# Spanning Structures of Graphs

Andrew J. McDowell

Thesis submitted to the University of London
for the degree of Doctor of Philosophy


2014

# Spanning Structures of Graphs

Department of Mathematics
Royal Holloway, University of London

*The mind is the worst place to keep your thoughts. Write them down, set them free.*

## Declaration of Authorship

I, Andrew J. McDowell, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed:

(Andrew J. McDowell)

Date:

# *Preface*

This PhD thesis is formed of work completed while studying at Royal Holloway, University of London. The three chapters that form the majority of the thesis each represent distinct projects that were studied in this time.

Chronologically, the research forming the first chapter, on factors in random graphs was the first to be completed, in collaboration with my supervisor, Stefanie Gerke. This culminated in the paper 'Non-vertex balanced factors in random graphs', published by the Journal of Graph Theory in 2014 [30].

The second chapter, titled Complexity on Eulerian circuits, was work done in collaboration with Iain Moffatt of Royal Holloway and Joanna A. Ellis-Monaghan and Greta Pangborn of Saint Michael's College, Colchester in the US. This work is largely encapsulated by the paper 'DNA origami and the complexity of Eulerian circuits with turning costs' published by the journal Natural Computing in 2014 [17].

The third chapter, on Adversarial resilience of matchings was work carried out in collaboration with Stefanie Gerke and Paul Balister of the University of Memphis. We aim to publish the results of this chapter in the near future.

# *Summary*

In this thesis we are interested in a range of various results regarding various spanning structures of graphs. The problems have little in common other than this shared global scale and graph setting.

In the first chapter we consider the problem of finding thresholds for the existence of an $H$-factor in the Erdős-Rényi random graph $G(n,p)$, for some fixed graph $H$. An $H$-factor is a collection of disjoint copies of the graph $H$, such that every vertex of $G(n,p)$ is contained in exactly one copy of $H$. It is a natural extension of a perfect matching and the study of threshold functions for these properties have been of interest for some time.

We generalise a recent result in this area, which found the thresholds for strictly balanced graphs, and use this generalisation to prove a conjecture from this paper regarding the threshold function for non-strictly balanced graphs. Our results provide the correct threshold for all non-vertex balanced $H$, which are those $H$ such that at least one vertex of $H$ lies only in less dense subgraphs of $H$ than $H$ itself. We also provide improved bounds on the remaining class of graphs for which the conjecture remains unproven, proving an upper bound on the threshold for all $H$ to within a polynomial log term of the known lower bound.

In addition, our generalisation of the original strictly balanced graph results are themselves of interest, and we demonstrate that they provide an almost immediate generalisation to finding the threshold for factors in directed random graphs.

In the second chapter, we analyse the computational complexity of finding a minimum cost Eulerian circuit when possible transitions for the circuit at each vertex in an Eulerian graph are assigned a cost.

This problem arises from a design strategy problem in bio-computing, related to DNA origami, a process by which artificial DNA is designed which will build itself into desired nano-scale structures. This method is a relatively new technique, with applications that have only begun to be explored and our results hopefully provide an insight into the feasibility of applying these results to potential bio-computing problems.

We demonstrate that the problem is, in general, NP-Hard by means of reduction to the 3-SAT problem.

We also demonstrate that the problem remains NP-Hard when restricted to graphs of bounded degree, in particular proving this is the case for 8-regular graphs. We also demonstrate a polynomial time algorithm for 4-regular graphs.

We also consider a possibly related problem, the Eulerian super-path problem which is that of finding whether an Eulerian circuit that respects certain pre-defined paths exists in an Eulerian graph. This problem is known to be NP-Complete and we believed it would remain so if all the pre-defined paths were of length 2, but we prove the opposite, providing a polynomial time algorithm for this case.

In the third chapter we consider a problem of adversarial resilience of perfect matchings in specific models of partially regular random bipartite graphs.

In our main result, we consider a graph with two partition sets of size $n$ and $(1+\varepsilon)n$ respectively, where each vertex in the first partition set has degree $d$, with neighbours chosen uniformly at random from the second partition set. We consider the case in which an adversary with the ability to remove a single edge incident to each vertex of the first partition set of the bipartite graph, aims to destroy the property of a perfect matching existing.

We demonstrate asymptotically tight thresholds for $d$ and $\varepsilon$ for which this adversary can and cannot eliminate the perfect matching.

# *Acknowledgements*

First and foremost I would like to thank my supervisor, Stefanie Gerke. I have been extremely fortunate to have had the opportunity to work with Stefanie. Her insightful advice, constant support and encouragement throughout my four years at Royal Holloway have been invaluable. It has been a pleasure to work with her.

Thank you also to Iain Moffatt, who has been a pseudo-supervisor to me, bringing me exciting new problems, introducing me to new branches of mathematics and always pushing me forwards in both my work and future career.

My thanks too, to all of the staff at the Royal Holloway maths department for their support and for providing a great environment to work in.

I have always, and especially during my PhD, had the most supportive family one could ask for. Thank you to my parents, Avril and Russell, to Hannah and Faith and to my whole extended family who have stayed interested and excited by my progress and encouraged me every step of the way. It can't be overstated just how much that has meant to me.

I am incredibly lucky to have the friends that I do. They've not only helped me get through the tough times, but they have made the good times the most memorable and fun I could imagine. There are too many special people that I'm privileged to count among my friends to name them all here, but in particular I'd especially like to thank Rahul and Bron, Natalie, Guillaume, Jenny, Marco and Lily, Bev, Alex, Elli, Dean, Adam and Mariam who all helped pick me up when I was down and are each a part of my fondest memories of the past four years.

And thank you to Anne-Marie, who has kept me focused, optimistic and driven, but most of all has kept me smiling.

# *Contents*

# *List of Figures*

Chapter 1

# *Introduction*

Much of the study of graph theory is concerned with the analysis and understanding of spanning structures of graphs. These are combinatorial objects that involve all of the vertices or edges of a graph, and are often the most interesting and difficult to study.

By a graph $G$ we mean a set of vertices $V(G) = \{1, 2, \ldots, n\}$ and edges $E(G)$ between these vertices, consisting of pairs of vertices of $G$. Real world systems modelled by graphs might include social networks, in which each vertex represents a person, and there exists an edge between two people if they know each other. Another simple example would be that of a train network, where the vertices are the stations, and an edge exists between stations one stop away from each other. In this thesis, we are not considering what the graph might represent, instead considering the pure abstraction of a graph and what properties we can determine from its structure.

The type of properties we are most interested in here are spanning structures of graphs, which can be of great interest purely for their mathematical properties and they can often yielding surprising and beautiful results. For example the existence of such a global structure can be entirely determined by local properties, while they in turn, often imply interesting local properties of a graph. Equally, many applications of graph theory involve finding or understanding the properties of spanning structures to model or solve a real world problem.

The most classical example of both cases of interest, is also the origin of the foundations of what we now call graph theory, namely the Königsberg bridge problem. The city of Königsberg (now Kaliningrad) is split by the river Pregel, then

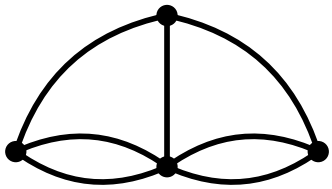


Figure 1.1: A graph representing the bridges of Königsberg.

connected by 7 bridges, into two shores, and two islands. An amusement among locals was to wander the bridges, attempting to cross each one exactly once, never doubling back. No solution was found and as many intellectual curiosities tend to, the problem came to the attention of the mathematically minded.

Leonard Euler, undoubtedly one of the greatest mathematicians of all time, published a solution to the problem in 1735 [45], demonstrating that no such route was possible. He did this by realising that the islands and bridges were themselves irrelevant. All that mattered was which island the traveller was on, and what paths were open to them at each step of the journey. He modelled this problem using vertices to represent the islands and edges for each bridge between them. The observation that any circuit, as it enters and leaves a vertex, uses up two of the edges adjacent to that vertex, meant that unless all of the vertices had even degree (number of edges connected or adjacent to that vertex), a circuit using each edge once would be impossible.

Surprisingly it turns out that this condition for a graph to possess what we now call an Eulerian circuit is not only necessary but also sufficient. If the graph is connected and all vertices are of even degree, then an Eulerian circuit not only can, but must exist.

This is one of the most beautiful examples of how local properties are deeply connected with the global structure of spanning structures and although it is a structure as old as the field of graph theory itself, Eulerian circuits still provide fertile ground for research and analysis today and we consider a problem derived from them in this thesis.

The second type of structure we consider is that of a matching, and its generalisation, that of a factor. A matching is essentially a pairing up of vertices that share an edge between them. This can also be thought of as a subgraph, formed by non-overlapping (disjoint) copies of a single edge. We call a matching perfect if every vertex is paired up with a unique neighbour.

A generalisation of this is to, instead of pairing up vertices which are connected, look for disjoint sets of three vertices which all belong to a single triangle, in other words, all three are pairwise connected by edges. If we can find disjoint triangles such that every vertex is in exactly one triangle, we call this structure a triangle factor. A further generalisation is, instead of a triangle, to look for an $H$-factor, where $H$ is any fixed graph. If $H$ is a single edge, we have a perfect matching, while if $H$ is a triangle, then we have a triangle factor as before. Although matchings and

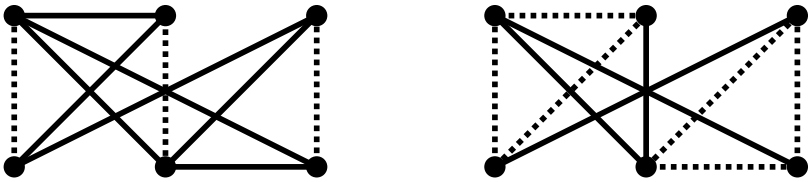


Figure 1.2: Two copies of the same graph, with a perfect matching and a triangle factor respectively, highlighted with dotted lines.

factors are very different from Eulerian circuits, being analogous to a covering of the vertices, rather than a journey that traverses all of the edges of the graph, both have fascinating properties and provide mathematics with a wealth of questions to consider. In the following chapters, we explore and answer a range of such questions, proving a number of new results using and about these structures.

## 1.1 Research Questions and Thesis Structure

In this thesis we tackle a number of diverse questions regarding various spanning structures of several different graph models.

The first chapter is focused on extending the work of Johansson, Kahn and Vu

who proved part of a conjecture that they themselves set in their seminal paper Factors in random graphs [37]. This conjecture concerns the threshold for the existence of factors, a natural and interesting extension of perfect matchings, in random graphs. We prove the conjectured threshold for a range of graphs not previously proven and provide tighter bounds for all remaining cases. The proof of these required several theorems, which in themselves are also useful and applicable results.

The second chapter considers the computational complexity of finding optimal Eulerian circuits in graphs where the paths the circuit may take are assigned costs at each vertex, determined by the pairings of edges a circuit might follow. These models arise from a physical process inherent in designing nano-machines from synthetic DNA strands. It is shown that these problems are, in general and when restricted to several sub-cases computationally difficult to solve while showing polynomial time solutions to several sub-cases of the problem.

The final chapter considers the problem of finding thresholds for which perfect matchings do or do not exist in random bipartite graphs, after an adversary is allowed to delete a number of edges from this random graph. The interest in these models arise from recent interest in ideas of network controllability and general problems of local resilience in graphs.

Chapter 2

# *Factors in random graphs*

## 2.1 Matchings and factors

A *matching* on a graph $G$ is a collection of disjoint edges such that every vertex of $G$ is covered by at most one edge and is one of the most studied types of spanning structures. A perfect matching is a matching where every vertex lies within an edge of the matching and they can be thought of as a pairing up of the vertices of the graph, where every vertex has exactly one partner.

Matchings are one of the easiest structures to explain and see direct (or at least simple) applications for and countless students have been introduced to the field of graph theory by being presented the problem of pairing up dancers at a ball with a graph representing the partners each dancer would be willing to dance with.

For such a bipartite graph, the problem of whether a matching exists is well understood, and entirely determined by Hall's condition [32], which states that a perfect matching exists if and only if, for every subset of either of the partition subsets, the neighbourhood of this set is at least as large as itself. Equally for general graphs, Tutte's Theorem [64] states that a graph $G$ has a perfect matching if and only if for every subset of $U \subseteq V(G)$ the subgraph induced by $V(G) \setminus U$ has at most $|U|$ connected components with an odd number of vertices.

A natural extension of a matching to consider is that of a factor. Rather than having each vertex covered by a single edge, a factor is instead a covering of the graph by disjoint copies of some other, smaller graph, for example a matching is an example of a $K_2$ factor while a triangle factor is a covering of the graph by disjoint copies of triangles, or $K_3$.

In this chapter, we are interested in random graphs, and in particular the Erdős-Rényi random graph $G(n, p)$. The study of matchings here too, has produced a range of research and interesting results. The Erdős-Rényi random graph is, rather than being a specific graph, in reality, a probability distribution or model for producing randomly generated graphs. The graph has $n$ vertices, and then edges are present with some probability $p$. Most studies of these models are interested in finding what properties graphs produced by this model are likely to have, for given values of $p$. In general, $p$ is allowed to be a function of the number, $n$, of vertices of the graph.

We are often interested in the ranges of $p$ for which the properties considered exist with probability tending to 0 or 1 as $n \to \infty$. The transition between these two states is surprisingly small. We call a property monotone if adding edges to a graph with the property always produces a new graph which also possesses the property. An example of a monotone property is 'containing a $K_3$', since adding edges cannot remove a triangle that already exists, while having an even number of edges or an induced 4-cycle are both examples of non-monotone properties. The properties of possessing either matchings or factors are both clearly monotone properties.

For such monotone properties, we have the concept of a threshold function. We formally define it in the next section, but essentially given a property and its threshold function, the random graph $G(n, p)$ with values of $p$ that grow more slowly (in terms of $n$) than this function does not possess the property, and if $p$ grows faster than the threshold it will possess it, both with probability tending to 1.

This small interval in which the property goes from almost certainly not existing to almost certain existence means that finding threshold functions for given properties is of great interest and provides a great deal of insight into the structure of these randomly generated graphs.

