Calamoneri. The $L(h, k)$-labelling problem: A survey and annotated bibliography. *The Computer Journal*, 49:585–608, 2006.

[15] G. J. Chang and D. Kuo. The $L(2, 1)$-labeling problem on graphs. *SIAM Journal on Discrete Mathematics*, 9:309–316, 1996.

[16] F. Chung and L. Lu. Connected components in random graphs with given degree sequences. *Annals of Combinatorics*, 6:125–145, 2002.

[17] F. Chung and L. Lu. The average distance in a random graph with given expected degrees. *Internet Mathematics*, 1:91–114, 2003.

[18] F. Chung and L. Lu. The volume of the giant component of a random graph with given expected degrees. *SIAM Journal on Discrete Mathematics*, 20:395–411, 2006.

[19] V. Chvátal and C. Thomassen. Distances in orientations of graphs. *Journal of Combinatorial Theory Series B*, 24:61–75, 1978.

[20] C. Cooper and A. Frieze. A general model of web graphs. *Random Structures and Algorithms*, 22:311–335, 2003.

[21] C. Cooper, R. Klasing, and M. Zito. Lower bounds and algorithms for dominating sets in web graphs. *Internet Mathematics*, 2:275–300, 2005.

[22] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos. Bidimensional parameters and local treewidth. *SIAM Journal on Discrete Mathematics*, 18:501–511, 2004.

[23] S. N. Dorogovtsev, J. F. Mendes, and A. N. Samukhin. Structure of growing networks with preferential linking. *Physical Review Letters*, 85:4633–4636, 2000.

[24] E. Drinea, M. Enachescu, and M. Mitzenmacher. Variations on random graph models for the web. Technical report, Harvard U., Dept. of Computer Science, 2001.

[25] P. Duchon, N. Eggemann, and N. Hanusse. Brief annoucement: Non-searchability of random power-law graphs. In *Proceedings of the 26th Annual ACM Symposium on Principles of Distributed Systems*, pages 380–381, 2007.

[26] P. Duchon, N. Eggemann, and N. Hanusse. Non-searchability of random power-law graphs. In *Proceedings of 11th International Conference on Principles of Distributed Systems*, volume 4878 of *Lecture Notes in Computer Science*, pages 274–285. Springer-Verlag, 2007.

[27] R. Durrett. *Random Graph Dynamics*. Cambridge University Press, 2006.

[28] N. Eggemann and S. D. Noble. Minimizing the oriented diameter of a planar graph. *Electronic Notes in Discrete Mathematics*, 34:267–271, 2009.

[29] J. Fiala, P. Golovach, and J. Kratochvíl. Distance constrained labelings of graphs of bounded treewidth. In *Proceedings of ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 360–372. Springer-Verlag, 2005.

[30] J. Fiala, T. Kloks, and J. Kratochvíl. Fixed-parameter complexity of λ-labelings. *Discrete Applied Mathematics*, 113:59–72, 2001.

[31] F. V. Fomin, M. Matamala, and I. Rapaport. Complexity of approximating the oriented diameter of chordal graphs. *Journal of Graph Theory*, 45:255–269, 2004.

[32] F. V. Fomin and D. M. Thilikos. Dominating sets in planar graphs: branch-width and exponential speed-up. *SIAM Journal on Computing*, 36:281–309, 2006.

[33] D. A. Fotakis, S. E. Nikoletseas, V. G. Papadopoulou, and P. G. Spirakis. NP-completeness results and efficient approximations for radiocoloring in planar graphs. In *Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science*, volume 1893 of *Lecture Notes In Computer Science*, pages 363–372. Springer-Verlag, 2000.

[34] D. A. Fotakis, S. E. Nikoletseas, V. G. Papadopoulou, and P. G. Spirakis. Radiocoloring in planar graphs: Complexity and approximations. *Theoretical Computer Science*, 340:514–538, 2005.

[35] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[36] J. R. Griggs and R. K. Yeh. Labelling graphs with a condition at distance 2. *SIAM Journal on Discrete Mathematics*, 5:586–595, 1992.

[37] G. Gutin. m-sources in complete multipartite graphs. *(in Russian) Seryya Fīzīka-Matèmatychnykh Navuk*, 5:101–106, 1989.

[38] G. Gutin. Minimizing and maximizing the diameter in orientation of graphs. *Graphs and Combinatorics*, 10:225–230, 1994.

[39] G. Gutin. Cycles and paths in semicomplete multipartite digraphs, theorems and algorithms: a survey. *Journal of Graph Theory*, 19:481–505, 1995.

[40] F. Havet and S. Thomassé. Complexity of $(p, 1)$-total labelling. Submitted.

[41] B. J. Kim, C. N. Yoon, S. K. Han, and H. Jeong. Path finding strategies in scale-free networks. *Physical Review E*, 65:1–4, 2002.

[42] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 163–170, 2000.

[43] J. Kleinberg. Complex networks and decentralized search algorithms. In *Proceedings of the International Congress of Mathematicians(ICM)*, pages 1019–1033, 2006.

[44] K. M. Koh and B. P. Tan. The diameter of an orientation of a complete multipartite graph. *Discrete Mathematics*, 149:131–139, 1996.

[45] K. M. Koh and B. P. Tan. The minimum diameter of orientations of complete multipartite graphs. *Graphs and Combinatorics*, 12:333–339, 1996.