As an example, the threshold for the existence of a perfect matching is the same as that for connectivity of the random graph (the existence of a path between any two vertices in $G(n, p)$) or the existence of a Hamiltonian path (a single path which contains every vertex). Threshold functions are equivalent up to constant factors

and but although more precise 'sharp thresholds' or 'hitting time' results can show more precisely that these properties do not occur at exactly the same values of $p$ (and surprisingly that sometimes they do), the fact that they share a common threshold gives an insight into the shape and nature of $G(n,p)$ evolution as $p$ is increased towards 1.

In this chapter we are interested in finding the threshold function for a large class of graphs, for which the function had previously been conjectured but not proven. We prove these conjectures for a large class of graphs, and provide improved bounds on the threshold function for the remaining graphs for which the conjecture remains open. We also prove a generalisation of these and previous results to a directed form of the Erdős-Rényi random graph.

## 2.2 Technical introduction

We will properly state our theorems later in the introduction, after introducing necessary notation and background. However, for readers already familiar with the background or willing to momentarily gloss over the details, let us immediately sketch our main results and methods. In a recent breakthrough (winning a 2012 Fulkerson Prize), Johansson, Kahn, and Vu [37] determined the threshold for a random graph $G$ to be factorable by a strictly balanced fixed graph $H$, and they conjectured the threshold for every $H$. Our main result, Theorem 2.1, establishes their conjecture for 'non-vertex-balanced' graphs $H$, a class of graphs disjoint from strictly balanced ones and a simple application of this result establishes bounds on the threshold to within a polynomial log term of the known lower bound (and conjectured value) for all $H$. The positive side, namely proving the existence of a factor for values of $p$ above the threshold, is the difficult one, and the main idea of the proof is, to cover a fraction of $G$ with copies of a densest subgraph of $H$, then contract that subgraph to a point, extend the cover, and repeat. However, after the first step, two things have changed: the graph $H$ may have become a multigraph and, more significantly, we have committed to correspondences between some vertices of

$G$ and $H$. We manage these difficulties through Theorem 2.2, asserting that, if the vertices of the random graph $G$ are partitioned into classes corresponding to vertices of $H$, then $G$ almost surely has an $H$-factor which respects the partitioning. Our proof follows the steps of the proof in [37]; it is not especially inventive, but neither is it easy. Krivelevich [43] needed a special case of this result, requiring the threshold for a partition-respecting cycle factor, and verified it, but [43] does not include the proof. The result is clearly useful, and it is therefore worth writing down the proof details. We prove a similar result for directed graphs.

Formally, for graphs $H$ and $G$, an *$H$-factor* of $G$ is a collection of vertex-disjoint copies of $H$ in $G$ such that the vertex sets of these copies of $H$, partition the vertices of $G$. Clearly $G$ can only contain an $H$-factor if $|V(H)|$ divides $|V(G)|$. We are mainly interested in large random graphs on $n$ vertices and we assume throughout the paper that $|V(H)|$ divides $n$.

The Erdős-Rényi random graph, $G(n,p)$, is defined to be the graph on $n$ vertices where each edge is present with probability $p$ independently of the absence or presence of any other edge. We call a function $f(n)$ a *threshold* for a graph property $K$ if,

$$\Pr(G(n,p) \text{ satisfies } K) \to \begin{cases} 1 & \text{if } p(n) = \omega(f(n)), \text{ and} \\ 0 & \text{if } p(n) = o(f(n)). \end{cases}$$

Since containing an $H$-factor is an increasing property (that is, adding edges does not destroy any $H$-factor) it is well known that a threshold function exists, see for example [36]. Note that a threshold is unique up to multiplicative positive constants so we will use $\Theta$ notation and with slight abuse of language we will speak of "the" threshold. The study of thresholds for the existence of factors for various classes of graphs $H$ in $G(n,p)$ has attracted considerable interest. The distinctions center around density properties of $H$. We define the *density* of a graph $H$ on at least two vertices, as

$$d(H) = \frac{|E(H)|}{|V(H)| - 1}.$$

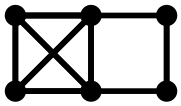


Figure 2.1: A non-vertex balanced graph.

Let $m(H)$ be the maximum density of any subgraph of $H$, that is,

$$m(H) = \max\left\{d(H') \colon H' \subseteq H, |H'| \geq 2\right\}.$$

A graph $H$ is called *balanced* if $m(H) = d(H)$, i.e., if no subgraph of $H$ has density greater than that of $H$, and *strictly balanced* if every proper subgraph of $H$ has density smaller than that of $H$.

For any vertex $v$ of $H$, define the *local density* at $v$ to be the maximum density restricted to subgraphs containing $v$,

$$m(v, H) = \max\left\{d(H') \colon H' \subseteq H, |H'| \geq 2, v \in V(H')\right\}.$$

A graph $H$ is *vertex balanced* if, for all $v \in H$, $m(v, H) = m(H)$.

Note that if $H$ is balanced then it is vertex balanced: a densest subgraph of $H$ is $H$ itself, so $m(v, H)$ and $m(H)$ are both given by $H' = H$, for $m(v, H) = m(H) = d(H)$. Taking the contrapositive, if $H$ is non-vertex balanced then it is not balanced, and not strictly balanced. Graphs may thus be partitioned into those that are non-vertex balanced, those that are strictly balanced, and the rest (those that are vertex balanced but not strictly balanced). An example of a non-vertex balanced graph is shown in Figure 2.1.

The thresholds for $H$-factors for various fixed graphs $H$ have been of interest for a long time. The case $H = K_2$ is simply the threshold for $G$ to have a perfect matching which has been known since 1966 [21], see also [9] for a more precise result. The next $H$-factor threshold result was for trees by Łuczak and Rucinski [49]. Note that $H = K_2$ (for a matching) and trees both are vertex-balanced. For sub-classes of non-vertex-balanced graphs, the threshold is known for graphs $H$ whose minimum degree is less than $m(H)$ [3, 36]. In 2008 the seminal paper by Johansson, Kahn,

and Vu [37] determined the threshold for all strictly balanced graphs (also resolving the so-called 'Shamir's problem' on hypergraph matchings). The special case of finding the threshold of an $H$-factor for the strictly balanced graph $H = K_3$ had been described by Janson, Łuczak and Ruciński as one of the two 'most challenging, unsolved problems in the theory of random structures' [36, p. 96] and was first posed by Ruciński in 1992 [57] (the second problem was 'Shamir's problem').

In their paper Johansson, Kahn and Vu conjecture thresholds for all graphs $H$, depending on whether $H$ is vertex-balanced or not. We restate this formally as Conjecture 1 in Section 2.3. Our first main result establishes this conjecture for all non-vertex-balanced graphs. More precisely, let $\mathrm{th}_H(n)$ be the threshold function for $G(n,p)$ to contain an $H$-factor. We prove the following.

**Theorem 2.1**

*If $H$ is non-vertex-balanced,*

$$\mathrm{th}_H(n) = \Theta\left(n^{-1/m(H)}\right).$$

The main idea of the proof is, first, to embed the dense subgraphs of $H$, giving a 'partial factor' covering a corresponding proportion of the vertices of $G(n,p)$. We then collapse each such subgraph of $H$ to a single vertex, giving a less dense strictly balanced graph (or possibly multigraph). Finally, we extend the partial factor to a full factor using Theorem 2.2, a generalisation to partitioned multigraphs of the strictly balanced result of [37].

To state Theorem 2.2 we need some more notation. Let $e_H = |E(H)|$, $v_H = |V(H)|$ and $V(H) = \{x_1, x_2, \ldots, x_{v_H}\}$. Define the $r$-fold blowup $B(H,r)$ of $H$ as a $v_H$-partite graph with parts $V_1, V_2, \ldots, V_{v_H}$, each of size $r$, with an edge between $v_i \in V_i$ and $v_j \in V_j$ iff there is an edge between $x_i$ and $x_j$ in $H$. In a slight abuse of notation, let $H(n,p)$ be the random subgraph of $B(H, n/v_H)$ obtained by retaining each edge with probability $p$. Likewise, given a multigraph $\mathcal{H}$, we define the random multigraph $\mathcal{H}(n,p)$: the blowup $B(\mathcal{H}, n/v_{\mathcal{H}})$ has as many edges between $v_i \in V_i$ and $v_j \in V_j$ as there are edges between $x_i$ and $x_j$ in $\mathcal{H}$, and again $\mathcal{H}(n,p)$ is the random subgraph of $B(\mathcal{H}, n/v_{\mathcal{H}})$ obtained by retaining each edge with probability $p$.

The setup suggests looking for a 'restricted' $\mathcal{H}$-factor of $\mathcal{H}(n,p)$ where, for each copy of $\mathcal{H}$, each vertex belongs to the corresponding part of $\mathcal{H}(n,p)$. The following theorem shows that below some threshold there is no factor, while above the threshold there is a factor of the restricted form.

**Theorem 2.2**

*Fix a multigraph $\mathcal{H}$ (which may be a simple graph $H$). If $\mathcal{H}$ is strictly balanced, then the threshold for $\mathcal{H}(n,p)$ to contain an $\mathcal{H}$-factor is*

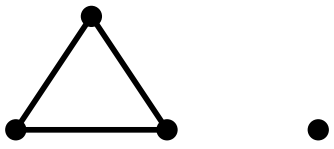
$$\mathrm{th}_{\mathcal{H}}(n) = \Theta\left(n^{-1/m(\mathcal{H})}(\log n)^{1/|E(\mathcal{H})|}\right),$$

*while if $\mathcal{H}$ is not strictly balanced, the threshold satisfies*

$$\mathrm{th}_{\mathcal{H}}(n) = \mathcal{O}(n^{-1/m(\mathcal{H})+o(1)}).$$

*In both cases, above the threshold there is w.h.p. an $\mathcal{H}$-factor in which, in every copy of $\mathcal{H}$, each vertex is in the corresponding part of $\mathcal{H}(n,p)$.*

We note that there is a key difference between the partitioned and usual $G(n,p)$ thresholds for non-strictly balanced graphs. In $G(n,p)$, we show that for non-vertex balanced $H$, the $o(1)$ term can be completely eliminated, while it remains in the form of a log term for strictly balanced $H$. In the partitioned random graph however, there will always be a log term. This can be seen by considering the graph consisting of a triangle and a single isolated vertex.



In $G(n,p)$, this graph is easy to embed, as it is equivalent to a partial factor of triangles, taken over the whole graph, but only covering 3/4 of the vertices. At this point, the remaining spare vertices immediately complete the factor. In the partitioned case, by fixing the position of these triangles, we are implying the existence of a full triangle factor over those corresponding partition sets, and as such,

by the result for strictly balanced graphs, we will require a log term, corresponding to the densest subgraphs.

Lastly we prove that the threshold for digraph factors coincides with that for graphs which is an easy consequence of Theorem 2.2. We define the random directed graph $D(n,p)$ with vertex set $V$ of size $n$, such that for each pair of vertices $u$ and $v$ in $V$, there is an edge between them with probability $p$ independently of all other edges, and each such edge is either $(u,v)$ or $(v,u)$, with probability half each. We can prove the threshold for both strictly balanced and non-vertex-balanced digraphs, in $D(n,p)$. Note that for strictly balanced graphs we can also prove the partitioned form, i.e., the digraph form of Theorem 2.2 also holds.

**Theorem 2.3**

*Fix a digraph $H$. If $H$ is strictly balanced, then the threshold function $\mathrm{th}_H(n)$, for the random directed graph $D(n,p)$ to contain an $H$-factor is*

$$\mathrm{th}_H(n) = \Theta\left(n^{-1/m(H)}(\log n)^{1/e_H}\right),$$

*while if $H$ is non-vertex-balanced,*

$$\mathrm{th}_H(n) = \Theta\left(n^{-1/m(H)}\right).$$

## 2.3 Preliminaries

We first note the following known results which we will need later.

**Theorem 2.4**

*(Ruciński [57]) Let $H$ be a graph with at least one edge, and $F_H(\varepsilon,n)$ be the threshold function for the property that $G(n,p)$ contains a partial $H$-factor covering all but at most $\varepsilon n$ vertices. Then for any fixed $\varepsilon > 0$ the threshold function satisfies,*

$$F_H(\varepsilon,n) = \Theta\left(n^{-1/m(H)}\right).$$

**Theorem 2.5**

*(Alon, Yuster [3], Ruciński [57]) Let $H$ be a graph with minimum degree $\delta(H)$, satisfying $\delta(H) < m(H)$. Then*

$$th_H(n) = \Theta\left(n^{-1/m(H)}\right).$$

In their respective papers, stronger results than what is stated above are actually proved, but we use threshold notation for consistency.

The following two results can be found as Theorems 2.1 and 2.2 in [37]. This chapter's aim is to provide a generalisation of the first result, which allows for improved bounds on the second.

**Theorem 2.6**

*[37] Let $H$ be a strictly balanced graph with $e_H$ edges. Then the threshold function,* $\mathrm{th}_H(n)$ *for $G(n,p)$ to contain an $H$-factor satisfies*

$$\mathrm{th}_H(n) = \Theta(n^{-1/d(H)}(\log n)^{1/e_H}).$$

**Theorem 2.7**

*[37] For $H$ an arbitrary fixed graph, the threshold function* $\mathrm{th}_H(n)$ *for $G(n,p)$ to contain a $H$-factor satisfies*

$$\mathrm{th}_H(n) = \mathcal{O}(n^{-1/m(H)+o(1)}).$$

In [37], the authors define threshold functions $\mathrm{th}_H^{[1]}(n)$ and $th_H^{[2]}(n)$, for a given fixed graph $H$. Firstly, $\mathrm{th}_H^{[1]}(n)$ is defined as the threshold for every vertex in $G$ to be covered by at least one copy of $H$, while $\mathrm{th}_H^{[2]}(n)$ is the threshold for the property of satisfying the following two conditions:

1. every vertex of $G$ is covered by at least one copy of $H$, and
2. for each $x \in V(H)$, there are at least $n/v_H$ vertices $x' \in V(G)$ for which some homomorphism of $H$ into $G$ takes $x$ to $x'$.

This threshold is clearly a lower bound for the threshold for finding a factor $\mathrm{th}_H(n)$ and in [37] the authors conjecture that they are, in fact, equal, proving this for strictly balanced $H$. In this chapter, we will show this conjecture also holds for a large class of non-strictly balanced graphs, including all non-vertex-balanced graphs.

The threshold $th_H^{[2]}(n)$ is completely determined for all graphs and stated without proof in [37]. For completeness we include a proof below. Let

$$s_v = \min\{e(H') : H' \subseteq H, v \in V(H'), d(H') = m(v,H)\}$$

and let $s(H)$ be the maximum over all $s_v$. Clearly $m(v,H) \leq m(H)$ for all $v$, with equality for at least one $v$.

We are now ready to state and prove the following:

**Lemma 2.8**

*If $H$ is vertex-balanced, (i.e. for all $v \in V(H)$, $m(v,H) = m(H)$) then*

$$\mathrm{th}_H^{[2]}(n) = \Theta\left(n^{-1/m(H)}(\log(n))^{1/s(H)}\right).$$

*Otherwise*

$$\mathrm{th}_H^{[2]}(n) = \Theta\left(n^{-1/m(H)}\right).$$

Proof We clearly have 4 cases to consider, namely verifying the two conditions of $\mathrm{th}_H^{[2]}(n)$ for vertex-balanced and non-vertex-balanced graphs.

Firstly we will look at vertex-balanced graphs, i.e. those that satisfy $m(v,H) = m(H)$ for all $v \in V(H)$. Condition 1 of $\mathrm{th}_H^{[2]}(n)$, namely that each vertex of $G$ is covered by at least one copy of $H$, is well studied and exact thresholds can be found as Theorem 3.22 in [36] and follow from results proved by Spencer in [62], and, in this case they are equal to our required bound.

Now we simply have to prove that condition 2 is also satisfied for

$$p = \omega(n^{-1/m(H)}(\log(n))^{1/s}).$$

We note that $Cp = \omega(n^{-1/m(H)})$ for any constant $C$.

Let $V(H) = \{1, 2, \ldots, v_H\}$, our result will follow from partitioning the edge set of $G(n,p)$ into the union of random graphs $G_0, G_1, \ldots, G_{v_H}$ with edge probability $p'$, where $1 - p = \prod_{i=0}^{v_H}(1-p') = (1-p')^{v_H+1}$ and repeatedly applying Theorem 2.4 to find partial factors of $H$. We first apply it in $G_0$, which has edge probability $p' > p/(v_H+1) = \omega\left(n^{-1/m(H)}\right)$, which is sufficient to apply Theorem 2.4 with $\varepsilon = 1/4$. This gives us a partial $H$-factor covering $3n/4$ of the vertices of $G(n,p)$ and hence $(1-\varepsilon)n/v_H = 3n/(4v_H)$ vertices of $G(n,p)$ are covered by each vertex of $H$ with high probability.

For each $i \in \{1, 2, \ldots, v_H\}$ we consider vertex $i$ of $H$ and the vertices of $G_0$, that are already covered by vertex $i$ in a homomorphism of $H$ into $G$, and then, the

random graph induced by the edges of $G_i$ on the vertex set of $G_0$, without those already covered vertices. This leaves us with a set of $n' = (1 - 3/(4v_H))n$ vertices in each $G_i$, that have not already been covered by a copy of the vertex $i$ of $H$, with an independent random edge set. We can consider this as equivalent to the random graph $G(n', p')$, where $p' > p/(v_H + 1) = \omega\left((n')^{-1/m(H)}\right)$ (assuming $|v_H| > 2$, since the case where $|v_H| = 2$ either corresponds to a trivial graph with no edges or a matching in which case both properties are satisfied by the elimination of isolated vertices which happens at $p = \Theta\left(n^{-1/1}(\log(n))^{1/1}\right)$ as required). This allows us to apply again Theorem 2.4 to find another set of partial factors on three quarters of the remaining vertices, giving us in total $(6/(4v_H) - 9/(16v_H^2))n > n/v_H$, for $v_H > 1$, vertices covered by vertex $i$ of $H$ as required.

We now consider graphs that are non-vertex-balanced and so do not satisfy $m(v, H) = m(H)$ for all $v \in V(H)$. As before, the threshold for covering is known, and is in fact lower than our required threshold here.

The same argument for proving condition 2 as above applies since we only required $p = \omega(n^{-1/m(H)})$, so it follows that, for these graphs, both conditions are satisfied for $p = \omega(n^{-1/m(H)})$. It only remains to show that the threshold is not lower than this for such $H$. This follows from another result, proved by Ruciński and Vince [58]. They prove, that for any vertex of $G(n, p)$, the threshold for it being covered by a particular vertex $v$ of $H$ is $n^{-1/m(v,H)}$.

With the result above in mind, we define the following; for a vertex $v_G \in V(G(n, p))$, we let $X_{v_G}$ be the indicator variable for $v_G$ being covered by a copy of $v$, where $v \in V(H)$ satisfies $m(v, H) = m(H)$, namely $X_{v_G} = 0$ if it is not covered, and $X_{v_G} = 1$ if it is. Suppose that condition 2 is satisfied with high probability. Therefore we have that

$$\mathbb{E}\left(\sum_{v_G \in G(n,p)} X_{v_G}\right) = \sum_{v_G \in G(n,p)} \mathbb{E}(X_{v_G}) > n/v_H.$$

Suppose that $p = o(n^{-1/m(v,H)})$. We know that for $p$ in this range, $X_{v_G} = 0$, with high probability, and therefore $\mathbb{E}(X_{v_G}) = o(1)$. Since there are only $n$ choices

of vertex for $v_G$, we have a contradiction. Therefore the threshold $\text{th}_H^{[2]}(n)$ is not $o(n^{-1/m(H)})$, and so must be $n^{-1/m(H)}$, as required. ∎

In [37], it is conjectured that $\text{th}_H^{[2]}(n) = \text{th}_H(n)$, so in light of the above, this can be restated as the following

**Conjecture 1**

*[37] If $H$ is vertex-balanced, i.e. if for all $v \in V(H)$, $m(v, H) = m(H)$, then*

$$\text{th}_H(n) = n^{-1/m(H)}(\log(n))^{1/s(H)}.$$

*Otherwise*

$$\text{th}_H(n) = n^{-1/m(H)}.$$

## 2.4 Theorem 2.1

The first case of Conjecture 1 has been proved for strictly balanced $H$, and now we will prove the second statement in its entirety, namely we prove that the threshold for containing an $H$-factor is $\text{th}_H(n) = \text{th}_H^{[2]}(n) = n^{-1/m(H)}$ for graphs where $m(v, H) < m(H)$ for some $v \in V(H)$. We begin by demonstrating that the result follows from Theorem 2.2, and then in Section 2.5, we return to prove Theorem 2.2.

In general terms, the main idea of this chapter, is to 'collapse' dense sub-graphs of $H$ to get a new graph (or possibly multigraph) $\mathcal{H}$, which we will formally define later.

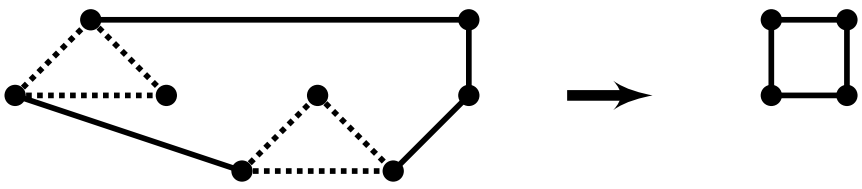


Figure 2.2: The collapsing method.

Since, we have $m(v, H) < m(H)$ for some $v$, we know that at least one vertex of $H$ does not belong to any dense subgraphs of $H$. As a result, we will only need to

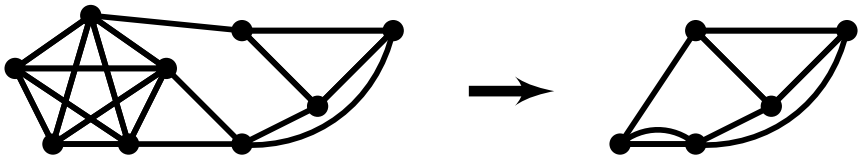


Figure 2.3: A more complicated non-vertex balanced graph.

cover a linear fraction of the vertices of $G(n,p)$ with these dense subgraphs, since our factor will contain at least $n/v_H$ vertices to be covered by copies of these less dense vertices.

Once we have embedded the dense subgraphs, we then use Theorem 2.2, treating these (collapsed in $H$) embedded graphs as single vertices and finding a new, equally or less dense $\mathcal{H}$-factor on these collapsed vertices, along with the remaining uncovered vertices of $G(n,p)$. This will translate to the required factor in our original graph.

To do this for the graph in Figure 2.2, we would simply require a generalisation that allows us to partition our vertices and choose which vertex of $\mathcal{H}$ will 'cover' the vertices of $G(n,p)$ in our factor. However, in a more general case, after collapsing vertices in $H$ we may no longer be left with a graph, but a multigraph, hence the required level of generalisation to use this method.

In the example in Figure 2.3, the densest subgraph is clearly the $K_5$, so we would collapse this to a single vertex. However, one vertex of $H$ contains edges to two vertices of this subgraph, leaving us with a multigraph. It is also worth noting that the density of $K_5$ is 2.5 and since every vertex has degree at least 3, this is an example of a non-balanced graph for which the threshold for the existence of a factor could not be proven by the minimum degree result [3], and where we can provide the optimal threshold, improved on that provided by Theorem 2.7.

To define our collapsing method formally, we begin with some observations on the effects of vertex collapsing on the density of $H$. We know that $m(v,H) = m(H)$ for some vertices $v$, and these are the vertices that we collapse. For each such $v$, we, in turn, choose a subgraph $H'$ such that $v \in H'$ and $d(H') = m(H)$. We now collapse all the vertices in $H'$ into a single vertex. Giving us a new (possibly multi)

graph, which we will call $H_1$, which has the vertices of $H \backslash H'$, with an additional vertex $v_1$, and an edge for each edge of $H$ with an endpoint in $H \backslash H'$. We continue this process, going from $H_i$ to $H_{i+1}$, at each stage, collapsing a subgraph of density $m(H)$ until none remain. The final graph which contains no subgraphs of density $m(H)$, we will call $\mathcal{H}$. We prove the rigour of this statement in the following lemma.

**Lemma 2.9**

*The collapsing process, described above, terminates after a finite number of steps, producing a unique multigraph $\mathcal{H}$, with $m(\mathcal{H}) < m(H)$.*

Proof Firstly note that the density of $H_1$, defined in the same way for multigraphs as for graphs, is

$$d(H_1) = \frac{e(H_1)}{(v(H_1) - 1)} = \frac{e(H) - e(H')}{v(H) - v(H') + 1 - 1} = \frac{e(H) - e(H')}{(v(H) - 1) - (v(H') - 1)}. \quad (2.1)$$

Noting that $d(H) \leq m(H) = d(H')$, we can see that the above gives us $d(H_1) \leq d(H) \leq m(H)$. If instead of $H$ and $H_1$, we consider any subgraph of $H$ containing the vertices we are going to collapse and the resulting subgraph of $H_1$, the same inequality shows that we have not created any subgraph in $H_1$ of density greater than $m(H)$. In fact, considering the following for positive numbers $a, b, c$ and $d$;

$$\frac{a - c}{b - d} \geq \frac{a}{b} \iff \frac{a}{b} \geq \frac{c}{d}$$

(assuming $b > d$), and noting that we only have equality on one side if we have it on both, it follows from (2.1) that any vertex that is in a subgraph of density $m(H)$ in $H_{i+1}$, must have also been in such a graph in $H_i$.

Since we are considering $H$ such that at least one vertex, $v$ satisfies $m(v, H) < m(H)$, the above shows that the collapsing process will never produce a subgraph of density $m(H)$ containing these vertices and hence they will never be collapsed. This ensures that once all subgraphs have been collapsed, we will not be left with a single point, and that $m(\mathcal{H}) < m(H)$.

The above also demonstrates that while the choice of dense subgraph to collapse will result in different $H_i$, ultimately, this process will always terminate with the

same final multigraph, which we call $\mathcal{H}$. To see why this follows, suppose a vertex lies in two different subgraphs, which we could choose to collapse. By (2.1) applied to the subgraph induced by the union of the two dense subgraphs, the new subgraph, formed by the collapsing process, will still have density $m(H)$ and so the remaining vertices will be collapsed at a later stage to the same point.

This implies the existence of an equivalence relation on the vertices of $H$, where we say $u, v \in V(H)$ satisfy $u \sim v$ if and only if they both lie in a subgraph of $H$ of density $m(H)$. This is indeed an equivalence relation since supposing a vertex lies in two subgraphs of density $m(H)$, by the above the union of these subgraphs also has density $m(H)$ and clearly reflexivity and symmetry are also satisfied by this definition. Formally $\mathcal{H}$ is the graph generated by collapsing these equivalence classes into single vertices. ∎

It is clear that if we can embed the collapsed, dense subgraphs of $H$, required for a factor, and then embed the edges of $\mathcal{H}$ we will have our required factor. Firstly, we prove that we can embed these dense subgraphs as required. Consider the graph $H'$ with vertex set $V(H)$ and edge set $E(H) - E(\mathcal{H})$, (i.e. $H'$ contains only those edges collapsed by the above process). Let $\mathrm{th}_{H'}(n)$ be the threshold for embedding a factor of $H'$ into $G(n,p)$.

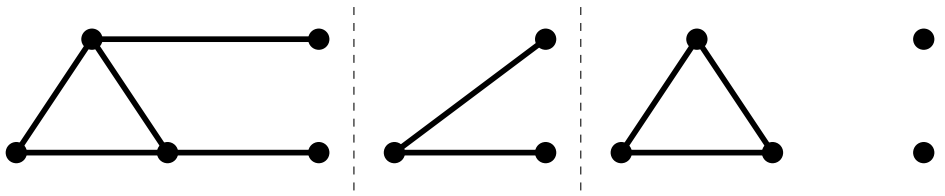


Figure 2.4: An example of an $H$ and its respective $\mathcal{H}$ and $H'$ graphs.