[46] K. M. Koh and E. G. Tay. On optimal orientations of cartesian products of even cycles. *Networks*, 32:299–306, 1998.

[47] K. M. Koh and E. G. Tay. On a conjecture concerning optimal orientations of the cartesian product of a triangle and an odd cycle. *Discrete Mathematics*, 232:153–161, 2001.

[48] K. M. Koh and E. G. Tay. Optimal orientations of graphs and digraphs: A survey. *Graphs and Combinatorics*, 18:745–756, 2002.

[49] A. E. Koller. *The Frequency Assignment Problem.* PhD thesis, Brunel University, 2005.

[50] A. E. Koller, S. D. Noble, and A. Yeo. Computing the minimum span of an L(2,1)-labelling of an outerplanar graph. In preparation.

[51] J. C. Konig, D. W. Krumme, and E. Lazard. Diameter-preserving orientations of the torus. *Networks*, 32:1–11, 1998.

[52] J. Kratochvíl, A. Proskurowski, and J. A. Telle. Covering directed multigraphs I. colored directed multigraphs. In *Graph-Theoretical Concepts in Computer Science, Proceedings of the 23rd WG '97*, volume 1335 of *Lecture Notes in Computer Science*, pages 242–257. Springer, 1997.

[53] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Symposium on Foundations of Computer Science*, pages 57–65, 2000.

[54] R. E. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM*, 22:155–171, 1975.

[55] Z. Liu, Y.-C. Lai, N. Ye, and P. Dasgupta. Connectivity distribution and attack tolerance of general networks with both preferential and random attachments. *Physics Letters A*, 303:337–344, 2002.

[56] L. Lovász. Graph minor theory. *Bulletin (New Series) of the American Mathematical Society*, 43:75–86, 2005.

[57] O. I. Marichev. *Handbook of Integral Transforms of Higher Transcendental Functions, Theory and Algorithmic Tables*. Ellis Horwood Limited, 1983.

[58] S. B. Maurer. The king chicken theorems. *Mathematics Magazine*, 53:67–80, 1980.

[59] J. E. McCanna. Orientations of the n-cube with minimum diameter. *Discrete Mathematics*, 68:309–310, 1988.

[60] C. McDiarmid. Concentration. In *Probabilistic Methods for Algorithmic Discrete Mathematics*, pages 195–248. Springer, 1998.

[61] T. F. Móri. On random trees. *Studia Scientiarum Mathematicarum Hungarica*, 39:143–155, 2002.

[62] T. F. Móri. The maximum degree of the Barabási-Albert random tree. *Combinatorics, Probability and Computing*, 14:339–348, 2005.

[63] C. H. Papadimitrou. *Computational Complexity*. Addison Wesley, 1994.

[64] J. Plesník. Diametrically critical tournaments. *Časopis Pro Pěstování Matematiky*, 100:361–370, 1975.

[65] J. Plesník. Remarks on diameters of orientations of graphs. *Acta Mathematica Universitatis Comenianae*, 46/47:225–236, 1985.

[66] K. B. Reid. Every vertex a king. *Discrete Mathematics*, 38:93–98, 1982.

[67] F. S. Roberts. Private communication to J. Griggs.

[68] F. S. Roberts and Y. Xu. On the optimal strongly connected orientations of city street graphs I: Large grids. *SIAM Journal on Discrete Mathematics*, 1:199–222, 1988.

[69] F. S. Roberts and Y. Xu. On the optimal strongly connected orientations of city street graphs II: Two east-west avenues or north-south streets. *Networks*, 19:221–233, 1989.

[70] F. S. Roberts and Y. Xu. On the optimal strongly connected orientations of city street graphs III: Three east-west avenues or north-south streets. *Networks*, 22:109–143, 1992.

[71] F. S. Roberts and Y. Xu. On the optimal strongly connected orientations of city street graphs IV: Four east-west avenues or north-south streets. *Discrete Applied Mathematics*, 49:331–356, 1994.

[72] N. Robertson and P. D. Seymour. Graph minors III: Planar tree-width. *Journal of Combinatorial Theory, Series B*, 36:49–64, 1984.

[73] N. Robertson and P. D. Seymour. Graph minors V. Excluding a planar graph. *Journal of Combinatorial Theory Series B*, 41:92–114, 1986.

[74] N. Robertson, P. D. Seymour, and R. Thomas. Quickly excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 62:323–348, 1994.

[75] N. Sarshar, P. O. Boykin, and V. P. Roychowdhury. Percolation search in power law networks: Making unstructured peer-to-peer networks scalable. In *Proceedings of the Fourth IEEE International Conference on Peer-to-Peer Computing (P2P'04)*, pages 2–9, 2004.

[76] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 216–226, 1978.

[77] R. Thomas. Tree-decomposition of graphs. Lecture notes, 1996.

[78] L. Šoltès. Orientations of graphs minimizing the radius or the diameter. *Mathematica Slovaca*, 36:289–296, 1986.

[79] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[80] H. Whitney. 2-isomorphic graphs. *American Journal of Mathematics*, 55:245–254, 1933.

[81] X. Yao, C.-S. Zhang, J.-W. Chen, and Y.-D. Li. On the formation of degree and cluster-degree correlations in scale-free networks. *Physica A*, 353:661–673, 2005.

[82] R. K. Yeh. A survey on labeling graphs with a condition at distance two. *Discrete Mathematics*, 306:1217–1231, 2006.