**Lemma 2.10**

*For a non-vertex-balanced graph $H$, and the corresponding $H'$ as defined above, the threshold for the existence of an $H'$-factor satisfies;*

$$\mathrm{th}_{H'}(n) = \Theta\left(n^{-1/m(H)}\right).$$

Proof $H'$ is a subgraph of $H$, and so $m(H') \leq m(H)$, and since $H'$ contains copies of the collapsed subgraphs of density $m(H)$, it must itself satisfy $m(H') = m(H)$. The edges of $H'$ are exactly those that were collapsed in the process that generated $\mathcal{H}$, and hence, any vertices of $H$ that were not collapsed, will be isolated in $H'$. In other words, these vertices will have degree 0 in $H'$. Such vertices must exist, since we assumed that $m(v, H) < m(H)$ for some vertex $v$. Since we then have $\delta(H') < m(H')$, we can apply Theorem 2.5, completing the proof as required. ∎

We now have an $H$-factor, without the edges from each copy of $H$ that are also present in $\mathcal{H}$. To embed these final edges, we now use our generalisation of Theorem 2.7, namely a specific application of Theorem 2.2 to find a factor of $\mathcal{H}$, with the collapsed subgraphs covered by the vertices we require from $H$.

As in [37], we work in (a multigraph generalised form of) $G(n, M)$, the graph chosen uniformly from all $M$-edge graphs on $V$ (although we will use a multigraph generalised form of $G(n, p)$ to prove our results) and derive a generalised form of Theorem 2.6. Since we will be operating with multigraphs and partitioned vertex sets, we need to define some notation.

Let $G$ be a graph on $n$ vertices and let $H$ be a fixed graph with vertex set $\{x_1, x_2, x_3, \ldots, x_{v_H}\}$. Let $\mathcal{H}$ be the multigraph obtained by repeated applications of vertex collapsing of subgraphs of $H$ of density $m(H)$, until no such subgraphs remain. Let $k_{\mathcal{H}} = |V(\mathcal{H})|$ and $h_{\mathcal{H}} = |E(\mathcal{H})|$.

We use the standard method of partitioning the edges of $G(n, p)$ into $G(n, p') \cup G(n, p')$ where there is an edge in $G(n, p)$ if and only if there is an edge in at least one of the $G(n, p')$. Since we are only interested in threshold functions, which are equivalent up to constant factors, and $1 - p = (1 - p')^2$, which implies that $p' > p/2$, we can apply Lemma 2.10 without sacrificing randomness of the edges between these embedded subgraphs. If $\mathcal{H}$ has vertex set $\{y_1, y_2, \ldots, y_{k_{\mathcal{H}}}\}$, we use Lemma 2.10 to find partial factors consisting of $n/v_H$ copies of these collapsed subgraphs (or single vertices, for those that were not collapsed) for each vertex in $\mathcal{H}$. This leaves us with $k_{\mathcal{H}}$ separate classes of the vertices of $G(n, p)$, each containing graphs of density

$m(H)$ or isolated vertices, each corresponding to a vertex of $\mathcal{H}$.

We now wish to find an $\mathcal{H}$-factor between these partitioned sets, but we are only interested in factors that connect the 'correct' vertices together from each partition set, and hence are not interested in the edges within each partition set, or those that are not the prescribed edges between the subgraphs we have already embedded.

We can consider a random multigraph, which we call $\mathcal{H}(n,p)$, using the edges of our second $G(n,p')$, such that if the required $\mathcal{H}$-factor exists here, it will translate into the required $H$-factor in $G(n,p)$. Firstly, the vertex set of $\mathcal{H}(n,p)$ consists of a single vertex for each of the isolated vertices and subgraphs of density $m(H)$, that we have embedded into $G(n,p)$. We maintain the partition of these new vertices into equal sets of size $n/v_H$, according to the vertex of $\mathcal{H}$, they correspond to in the initial embedding. Note that this means that $\mathcal{H}(n,p)$ does not have $n$ vertices, but rather $k_{\mathcal{H}}n/v_H$, which is however, a constant multiple of $n$.

For the edge set of $\mathcal{H}(n,p)$, we use the second set of edges $G(n,p')$, to ensure independence. We consider a pair of vertices, $v_1$ and $v_2$ in $\mathcal{H}(n,p)$, noting that we can also consider $v_1$ and $v_2$ as sets of vertices of $G(n,p)$, and the mapping $\sigma : v(H) \to G(n,p)$ that describes the already embedded subgraphs that form the vertices of $\mathcal{H}(n,p)$. For each $x_1$ and $x_2 \in V(H)$, with $(x_1,x_2) \in E(H)$ and $\sigma(x_1) \in v_1$ and $\sigma(x_2) \in v_2$, if $(\sigma(x_1),\sigma(x_2)) \in e(G(n,p))$, then we have an edge between $v_1$ and $v_2$ in $\mathcal{H}(n,p)$, noting that we consider each such edge separately. In this way, any factor of $\mathcal{H}$ found in $\mathcal{H}(n,p)$ will automatically translate into a factor of $H$ in $G(n,p)$.

$\mathcal{H}(n,p)$ can also be thought of as a random $k_{\mathcal{H}}$-partite multigraph with $k_{\mathcal{H}}n/v_H$ vertices, and edges between vertices $x$ and $y$ with probability $p$ for each edge between their origin vertices in $H$, and 0 otherwise. Essentially, a series of $h_{\mathcal{H}} = |E(\mathcal{H})|$ bipartite graphs, in the same 'shape' as $\mathcal{H}$. We will work using this random graph model (and the corresponding $\mathcal{H}(n,p)$ model), to prove our results.

It may be helpful for some readers to visualise this as a $k_{\mathcal{H}}$-partite random graph with different edge probabilities for some of the edges between partition sets, rather than a multigraph. For example, a single edge with probability $p^2$, rather than two

edges, each with probability $p$ between vertices. The varying probabilities make this model cumbersome to work with, however, and the multigraph notation is more convenient for use.

For some choices of $H$ it may simply be possible to set all edge probabilities to the minimum of these values, and still find the factor, but our earlier graph, containing a $K_5$, is an example of a graph for which this method would fail.

At this point, we have the exact set-up for Theorem 2.2. As we have shown that $m(\mathcal{H}) < m(H)$, it implies that for $p = \omega(n^{-1/m(H)}) = \omega(n^{-1/m(\mathcal{H})+o(1)})$ (or, if $\mathcal{H}$ is strictly balanced $p > \omega(n^{-1/m(\mathcal{H})}(\log n)^{1/|E(\mathcal{H})|})$) such an $\mathcal{H}$-factor, a.a.s exists, and hence our $H$-factor exists in our original $G(n,p)$, as required.

## 2.5 Theorem 2.2

The proof of Theorem 2.2 largely follows the same steps as the original graph result in [37]. To illustrate the key ideas, we outline the initial setup and then draw out several of the key ideas of the proof, highlighting where they differ from [37]. We begin the proof of Theorem 2.2, with a multigraph generalised version of their Theorem 3.1. This theorem essentially shows that the number of factors in $\mathcal{H}(n,p)$ is close to expectation, by demonstrating that the equivalent process of removing edges from the complete graph, does not remove too many factors at each step.

We state the following theorem for a general multigraph, not necessarily derived from some collapsed graph. The only difference this causes is that in our use above the vertex sets in each partition set are slightly smaller (i.e. of size $n/v_H$ rather than $n/k_{\mathcal{H}}$ since in our application we are not considering the vertices collapsed by the collapsing process), but the proof follows in the same manner (and can be found in the Arxiv version of the paper for this result [29]) with $p = \omega(n^{-1/m(H)}) = \omega(n^{-1/m(\mathcal{H})+\varepsilon})$ as required.

**Theorem 2.11**

*Let $k_{\mathcal{H}} = v(\mathcal{H})$ and $h_{\mathcal{H}} = e(\mathcal{H})$. If $\mathcal{H}$ is strictly balanced let*

$$p = p(n) = \omega\left(n^{-1/m(\mathcal{H})}(\log(n))^{1/s(\mathcal{H})}\right),$$

*and*

$$p = \omega\left(n^{-1/m(\mathcal{H})+\varepsilon}\right),$$

*for some* $\varepsilon > 0$ *otherwise. Let* $M = M(n) = h_{\mathcal{H}}(n/k_{\mathcal{H}})^2 p$*, and let* $\Phi(G)$ *be the number of the* $\mathcal{H}$*-factors in a graph* $G$*, then*

$$\Pr(\Phi(\mathcal{H}(n,p)) \geq (n^{k_{\mathcal{H}}-1}p^{h_{\mathcal{H}}})^{n/k_{\mathcal{H}}}e^{-\mathcal{O}(n)}) \geq 1 - n^{-\omega(1)}.$$

Proof The proof follows the same steps as the original and we move between models equivalent to $G(n,p)$ and $G(n,M)$ (the graph on $n$ vertices chosen uniformly at random from all such graphs with $M$ edges) but with our required partitioned and multi-edge structure to prove our results.

Let $T = T(n) = h_{\mathcal{H}}(n/k_{\mathcal{H}})^2 - M$. Let $e_1, e_2, \ldots, e_{h_{\mathcal{H}}(n/k_{\mathcal{H}})^2}$ be an independent, uniformly random ordering of the edge set of the complete form of our multigraph (i.e $\mathcal{H}(n,p)$ with $p = 1$), which we shall call $KM_n$. Set $G_i$ to be $KM_n - \{e_1, e_2, \ldots, e_i\}$.

We can move between these models as $\mathcal{H}(n,p)$ has $M$ edges with probability $\Omega(1/n)$ and as such if $\mathcal{H}(n,p)$ possesses some property with probability $1 - n^{-\omega(1)}$ then the equivalent $G_i$ with $M$ edges must therefore possess it with probability at least $1 - nn^{-\omega(1)} = 1 - n^{-\omega(1)}$ as required. Equally the number of edges in $\mathcal{H}(n,p)$ is between $M/2$ and $2M$ with probability $1 - n^{-\omega(1)}$ which follows from using a simple Chernoff bound on the (binomially distributed) number of edges. Therefore, if a property holds with probability $1 - n^{-\omega(1)}$ for all $G_i$ with $M/2 \leq M' \leq 2M$ such that $G_i$ has $M'$ edges, then the probability that it also holds for $\mathcal{H}(n,p)$ must be at least $1 - 2n^{-\omega(1)} = 1 - n^{-\omega(1)}$ as required. As all our probability of failure terms will be of the order of $n^{-\omega(1)}$ we can move between these models as required.

Let $\mathcal{F}(G)$ be the set of $\mathcal{H}$ factors on $G$ and we define $\mathcal{F}_i := \mathcal{F}(G_i)$. We then let $\xi_i$ be the fraction of members of $\mathcal{F}_{i-1}$ containing $e_i$. Then, as for the standard graph case, we have for any $t$,

$$|\mathcal{F}_t| = |\mathcal{F}_0|\frac{|\mathcal{F}_1|}{|\mathcal{F}_0|}\ldots\frac{|\mathcal{F}_t|}{|\mathcal{F}_{t-1}|} = |\mathcal{F}_0|(1-\xi_1)\ldots(1-\xi_t),$$

and that therefore, we have

$$\log|\mathcal{F}_t| = \log|\mathcal{F}_0| + \sum_{i=1}^{t}\log(1-\xi_i). \tag{2.2}$$

Here our sums start to differ somewhat from the standard graph case; we have

$$\log|\mathcal{F}_0| = \log((n/k_{\mathcal{H}})!)^{k_{\mathcal{H}}-1} = \frac{k_{\mathcal{H}}-1}{k_{\mathcal{H}}} n\log n - \mathcal{O}(n).$$

Also, we have

$$\mathbb{E}\ \xi_i = \frac{h_{\mathcal{H}} n/k_{\mathcal{H}}}{h_{\mathcal{H}}(n/k_{\mathcal{H}})^2 - i + 1} =: \gamma_i = \mathbb{E}[\xi_i | e_1, \ldots, e_{i-1}]$$

for any choice of $e_1, \ldots, e_{i-1}$.

Therefore we have

$$\sum_{i=1}^{t} \mathbb{E}\xi_i = \sum_{i=1}^{t} \gamma_i = \frac{h_{\mathcal{H}} n}{k_{\mathcal{H}}} \log \frac{h_{\mathcal{H}}(n/k_{\mathcal{H}})^2}{h_{\mathcal{H}}(n/k_{\mathcal{H}})^2 - t} + o(1)$$

provided that $h_{\mathcal{H}}(n/k_{\mathcal{H}})^2 - t > \omega(n)$. We use the same property as [37], namely $\mathcal{A}_t$, which is the event that

$$\left\{ \log|\mathcal{F}_t| > \log|\mathcal{F}_0| - \sum_{i=1}^{t} \gamma_i - \mathcal{O}(n) \right\}.$$

As before, we aim to show that with high probability $\mathcal{A}_t$ does not fail, i.e.

$$\text{for } t \le T, \Pr(\overline{\mathcal{A}_t}) = n^{-\omega(1)}. \tag{2.3}$$

This implies our theorem, since we then have

$$\log \Phi(\mathcal{H}(n,p)) = \log|\mathcal{F}_T| > \frac{k_{\mathcal{H}}-1}{k_{\mathcal{H}}} n\log n + \frac{h_{\mathcal{H}} n}{k_{\mathcal{H}}} \log p - \mathcal{O}(n)$$

(since $M = h_{\mathcal{H}}(n/k_{\mathcal{H}})^2 - T = h_{\mathcal{H}}(n/k_{\mathcal{H}})^2 p$). To prove (2.3), we use the same methods as [37], namely an Azuma's inequality, martingale argument. As before, we will define two auxiliary properties $\mathcal{B}_i$ and $\mathcal{R}_i$ for $i \le 1 \le T-1$, that will allows us to establish control over the concentration of our variables. We set our martingale to have a difference sequence of

$$Z_i = \begin{cases} \xi_i - \gamma_i \text{ if } \mathcal{B}_j \text{ and } \mathcal{R}_j \text{ hold for all } j < i \\ 0 \text{ otherwise.} \end{cases}$$

And so our martingale is $X_t = \sum_{i=1}^{t} Z_i$. We leave the formal definitions of $\mathcal{B}_i$ and $\mathcal{R}_i$ for Section 2.6.3, but in general terms, $\mathcal{R}_i$ states that each vertex is in a

number of copies of $\mathcal{H}$ close to the expected value, along with a second technical condition, while $\mathcal{B}_i$ states that the maximum number of factors using a particular copy of $\mathcal{H}$ is close to the average over all copies of $\mathcal{H}$. For all $i \leq T$, we will have that $\mathcal{B}_{i-1}$ and $\mathcal{R}_{i-1}$ imply

$$\xi_i = o(1/\log n). \tag{2.4}$$

Our martingale analysis will give us that $\Pr(|X_t| > n) < n^{-\omega(1)}$ (i.e. $|X_t| \leq \mathcal{O}(n)$ w.h.p.), and if we have $\mathcal{B}_i$ and $\mathcal{R}_i$ for $i < t \leq T$, we will then have that $X_t = \sum_{i=1}^{t} \xi_i - \gamma_i$ and therefore with probability tending to 1,

$$\sum_{i=1}^{t} \xi_i < \sum_{i=1}^{t} \gamma_i + \mathcal{O}(n) < \mathcal{O}(n \log n).$$

Using this, (2.2), (2.4) and the series expansion for $\log(1-x)$ we get that

$$\log|\mathcal{F}_t| > \log|\mathcal{F}_0| - \sum_{i=1}^{t}(\xi_i + \xi_i^2) > \log|\mathcal{F}_0| - \sum_{i=1}^{t}\gamma_i - \mathcal{O}(n).$$

As in the graph case, we are left with three possibilities for the failure of this to occur and hence,

$$\Pr(\overline{\mathcal{A}}_t) < \sum_{i<t}\Pr(\overline{\mathcal{R}}_i) + \sum_{i\leq t}\Pr(\wedge_{j<i}(\mathcal{B}_j\mathcal{R}_j) \wedge \overline{\mathcal{A}}_i) + \sum_{i<t}\Pr(\mathcal{A}_i\mathcal{R}_i\overline{\mathcal{B}}_i).$$

The previously mentioned martingale analysis shows that the second term is at most $n^{-\omega(1)}$, and we follow the same processes as [37] in Section 2.6 to show that, for $i \leq T$,

$$\Pr(\overline{\mathcal{R}}_i) < n^{-\omega(1)} \tag{2.5}$$

and

$$\Pr(\mathcal{A}_i\mathcal{R}_i\overline{\mathcal{B}}_i) < n^{-\omega(1)}. \tag{2.6}$$

These three bounds give us the required result. ∎

In the next section, rather than just present a slightly modified reproduction of [37], and to make this generalisation of more value to the reader, we have first drawn out what Johansson, Kahn and Vu referred to as 'the heart of the matter' and presented it as a stand-alone result, with our required generalisation and then continuing with the surrounding proofs in Section 2.6.

As demonstrated above, the factor result follows from showing that

$$\Pr(\overline{\mathcal{R}}_i) < n^{-\omega(1)}$$

and

$$\Pr(\mathcal{A}_i\mathcal{R}_i\overline{\mathcal{B}}_i) < n^{-\omega(1)}.$$

In proving the second inequality, a second graph property $\mathcal{C}$ is introduced. The following shows that the failure of $\mathcal{C}$ results in two sets differing by a single vertex, such that the number of factors on the complement of these sets (subject to some restrictions) vary significantly (which can be shown with high probability to not occur). The proof of this revolves around the use of entropy results that we describe below, while in Section 2.6.6, concentration results are used to demonstrate that the resulting event, with high probability, does not occur.

### 2.5.1 Entropy

We follow the results of Chapter 6 of [37] but are left with a modification to make to their Lemma 6.1. As in the original, we have $H(X)$ to be the base entropy of a discrete random variable $X$, i.e.,

$$H(X) = \sum_x p(x) \log \frac{1}{p(x)},$$

where $p(x) = \Pr(x = X)$. Now, in our case, given a vertex $y$ in a random multigraph $G$, we use $X(y, G)$ to be the copy of $\mathcal{H}$ in a randomly chosen $\mathcal{H}$-factor, and $h(y, G) = H(X(y, G))$. We will require a slightly different result than in the original, as we will only be interested in vertices $y$ from a single partition set of our random multigraph. We suppose that our multigraph has the same structure as $\mathcal{H}(n, p)$, i.e. any copy of $\mathcal{H}$ will contain one vertex from each partition set of $G$. Given $V_1$, a partition set of $G$, and recalling that $\Phi(G)$ is the number of $\mathcal{H}$-factors in $G$, we have the following

**Lemma 2.12**

$$\log \Phi(G) \leq \sum_{y \in V_1} h(y, G).$$

Proof This result follows in the same way as in the original, using a variant statement of Shearer's Lemma. Given a random vector $Y = (Y_i : i \in I)$, and $\mathcal{S}$, a collection of subsets of $I$, with repeats allowed, such that each element of $I$ belongs to at least $t$ members of $\mathcal{S}$, then for $S \in \mathcal{S}$, let $Y_S$ be the random vector $Y_i : i \in S$. Given this setup, the variant states that, $H(Y) \leq t^{-1} \sum_{S \in \mathcal{S}} H(Y_S)$. If we let $Y$ be the indicator for the random $\mathcal{H}$-factor (an $\mathcal{H}$-factor chosen uniformly at random from the set of all $\mathcal{H}$-factors in the complete form of our multigraph), then $I$ is the set of copies of $\mathcal{H}$ in the complete form of our multigraph and $\mathcal{S}$ is the collection of sets $S_x$, where $S_x$ is the collection of copies of $\mathcal{H}$ containing a vertex $x$, taken only over $x \in V_1$. We have, therefore, that each copy of $\mathcal{H}$ belongs to exactly one $S_x$ in $\mathcal{S}$, and so $t = 1$. It follows that;

$$H(Y) = \sum_{\Phi(G)} \frac{1}{\Phi(G)} \log(\Phi(G)) = \log(\Phi(G))$$

and since $H(Y_S) = h(y, G)$, the proof is complete. ∎

The second entropy result of [37], namely Lemma 6.2, is not specific to graphs, and hence requires no generalisation for our uses. We state it below for reference.

We let $S$ be a finite set, $W : S \to \mathcal{R}^+$, and let $X$ be the random variable taking values in $S$ with probability

$$\Pr(X = x) = W(x)/W(S),$$

where, for a set $A \subseteq S$, $W(A)$ is the sum of $W$ over the members of $A$, i.e. $W(A) = \sum_{x \in A} W(x)$.

**Lemma 2.13**

*If $H(X) > \log |S| - \mathcal{O}(1)$, then there are $a, b \in range(W)$ with*

$$a \leq b < \mathcal{O}(a)$$

*such that for $J = W^{-1}[a, b]$ we have,*

$$|J| = \Omega(|S|)$$

*and*

$$W(J) > 0.7W(S).$$

### 2.5.2 The heart of the matter

We let $\Phi(G)$ be the number of $\mathcal{H}$-factors on a partitioned multigraph $G$, $\mathcal{V}_0$ be the set of vertex sets of size $k_{\mathcal{H}}$ in $\mathcal{H}(n,p)$ with a vertex from each partition set, and $\mathcal{H}(x,G)$ be the set of copies of $\mathcal{H}$ in $G$ containing the vertex $x$, again with each vertex from a separate partition set. We define $D(x,G) = |\mathcal{H}(x,G)|$ to be the number of copies of $\mathcal{H}$ in $G$ containing a vertex $x$, while $D(p)$ is the expectation of $D(x,\mathcal{H}(n,p))$ in $\mathcal{H}(n,p)$ given a randomly chosen $x$.

For $Z$ a disjoint union of elements of $\mathcal{V}_0$, we define $w : \mathcal{V}_0 \to \mathcal{R}^+$ as $w(Z) = \Phi(\mathcal{H}(n,p)\setminus Z)$, i.e. the number of partial $\mathcal{H}$-factors in $\mathcal{H}(n,p)$ with only the vertices in $Z$ not covered.

Lastly, fixing a set of vertices, $Y$ of size $k_{\mathcal{H}}-1$, taken from separate partition sets and two vertices $x$ and $y$, both from the remaining partition set in $V(\mathcal{H}(n,p)\setminus Y)$ we define $w_{x,y} : \mathcal{H}(x,\mathcal{H}(n,p)-\{Y\cup\{y\}\}) \to \mathcal{R}^+$ as $w_{x,y}(K) = w(K\cup Y\cup\{y\})$. In simple terms, $w_{x,y}$ can be thought of as the number of $H$-factors on $\mathcal{H}(n,p)\setminus\{Y\cup\{y\}\}$ that use $K$ as the copy of $\mathcal{H}$ containing $x$ in the $\mathcal{H}$-factor.

**Definition 2.1 (Condition $\mathcal{A}$)**

We say $\mathcal{H}(n,p)$ satisfies $\mathcal{A}(p)$ if the following holds

$$\log(\Phi(\mathcal{H}(n,p))) > \frac{k_{\mathcal{H}}-1}{k_{\mathcal{H}}} n\log n + \frac{h_{\mathcal{H}} n}{k_{\mathcal{H}}}\log p - \mathcal{O}(n).$$

**Definition 2.2 (Condition $\mathcal{R}_b$)**

We say that $\mathcal{H}(n,p)$ satisfies $\mathcal{R}_b(p)$ if the following condition holds.

For each $x \in V$, $|D(x,\mathcal{H}(n,p)) - D(p)| = o(D(p))$.

Informally, $\mathcal{A}(p)$ says that the number of factors is close to expectation, while $\mathcal{R}_b(p)$ says the same for the number of copies of $\mathcal{H}$ that each vertex of $\mathcal{H}(n,p)$ is in.

For a $(k_{\mathcal{H}}-1)$ subset $Y$ of $V(\mathcal{H}(n,p))$, as always with each vertex taken from different partition sets, let $\mathcal{V}_0(Y)$ be the set of $k_{\mathcal{H}}$-subsets containing $Y$, with the final vertex taken from the remaining partition set.

**Definition 2.3 (Condition $\mathcal{C}$)**

We define $\mathcal{C}$ for $\mathcal{H}(n,p)$ as follows: $\mathcal{H}(n,p)$ satisfies $\mathcal{C}$ if for all $(k_{\mathcal{H}}-1)$ subsets $Y$ of $\mathcal{H}(n,p)$, we have the following:

$$\max w(\mathcal{V}_0(Y)) \leq \max\{n^{-2(k_{\mathcal{H}}-1)}\Phi(\mathcal{H}(n,p)), 2\text{med}w(\mathcal{V}_0(Y))\}.$$

We prove the following Theorem;

**Theorem 2.14**

*$\mathcal{A}\mathcal{R}_b\overline{\mathcal{C}}$ implies that there exists a set of vertices $Y$, each taken from $|Y| = v_H - 1$ different partition sets of $\mathcal{H}(n,p)$, and $x, y$ in the remaining partition set, such that we can find a collection $J$ of elements of $\mathcal{H}(y, \mathcal{H}(n,p) - (Y \cup \{x\}))$ and $J'$ from $\mathcal{H}(x, \mathcal{H}(n,p) - (Y \cup \{y\}))$ with $|J| > \Omega(|\mathcal{H}(y, \mathcal{H}(n,p) - (Y \cup \{x\}))|)$, and $w_{y,x}^{-1}|J| = w_{x,y}^{-1}|J'| = [a,b]$ with $a \leq b < \mathcal{O}(a)$ satisfying*

$$\sum_{X \in J} w_{y,x}(X) > 0.7w(Y \cup \{x\})$$

*and*

$$\sum_{X \in J'} w_{x,y}(X) \leq 0.5w(Y \cup \{x\}).$$

Proof Suppose that $\mathcal{A}$, $\mathcal{R}_b$ hold but that $\mathcal{C}$ fails. Therefore we can find at least one set $Y$ at which $\mathcal{C}$ fails. We therefore know that there exists $x$, such that $w(Y \cup \{x\})$ is maximum for choices of $x$ and satisfies

$$w(Y \cup \{x\}) > n^{-2(k_{\mathcal{H}}-1)}\Phi(\mathcal{H}(n,p)).$$

We now choose $y$ with $w(Y \cup \{y\}) \leq \text{med } w(\mathcal{V}_0(Y))$, and $h(y, \mathcal{H}(n,p) - (Y \cup \{x\}))$ maximal, given this constraint.

Given $\mathcal{A}$, we know that

$$\log(\Phi(\mathcal{H}(n,p))) > \frac{k_{\mathcal{H}}-1}{k_{\mathcal{H}}} n \log n + \frac{h_{\mathcal{H}} n}{k_{\mathcal{H}}} \log p - \mathcal{O}(n).$$

While the failure of $\mathcal{C}$ tells us that

$$w(Y \cup \{x\}) = \Phi(\mathcal{H}(n,p) - (Y \cup \{x\})) > n^{-2(k_{\mathcal{H}}-1)}\Phi(\mathcal{H}(n,p)),$$

hence, combining the two we have,

$$\log \Phi(\mathcal{H}(n,p) - (Y \cup \{x\})) > \frac{k_{\mathcal{H}} - 1}{k_{\mathcal{H}}} n \log n + \frac{h_{\mathcal{H}} n}{k_{\mathcal{H}}} \log p - \mathcal{O}(n). \tag{2.7}$$

We use Lemma 2.12 and apply it to the graph $\mathcal{H}(n,p) - (Y \cup \{x\})$. Letting $V_1$ be the partition set containing $x$ and $y$, we have,

$$\log \Phi(\mathcal{H}(n,p) - (Y \cup \{x\})) \leq \sum_{z \in V_1} h(z, \mathcal{H}(n,p) - (Y \cup \{x\})). \tag{2.8}$$

However, we know that we chose $y$ to have maximal entropy, chosen from a set of at least half of possible such $z$, and we also have that for any random variable $X$, the entropy $H(X) \leq \log(|\text{range}(X)|)$ (with equality only if the variable is uniformly distributed).

The range of our random variable is contained in the set of copies of $H$ containing the fixed vertex $z$ in $\mathcal{H}(n,p) - (Y \cup \{x\})$, which by definition, is of size $D(z, \mathcal{H}(n,p) - (Y \cup \{x\})) \leq D(z, \mathcal{H}(n,p))$. We know from $\mathcal{R}_b$ that this is less than $(1 + o(1))D(p)$. Hence, we have that at least half the $z$'s in (2.8) satisfy $h(z, \mathcal{H}(n,p) - (Y \cup \{x\})) \leq h(y, \mathcal{H}(n,p) - (Y \cup \{x\}))$ and the remaining, $n/2k_{\mathcal{H}}$ all satisfy $h(z, \mathcal{H}(n,p) - (Y \cup \{x\})) \leq \log((1 + o(1))D(p))$. Therefore, (2.8) gives us,

$$\begin{aligned}
\log(\Phi(\mathcal{H}(n,p) - (Y \cup \{x\}))) &\leq \sum_{z \in V_1} h(z, \mathcal{H}(n,p) - (Y \cup \{x\})) \\
&\leq \frac{n}{2k_{\mathcal{H}}}(h(y, \mathcal{H}(n,p) - (Y \cup \{x\}) + \log(1 + o(1))D(p))) \\
&\leq \frac{n}{2k_{\mathcal{H}}}(h(y, \mathcal{H}(n,p) - (Y \cup \{x\}) + (k_{\mathcal{H}} - 1)\log n + h_{\mathcal{H}} \log p).
\end{aligned}$$

Rearranging, to get $h(y, \mathcal{H}(n,p) - (Y \cup \{x\}))$ on the left, and substituting from (2.7) we have,

$$h(y, \mathcal{H}(n,p) - (Y \cup \{x\})) \geq (k_{\mathcal{H}} - 1)\log n + h_{\mathcal{H}} \log p - \mathcal{O}(1).$$

By $\mathcal{R}_b$ we have that

$$\begin{aligned}
\log(D(y, \mathcal{H}(n,p) - (Y \cup \{x\}))) &\leq \log D(y, \mathcal{H}(n,p)) \leq \log((1 + o(1))D(p)) \\
&= (k_{\mathcal{H}} - 1)\log n + h_{\mathcal{H}} \log p + \log(1 + o(1)),
\end{aligned}$$

and hence combining with the above, we have

$$h(y, \mathcal{H}(n,p) - (Y \cup \{x\})) > \log(D(y, \mathcal{H}(n,p) - (Y \cup \{x\})) - \mathcal{O}(1). \tag{2.9}$$

We now use our functions $w_{y,x}$ and $w_{x,y}$, previously defined as;

$$w_{y,x}(K) = w(K \cup Y \cup \{x\}) \text{ and similarly } w_{x,y}(K) = w(K \cup Y \cup \{y\}).$$

With $w_{y,x}$ defined on $H(y, \mathcal{H}(n,p) - (Y \cup \{x\}))$; the set of copies of $\mathcal{H}$ containing $y$ in $\mathcal{H}(n,p) - (Y \cup \{x\})$, and similarly, $w_{x,y}$ defined on $H(x, \mathcal{H}(n,p) - (Y \cup \{y\}))$. Simply put, for a copy of $\mathcal{H}$ containing $y$ in $\mathcal{H}(n,p) - (Y \cup \{x\})$, $w_{y,x}$ is the number of $\mathcal{H}$-factors on this set, using that copy of $\mathcal{H}$.

If we consider the random variable $X(y, \mathcal{H}(n,p) - (Y \cup \{x\}))$, which is the copy of $\mathcal{H}$ containing $y$ in a uniformly at random chosen $\mathcal{H}$-factor on $\mathcal{H}(n,p) - (Y \cup \{x\})$, we can see that the probability that $X(y, \mathcal{H}(n,p) - (Y \cup \{x\})) = \mathcal{H}'$ for $\mathcal{H}' \in \mathcal{H}(y, \mathcal{H}(n,p) - (Y \cup \{x\}))$, is

$$w_{y,x}(\mathcal{H}') / \sum_{Z \in \mathcal{H}(y, \mathcal{H}(n,p) - (Y \cup \{x\}))} w_{y,x}(Z).$$

Also note that the denominator is equal to $w(\mathcal{H}(n,p) - (Y \cup \{x\}))$, since by summing only over copies of $\mathcal{H}$, we are counting each $\mathcal{H}$-factor exactly once.

Similarly, $X(x, \mathcal{H}(n,p) - (Y \cup \{y\}))$ is determined by $w_{x,y}$, and the sum $\sum_Z w_{x,y}(Z)$ is equal to $w((Y \cup \{y\}))$.

By the above, we have the setup used for Lemma 2.13, with $S = \mathcal{H}(y, \mathcal{H}(n,p) - (Y \cup \{x\}))$. Noting that $|S| = D(y, \mathcal{H}(n,p) - (Y \cup \{x\}))$, (2.9) gives us the required condition, and we are able to apply the result to $w_{y,x}$. This implies that there exist $a$ and $b \in range(w_{y,x})$, for which we can set $J := w_{y,x}^{-1}([a,b])$, and it will satisfy the following:

$$|J| > \Omega(|\mathcal{H}(y, \mathcal{H}(n,p) - (Y \cup \{x\}))|)$$

and

$$\sum_{z \in J} w_{y,x}(Z) > 0.7 \sum_{z \in H(y, G - (Y \cup \{x\})} w_{y,x}(Z) = 0.7\, w(Y \cup \{x\}).$$

In simple terms, $J$ is of the same magnitude in size as the whole pre-image of $w_{y,x}$, and its elements have overall weight at least a constant multiple of that of the whole set.

Equally we can set $J' = w_{x,y}^{-1}([a,b])$, and we know that

$$\sum_{J'} w_{x,y}(Z) \leq w(Y \cup \{y\}) < 0.5w(Y \cup \{x\}).$$

The first inequality follows from simply summing over the full set containing $J'$, and the second from our original definition of $y$ and $x$. This completes the proof. ∎

Proving that the existence of $Y$, $x$ and $y$ with such properties in our random multigraph is a.a.s. unlikely to happen, requires a range of concentration and technical lemmas, demonstrated in the following sections. In applying this result to Shamir's problem, if instead of considering factors, a matching of hyperedges is required, it has been shown that the proof follows with much more ease using a union bound argument, reducing the technical complexity of the proof considerably [25].

## 2.6 Generalisation of remaining results from [37]

The generalisation to partitioned structures and multigraphs of the remaining results and properties of [37], follow largely from careful consideration of sums and bounds, and formulation of polynomials. The following sections follow the structure of [37] closely, and are largely a technical exercise, that offer little to those who have read the original paper.

To broadly highlight why the generalisation should follow, we note that while limiting the factors to these partitions appears to drastically limit the number of possible copies of $H$, since each partition set is of size linear in $n$, we still have $\mathcal{O}(n^{k_{\mathcal{H}}}) = \mathcal{O}\binom{n}{k_{\mathcal{H}}}$ possible choices of vertices for each $H$, as in the standard case.

We also address the threshold required for applying Theorem 2.2 in obtaining Theorem 2.1. We are not guaranteed strict balance for the resulting $\mathcal{H}$, but regardless, the collapsing process eliminates all subgraphs of density $m(H)$, and hence,

$m(\mathcal{H}) < m(H)$ and so for $p > n^{-1/m(H)}$, we have a greater probability than required within the proof and so with Theorem 2.11, applied to $\mathcal{H}(n,p)$, on the partial factors already embedded during the collapsing process, we have Theorem 2.1 as required.

Throughout the proofs, for clarity in understanding our main result, we treat $\mathcal{H}$ as the graph formed by the collapsing process on some $H$, and that we have $p > n^{-1/m(H)}$ as in Theorem 2.1, but for proving Theorem 2.2 in full generality, $\mathcal{H}$ may not necessarily be derived from some $H$, and we only have $p > n^{-1/m(\mathcal{H})+o(1)}$ (or with a log term for the strictly balanced case). In this case the proof is unchanged, as throughout, as in [37], we only require that if $p = \mathcal{O}(p^{-1/a})$, $\mathcal{H}$ contains no subgraphs of density equal to $1/a$, and that $n^{k_{\mathcal{H}}-1}p^{h_{\mathcal{H}}} = \omega(\log n)$, which follows immediately from the conditions in Theorem 2.2, given that the $o(1)$ term decreases sufficiently slowly.

### 2.6.1 Concentration Results

Firstly we address the usage of the various concentration results in Section 5 of [37]. These results are largely special cases of results by V. Vu that can be found in [41] (with J.H.Kim), [65] and [66].

We will utilise the various polynomial results here without modification, and hence will not repeat the proofs here again.

We will require some of the notation used in this section later, which we outline now. We let $f = f(t_1, t_2, \ldots, t_n)$, be a polynomial of degree $d$ with real coefficients. We say $f$ is normal if its coefficients are positive, with the maximum coefficient being 1, and we note that the results here are also true for $\mathcal{O}(1)$ normal polynomials, which simply means that the polynomial's coefficients have some fixed bound.

We will consider polynomials that are multilinear which means we can express $f$ in the form $f(t) = \sum \alpha_U t_U$, where $U$ ranges over subsets of $[n]$ and $t_U := \prod_{u\in U} t_u$.

Lastly, we need that for a set $L \subseteq [n]$, the partial derivative of order $|L|$ with respect to the variables indexed by $L$ is $\sum_{U\supseteq L} \alpha_U t_{U\setminus L}$, and its expectation, denoted $\mathbb{E}_L$ or $\mathbb{E}_L f$ is $\sum\{\alpha_U \prod_{i\in U\setminus L} p_i : U \supseteq L\}$, where $t_i \sim Ber(p_i)$. Set $\mathbb{E}_j f = \max_{|L|=j} \mathbb{E}_L f$.

We write $\mathbb{E}_L^{'} = \mathbb{E}_L^{'} f$ for the expectation of the non-constant part of the partial derivative of $f$, with respect to $L$, noting that for homogeneous, $f$ of degree $d$, and $0 < |L| < d$, we have $\mathbb{E}_L^{'} f = \mathbb{E}_L f$.

We take the original example used to illustrate the usage of these results, namely that we consider our polynomial $f$ to be the number of copies of $\mathcal{H}$ in our random multigraph $\mathcal{H}(n,p)$ containing a particular, fixed vertex $x_0$.

We have that $f = \sum_U t_U$ where $U$ runs over edge sets of copies of $\mathcal{H}$ in our complete multigraph, containing our vertex $x_0$. We have $\mathbb{E}f = \Theta(n^{k_\mathcal{H}-1}p^{h_\mathcal{H}})$, while for any non empty subset $L$ of edges from the complete graph, the partial derivative will be $\mathbb{E}_L f = \sum_{U \supseteq L} t_{U \setminus L}$. This will be 0 for $L$ that do not satisfy our multigraph structure requirements (i.e. at most one edge from each bipartite pairing forming the multigraph), in the same way that choosing an $L$ not forming a subgraph of $H$ would do, in the standard graph case.

In all theorems in this section we are interested in ensuring that the maximum value of the derivative does not exceed a certain magnitude, and so in this sense, we are not interested in these cases, and so they cause no issue in this generalisation.

Given that our choice of $L$ does satisfy our structure requirements, (and hence will be contained within at least one copy of $\mathcal{H}$ in the complete graph), we can consider the graph formed by the edges of $L$, and the vertex end-points of these edges. Letting $k'_\mathcal{H}$ and $h'_\mathcal{H}$ be the number of vertices and edges respectively of $L$, then if $L$ contains $x_0$, we have $\mathbb{E}_L f = \mathcal{O}(n^{k_\mathcal{H}-k'_\mathcal{H}}p^{h_\mathcal{H}-h'_\mathcal{H}})$, and $\mathcal{O}(n^{k_\mathcal{H}-k'_\mathcal{H}-1}p^{h_\mathcal{H}-h'_\mathcal{H}})$ otherwise. Either way, we have,

$$\mathbb{E}f/\mathbb{E}_L f = \Omega(n^{k'_\mathcal{H}-1}p^{h'_\mathcal{H}}).$$

While we do not have strict balance of $\mathcal{H}$, we do have that it contains no subgraphs of density $m(H)$, and hence we have that $h'_\mathcal{H}/(k'_\mathcal{H}-1) < m(H)$ and recalling that $p = \omega(n^{-1/m(H)})$, we have that $\mathbb{E}f = \Omega(1)$ and that $\mathbb{E}f/\mathbb{E}_L f \geq n^{\Omega(1)}$, as is required for applying the results in this section.

For reference we include the results from the concentration section of [37] below.

**Theorem 2.15**

*The following holds for any fixed positive integer $d$ and positive constant $\epsilon$. Let $f$ be a multilinear, homogeneous, normal polynomial of degree $d$ such that $\mathbb{E}f \geq n^{\epsilon}\max_{1\leq j\leq d}\mathbb{E}_j f$. Then*

$$\Pr(|f-\mathbb{E}f|>\epsilon\mathbb{E}f)=n^{-\omega(1)}.$$

**Theorem 2.16**

*The following holds for any fixed positive integer $d$ and positive constant $\epsilon$. Let $f$ be a multilinear, normal, homogeneous polynomial of degree $d$ such that $\mathbb{E}f=\omega(\log n)$ and $\max_{1\leq j\leq d-1}\mathbb{E}_j f\leq n^{\epsilon}$. Then*

$$\Pr(|f-\mathbb{E}f|>\epsilon\mathbb{E}f)=n^{-\omega(1)}.$$

**Theorem 2.17**

*The following holds for any fixed positive integer $d$ and positive constant $\epsilon$. Let $f$ be a multilinear, normal, homogeneous polynomial of degree $d$ such that $\mathbb{E}f=\omega(\log n)$ and $\max_{1\leq j\leq d-1}\mathbb{E}_j f\leq n^{\epsilon}\mathbb{E}f$. Then*

$$\Pr(|f-\mathbb{E}f|>\epsilon\mathbb{E}f)=n^{-\omega(1)}.$$

**Corollary 2.18**

*The following holds for any fixed positive integer $d$ and positive constant $\epsilon$. Let $f$ be a multilinear, normal, homogeneous polynomial of degree $d$ such that $\mathbb{E}f\leq A$ where $A=A(n)$ satisfies*

$$A\geq\omega(\log n)+n^{\epsilon}\max_{0<j<d}\mathbb{E}_j f,$$

*then*

$$\Pr(f>(1+\epsilon)A)\leq n^{-\omega(1)}.$$

**Theorem 2.19**

*The following holds for any fixed positive integer $d$ and positive constant $\epsilon$. Let $f$ be a multilinear, normal polynomial of degree $d$ with $\mathbb{E}f=\omega(\log n)$ and $\max_{L\neq\emptyset}\mathbb{E}'_L f\leq n^{\epsilon}\mathbb{E}f$. Then*

$$\Pr(|f-\mathbb{E}f|>\epsilon\mathbb{E}f)\leq n^{-\omega(1)}.$$

**Corollary 2.20**

*The following holds for any fixed positive integer $d$ and positive constant $\epsilon$. Let $f$ be a multilinear, normal polynomial of degree $d$ such that $\mathbb{E}f \leq A$ where $A = A(n)$ satisfies*

$$A \geq \omega(\log n) + n^{\epsilon} \max_{L \neq \emptyset} \mathbb{E}'_L f,$$

*then*

$$\Pr(f > (1+\epsilon)A) \leq n^{-\omega(1)}.$$

**Theorem 2.21**

*The following holds for any fixed positive integer $d$ and positive constant $\epsilon$. Let $f$ be a multilinear, normal polynomial of degree $d$ with $\max_L \mathbb{E}'_L f \leq n^{\epsilon}$. Then for any $\beta(n) = \omega(1)$,*

$$\Pr(f > \beta(n)) = n^{-\omega(1)}.$$

### 2.6.2 Martingale

The proof of the bound on the martingale follows exactly as that for the original paper. We have all the same bounds, namely that $|Z_i| < \varepsilon := \log^{-1} n$, and that $\sum_{i=1}^{t} \gamma_i = \mathcal{O}(n \log n)$. and the proof makes no use of the graph setting for the problem. We include the steps below for reference.

We let $X_t = Z_1 + \cdots + Z_t$, and aim to show that

$$\Pr(X_t \geq n) < n^{-\omega(1)}.$$

We have that $Z_i$ is a function of the random sequence $e_1, \ldots, e_i$, but that

$$\mathbb{E}(Z_i | e_1, \ldots, e_{i-1}) = 0,$$

for any choice of the $e_j$'s. Using (2.4), it will follow from the properties $\mathcal{R}$ and $\mathcal{B}$, that $|Z_i| < \varepsilon := \log^{-1} n$. We can apply Markov's inequality to derive the following, for any positive $h$;

$$\Pr(X_t \geq n) = \Pr(e^{h(Z_1 + \cdots + Z_t)} \geq e^{hn}) \leq \mathbb{E}(e^{h(Z_1 + \cdots + Z_t)}) e^{-hn}. \tag{2.10}$$

Using $Z_i = \xi_i - \gamma_i$, we have that $\mathbb{E}(\xi_i | e_1, \ldots, e_{i-1}) = \gamma_i$. Using $0 \leq \xi_i \leq \varepsilon$ and the convexity of $e^x$, we have

$$\mathbb{E}(e^{hZ_i} | e_1, \ldots, e_{i-1}) \leq e^{-h\gamma_i} \left( \left( 1 - \frac{\gamma_i}{\varepsilon} \right) + \frac{\gamma_i}{\epsilon} e^{h\varepsilon} \right).$$

Taylor series expansions show that the right hand side is at most $e^{h^2 \epsilon \gamma_i}$, for any $0 \leq h \leq 1$. Using induction on $t$ we derive the following.

$$\begin{aligned} \mathbb{E}(e^{h(Z_1 + \cdots + Z_t)}) &= \mathbb{E}(\mathbb{E}(e^{h(Z_1 + \cdots + Z_t)} | e_1, \ldots, e_{t-1})) \\ &= \mathbb{E}(e^{h(Z_1 + \cdots + Z_{t-1})} \mathbb{E}(e^{hZ_t} | e_1, \ldots, e_{t-1})) \\ &\leq \mathbb{E}(e^{h(Z_1 + \cdots + Z_{t-1})} e^{h^2 \varepsilon \gamma_t}) \\ &\leq e^{h^2 \varepsilon \sum_{i=1}^{t} \gamma_i}. \end{aligned}$$

Combined with (2.10), we have

$$\Pr(X_t \geq n) \leq e^{h^2 \varepsilon \sum_{i=1}^{t} \gamma_i - hn}.$$

We have that $\sum_{i=1}^{t} \gamma_i = \mathcal{O}(n \log n)$ and $\varepsilon = \log^{-1} n$, and so setting $h$ to be a sufficiently small positive constant, leaves the right hand side as $e^{-\Omega(n)} = n^{-\omega(1)}$, as required.

### 2.6.3 The Properties $\mathcal{B}$ and $\mathcal{R}$

We now define our slightly altered properties $\mathcal{B}_i$ and $\mathcal{R}_i$. We note, as before that in proving (2.5) and (2.6) we can operate in the random graph $\mathcal{H}(n, p_i)$ rather than $G_i$, where

$$p_i = 1 - \frac{i}{h_{\mathcal{H}} (n/k_{\mathcal{H}})^2}.$$

We will define graph properties $\mathcal{B}$ and $\mathcal{R}(p)$ and then the event $\mathcal{B}_i$ will be $\{G_i \text{ satisfies } \mathcal{B}\}$ and $\mathcal{R}_i$ will be $\{G_i \text{ satisfies } \mathcal{R}(p_i)\}$.

In defining $\mathcal{B}$, we use the same notation for the functions $W$, namely that for a finite set $A$ and $W : A \to [0, \infty)$, set

$$\overline{W}(A) = |A|^{-1} \sum_{a \in A} W(a),$$

$$\max W(A) = \max_{a \in A} W(a),$$

and lastly that,

$$\text{maxr } W(A) = \overline{W}(A)^{-1} \max W(A),$$

with med $W(A)$ the median of $W$ on $A$.

For a multigraph $G$ with our required partition structure, and vertex set $V$, let $Z$ be a choice of $k_{\mathcal{H}}$ vertices from $V$, with each element taken from a different vertex partition set. We then let $w_G(Z) = \Phi(G - Z)$. Therefore, $w_G(Z)$ is the number of $\mathcal{H}$-factors in the multigraph induced by $G$ on the vertex set $V \backslash Z$.

It could also be thought of as the number of $\mathcal{H}$-factors in $G$, containing $Z$ as a copy of $\mathcal{H}$, if all edges between the vertices of $Z$ had been added in, where they are not already present.

We also use $w_G(K) = w_G(V(K))$ for $K \in \mathcal{H}(G)$, the set of copies of $\mathcal{H}$ in $G$ (which will contain one vertex from each partition set of $G$).

We now define property $\mathcal{B}$, for a multigraph G as for the graph case, namely

$$\mathcal{B}(G) = \{\text{maxr } w_G(\mathcal{H}(G))) = \mathcal{O}(1)\}. \tag{2.11}$$

As in the graph case, $\mathcal{B}(G)$ states that no copy of $\mathcal{H}$ in $G$ is contained in much more than the average number of $\mathcal{H}$-factors, for a copy of $\mathcal{H}$.

We define $\mathcal{R}(p)$, for the most part, in the same manner as for a graph, with two parts to the definition. For the first part, we use almost the same set-up. We have $G$, our random multigraph and $V$ its vertex set, and given $A \subseteq V(\mathcal{H})$, $E' \subseteq E(\mathcal{H}) \backslash E(\mathcal{H}[A])$, an injection $\psi$, from $A$ to $V$ (mapping vertices to the correct partition set of $V$ corresponding to their location in $\mathcal{H}$). We let $X(G)$ be the number of injections $\phi : V(\mathcal{H}) \to V$ with

$$\phi \equiv \psi \text{ on A} \tag{2.12}$$

and

$$xy \in E' \Rightarrow \phi(x)\phi(y) \in E(G).$$

We can write $X(\mathcal{H}(n,p))$ in an obvious way, as a polynomial in variables $t_e = \mathbf{1}_{\{e\in E(\mathcal{H}(n,p))\}}$, for $e$, an edge in the complete form of our multigraph:

$$X(\mathcal{H}(n,p)) = q(t) = \sum_{\phi} t_{\phi(E')},$$

where $t$ is the indicator function for edges of the multigraph, and the sum is over all injections $\phi$ satisfying (2.12).

As in the original paper, we have that this function $q(t)$, is multilinear, $\mathcal{O}(1)$-normal and homogeneous of degree $d = |E'|$. We use the same definition for $\mathbb{E}^*$;

$$\mathbb{E}^* = \max\{\mathbb{E}_L q : |L| < d\}. \tag{2.13}$$

We also use the same definition for $D(p)$, using it as the expected number of copies of $\mathcal{H}$ in $\mathcal{H}(n,p)$ using a given vertex $x \in V$, while $D(x,G)$ is the actual number of copies containing $x$ in $G$. It is clear that

$$D(p) = (n/v_H)^{k_\mathcal{H}-1}p^{h_\mathcal{H}} = \Theta(n^{k_\mathcal{H}-1}p^{h_\mathcal{H}}).$$

We are now ready to define $\mathcal{R}(p)$, which is identical to the graph formulation, not taking into account our slight changes to the above notation.

**Definition 2.4 (Condition $\mathcal{R}$)**

We say that a random multigraph $G$ satisfies $\mathcal{R}(p)$ if the following two conditions hold.

(a) For $A$, $E'$ and $\psi$ (and associated notation) as above: if $\mathbb{E}^* = n^{-\Omega(1)}$, then for any $\beta(n) = \omega(1)$, $X(G) < \beta(n)$ for large enough $n$; if $E^* \geq n^{-o(1)}$, then for any fixed $\epsilon > 0$ and large enough $n$, $X(G) < n^\epsilon \mathbb{E}^*$.

(b) For each $x \in V$, $|D(x,G) - D(p)| = o(D(p))$

We can now prove that these conditions give the required bound on the size of $\xi_i$. The proof follows in the same fashion as in the original paper:

**Lemma 2.22**

*For $i \leq T = h_\mathcal{H}(n/k_\mathcal{H})^2 - M$ as defined in Section 2.4, $\mathcal{B}_{i-1}$ and $\mathcal{R}_{i-1}$ (i.e. that $G_{i-1}$ satisfies $\mathcal{B}$ and $\mathcal{R}(p_i)$) imply (2.4)*

Proof Write $w$ for $w_{G_{i-1}}$. We aim to show that $\mathcal{B}_{i-1}$ and $\mathcal{R}_{i-1}$ imply that, for any $K \in \mathcal{H}(G_{i-1})$,

$$w(K)/\Phi(G_{i-1}) = \mathcal{O}(1/D(p_{i-1})).$$

As before, the left hand side of this equation is the fraction of $\mathcal{H}$-factors in $G_{i-1}$ that use $K$, and we prove this result in the same fashion, since we have;

$$\begin{aligned}\Phi(G_{i-1}) &= \tfrac{k_{\mathcal{H}}}{n} w(\mathcal{H}(G_{i-1})) \\ &= \tfrac{k_{\mathcal{H}}}{n} \Omega(|\mathcal{H}(G_{i-1})| \max w(\mathcal{H}(G_{i-1})) \\ &= \Omega(D(p_{i-1})w(K)).\end{aligned}$$

The first line follows, since each $\mathcal{H}$-factor will be counted $n/v_H$ times by summing the $w$ function over all copies of $\mathcal{H}$. The second line follows from applying $\mathcal{B}$, while the third comes from part (b) of $\mathcal{R}_{i-1}$, and noting that $(k_{\mathcal{H}}n/k_{\mathcal{H}})D(p) = k_{\mathcal{H}}|\mathcal{H}(G_{i-1})|$.

We also have, using the same arguments as in the original, that part (a) of $\mathcal{R}_{i-1}$ implies that the number of $K \in \mathcal{H}(G_{i-1})$, containing a given edge $e \in E(G_{i-1})$ is at most $\beta(n)$, satisfying $\beta^{-1}D(p_{i-1}) = \omega(\log n)$. Here we also use that $D(p_{i-1}) = n^{k_{\mathcal{H}}-1}p_{i-1}^{h_{\mathcal{H}}} = \omega(\log n)$, for $i \leq T$. ∎

Lastly we state the required '$p$-version' of $\mathcal{A}_t$:

$$\mathcal{A}(p) = \left\{ \log|\mathcal{F}(G)| > \log|\mathcal{F}_0| - \sum_{i=1}^{t} \gamma_i - \mathcal{O}(n) \right\},$$

where $t = \lceil (1-p)h_{\mathcal{H}}(n/k_{\mathcal{H}})^2 \rceil$. Recall from the end of Theorem 2.11, that our required result will follow from the following lemmas

**Lemma 2.23**

*For $p > \omega(n^{-1/m(H)})$,*

$$\Pr(\mathcal{H}(n,p) \textit{ satisfies } \mathcal{R}(p)) = 1 - n^{-\omega(1)}.$$

**Lemma 2.24**

*For $p > \omega(n^{-1/m(H)})$,*

$$\Pr(\mathcal{H}(n,p) \textit{ satisfies } \mathcal{A}(p)\mathcal{R}(p)\overline{\mathcal{B}}) = n^{-\omega(1)}.$$

We prove these lemmas in the next subsections.

### 2.6.4 Regularity

We now prove Lemma 2.23, i.e. that with probability $1 - n^{-\omega(1)}$, $G = \mathcal{H}(n,p)$ satisfies both parts of the definition of $\mathcal{R}(p)$.

Part (a) follows easily in the same fashion as the original paper. There are only $n^{\mathcal{O}(1)}$ choice for each of $A$, $E'$ and $\psi$, so we simply show that the probability of one of these violating (a) is $n^{-\omega(1)}$. As we noted earlier, we can express the random variable as a polynomial, and since it is homogeneous, multilinear, and $\mathcal{O}(1)$-normal, we can apply the probability results from the concentration chapter of [37] directly, which gives us the result as required.

Proving (b) requires only slightly more adaptation to our scenario. We again express $D(x,G)$ as a polynomial of degree $h_{\mathcal{H}} = |E(\mathcal{H})|$, in the variables $t_e = \mathbf{1}_{e \in E(G)}$, where $e$ belongs to the complete form of our random multigraph. Therefore we have,

$$D(x,G) = f(t) := \sum \{t_K : K \in \mathcal{H}_0(x)\},$$

where $\mathcal{H}_0(x)$ is the set of copies of of $\mathcal{H}$, containing $x$, in the complete multigraph, and $t_K = \prod_{e \in K} t_e$.

As noted earlier, we have,

$$\mathbb{E}f = (n/k_{\mathcal{H}})^{k_{\mathcal{H}}-1} p^{h_{\mathcal{H}}} = \Theta(n^{k_{\mathcal{H}}-1} p^{h_{\mathcal{H}}}) = \omega(\log n).$$

We aim to use one of the concentration results from [37], namely Theorem 2.17, which requires the above and that $\max_{1 \le j \le d-1} \mathbb{E}_j f \le n^{-\epsilon} \mathbb{E}f$. For $L$ a subset of the edges of the complete multigraph, with $1 \le |L| = l < h_{\mathcal{H}}$, we have

$$\mathbb{E}_L f = p^{h_{\mathcal{H}}-l} N(L),$$

where $N(L)$ is the number of $K \in \mathcal{H}_0(x)$ with $L \subseteq E(K)$. Let $I = V(L) \cup \{x\}$, $V(L)$ being the set of vertices incident to the edges of $L$, and $k'_{\mathcal{H}} = |I|$. Then $N(L) = \Theta(n^{k_{\mathcal{H}}-k'_{\mathcal{H}}})$ if the graph $\mathcal{H}' := (I,L)$ is isomorphic to a subgraph of $\mathcal{H}$, and zero otherwise. Therefore we have, recalling that $p > n^{-1/m(H)}$ (or $p > n^{-1/m(\mathcal{H})+\varepsilon}$

for a general multigraph) and $d(H) = e_H/(v_H - 1)$,

$$\begin{aligned} \mathbb{E}f/\mathbb{E}_L f &= \Omega(n^{k'_{\mathcal{H}}-1}p^l) = \Omega(n^{[(k'_{\mathcal{H}}-1)/l-1/m(H)]l}) \\ &= \Omega(n^{[1/d(\mathcal{H}')-1/m(H)]l}) = n^{\Omega(1)} \end{aligned} .$$

Using the fact that $\mathcal{H}$ contains no subgraphs of density $m(H)$ (or denser). We therefore have the required conditions to use the concentration theorem and the result follows, which provides us with part (b) of $\mathcal{R}$ as required.

### 2.6.5 Proof of Lemma 2.24

We now begin the proof of Lemma 2.24, continuing in the same vein as the original paper, we will prove that $\mathcal{B}$ is satisfied, using an auxiliary event, $\mathcal{C}$. Most of these results follow in an identical manner to the original, but with small conditions on the choice of sets, and differing constant powers in the equations (largely from use of $k_{\mathcal{H}}$ rather than $v_H$, a difference which is more pronounced in the collapsed form of the theorems, presented in [29]). We include these modified results for completeness.

We write $\mathcal{V}_0$ to be the collection of $k_{\mathcal{H}}$-sets of $V = V(\mathcal{H}(n,p))$, with a single vertex from each partition set of $\mathcal{H}(n,p)$. For a set $Y \subseteq V$, with $|Y| \leq k_{\mathcal{H}}$ and at most one vertex from each partition set, we write $\mathcal{V}_0(Y)$ for the set $\{Z \in \mathcal{V}_0 : Z \supseteq Y\}$. We then extend our earlier weight function $w = w_{\mathcal{H}(n,p)}$ to these sets $Y$, by setting

$$w(Y) = \sum \{w(Z) : Z \in \mathcal{V}_0(Y)\}.$$

We define our new property $\mathcal{C}$ for $\mathcal{H}(n,p)$ as follows: $\mathcal{H}(n,p)$ satisfies $\mathcal{C}$ if for all such $Y$, as defined above, with $|Y| = k_{\mathcal{H}} - 1$,

$$\max w(\mathcal{V}_0(Y)) \leq \max\{n^{-2(k_{\mathcal{H}}-1)}\Phi(\mathcal{H}(n,p)), 2\text{med } w(\mathcal{V}_0(Y)\}$$

We will then prove Lemma 2.24, by proving the two following results.

**Lemma 2.25**

$\Pr(\mathcal{A}\mathcal{R}\overline{\mathcal{C}}) = n^{-\omega(1)}$.

**Lemma 2.26**

$\Pr(\mathcal{R}\mathcal{C}\overline{\mathcal{B}}) = n^{-\omega(1)}$.

We have already demonstrated the first part of the proof of Lemma 2.25 in Section 2.5.2. For clarity, to follow the arguments of the original paper, we will first prove Lemma 2.26, before returning to Lemma 2.25 in the next Section. We must firstly show that

$$|\{K \in \mathcal{V}_0 : w(K) \geq \delta \max w(\mathcal{V}_0)\}| = \Omega(|\mathcal{V}_0|) \tag{2.14}$$

Noting that $|\mathcal{V}_0| = \Omega(n^{k_\mathcal{H}})$, and that this implies $\max rw(\mathcal{V}_0) = \mathcal{O}(1)$. We now need another small modification of a lemma from the graph case. We let $\psi(X) = \max w(\mathcal{V}_0(X))$ and let $B$ be a positive number, recall that $V$ is the vertex set of $\mathcal{H}(n,p)$, and is of size $n$, with $k_\mathcal{H}$ partition sets of size $n/k_\mathcal{H}$.

**Lemma 2.27**

*Suppose that for each $Y \subseteq V$, satisfying $|Y| = k_\mathcal{H} - 1$ and with at most one vertex from each partition set of $V$, and $\psi(Y) \geq B$ we have*

$$\left|\left\{Z \in \mathcal{V}_0(Y) : w(Z) \geq \frac{1}{2}\psi(Y)\right\}\right| \geq \frac{n/k_\mathcal{H} - k_\mathcal{H}}{2}.$$

*Then for any $X \subseteq V$ with $|X| = k_\mathcal{H} - i$, at most one vertex from each partition set and $\psi(X) \geq 2^{i-1}B$, we have*

$$\left|\left\{Z \in \mathcal{V}_0(X) : w(X) \geq \frac{1}{2^i}\psi(X)\right\}\right| \geq \left(\frac{n/k_\mathcal{H} - k_\mathcal{H}}{2}\right)^i \frac{1}{(i-1)!}. \tag{2.15}$$

Proof We write $N_i$ for the right hand side of the above equation, and proceed by induction on $i$. The case $i = 1$ is the hypothesis of the lemma. We then assume $X$ as stated, and choose $Z \in \mathcal{V}_0(X)$ with $w(Z) = \psi(X)$ (i.e. $Z$ such that $w$ is maximal). We let $y \in Z \backslash X$ and $Y = X \cup \{y\}$. We then have that $|Y| = k_\mathcal{H} - (i-1)$ and that $\psi(Y) = \psi(X) \geq 2^{i-1}B (\geq 2^{i-2}B)$, and so, by the inductive hypothesis there are at least $N_{i-1}$ sets $Z' \in \mathcal{V}_0(Y)$ with $w(Z') \geq 2^{-(i-1)}\psi(Y)$. For each such $Z'$, $Z' \backslash \{y\}$ is a $(k_\mathcal{H} - 1)$-subset of $V$ with $\psi(Z' \backslash \{y\}) \geq w(Z') \geq B$. So then for each such $Z'$, there are at least $(n/v_H - k_\mathcal{H})/2$ sets $Z'' \in \mathcal{V}_0(Z' \backslash \{y\})$ with

$$w(Z'') \geq \psi(Z' \backslash \{y\})/2 \geq 2^{-i}\psi(X).$$

■

Therefore the number of such pairs, $(Z, Z'')$ is at least $N_{i-1}(n/v_H - k_\mathcal{H})/2$. Equally, for each $Z''$, each corresponding $Z'$ is $Z'' \backslash \{u\} \cup \{y\}$ for some $u \in Z'' \backslash (X \cup \{y\})$.

Therefore the number of such $Z'$ is at most $i-1$, providing our factorial term. This completes the proof.

We now continue the proof of Lemma 2.26. We set $\delta = 2^{k_{\mathcal{H}}}$ and then $\mathcal{C}$ implies the hypothesis of the above lemma, with

$$B = (2n)^{-(k_{\mathcal{H}}-1)}\Phi(\mathcal{H}(n,p)).$$

We also clearly have that

$$\psi(\emptyset) \geq n^{-(k_{\mathcal{H}}-1)}\Phi(\mathcal{H}(n,p)) = 2^{k_{\mathcal{H}}-1}B.$$

We also set $\gamma = (2^{k_{\mathcal{H}}+1}(k_{\mathcal{H}}-1)!)^{-1}$, and using (2.15), we now have

$$|\{K \in \mathcal{V}_0 : w(K) \geq \delta \max w(\mathcal{V}_0)\}| > \gamma n^{k_{\mathcal{H}}}.$$

We let $J$ be the largest power of 2, not exceeding $\max w$ and

$$\mathcal{Z} = \{Z \in \mathcal{V}_0 : w(Z) > \delta J\}.$$

In a sense, $\mathcal{Z}$ can be thought of as the vertex sets of size $k_{\mathcal{H}}$, whose complement has a relatively large number of factors. For any set $X \subseteq V$ with $|X| \leq k_{\mathcal{H}}$, with at most a single vertex from each partition set, let $\mathcal{Z}(X) = \{Z \in \mathcal{Z} : X \subset Z\}$, and say such a set $X$ is *good* if $|\mathcal{Z}(X)| > \gamma n^{k_{\mathcal{H}}-|X|}$. In particular we know that the empty set is *good*. We then fix an ordering $a_1, \dots, a_{k_{\mathcal{H}}}$ of $V(\mathcal{H})$. For distinct vertices, $x_1, \dots, x_r \in \mathcal{H}(n,p)$, we define $S(x_1, \dots, x_r)$ to be the collection of copies $\phi$ of $\mathcal{H}$ in the complete multigraph $KM_n$, for which

$$\phi(a_i) = x_i \text{ for } i \in [r],$$

$$\phi(a_i)\phi(a_j) \in E(\mathcal{H}(n,p)) \text{ whenever } i,j \geq r \text{ and } a_ia_j \in E(\mathcal{H})$$

and that $\phi(V(\mathcal{H})) \in \mathcal{Z}$.

For each $r \in \{0, \dots, k_{\mathcal{H}}\}$ let $N_r = N(a_r) \cap \{a_{r+1}, \dots, a_{k_{\mathcal{H}}}\}$, and $d_r = |N_r|$, where $N(a_r)$, means the neighbourhood of $a_r$ (in $\mathcal{H}$). We now let $\mathcal{Y}(x_1, \dots, x_r)$ be the event

$$\{|S(x_1, \dots, x_r)| = \Omega(p^{d_r+\dots+d_{k_{\mathcal{H}}-1}}n^{k_{\mathcal{H}}-r})\}.$$

Note that in particular we have, $d_{k_{\mathcal{H}}} = 0$, and, recalling that $\mathcal{H}(G)$, is the set of copies of $\mathcal{H}$ in $\mathcal{H}(n,p)$,

$$S(\emptyset) = \{\phi \in \mathcal{H}(G) : w(\phi(V(\mathcal{H}))) > \delta J\}$$

and that $\mathcal{Y}(\emptyset)$ is the event

$$\{|S(\emptyset)| = \Omega(p^{h_{\mathcal{H}}} n^{k_{\mathcal{H}}})\}.$$

Then, for $v_1, \ldots, x_r \in V$, and from distinct partition sets, let $Q(x_1, \ldots, x_r)$ be the event

$$\{\{x_1, \ldots, x_r\} \text{ is good}\} \wedge \overline{\mathcal{Y}}(x_1, \ldots, x_r)\}.$$

Since we have shown that $\mathcal{C}$ implies that the empty set is good, we have that $\overline{\mathcal{B}}\mathcal{C} \subseteq Q(\emptyset)$ and therefore, we simply require to show that

$$\Pr(\mathcal{R}\mathcal{Q}(\emptyset)) = n^{-\omega(1)},$$

to prove Lemma 2.26, we continue (as always, in the same fashion as for the graph case), by proving a slightly more general argument for induction purposes, namely, that for any choice of $r$ and vertices $x_1, \ldots, x_r$,

$$\Pr(\mathcal{R}\mathcal{Q}(x_1, \ldots, x_r)) = n^{-\omega(1)}. \tag{2.16}$$

Our induction is on $k_{\mathcal{H}} - r$, with our initial step $r = k_{\mathcal{H}}$, trivially following since the definition of being good for subsets of size $k_{\mathcal{H}}$ is to belong to $\mathcal{Z}$. For general $r < k_{\mathcal{H}}$, we set $X = \{x_1, \ldots, x_r\}$. and then let $\mathcal{P}$ be the event

$$\{y \in V \backslash X, X \cup \{y\} \text{ good } \Rightarrow \mathcal{Y}(x_1, \ldots, x_r, y)\}.$$

By the inductive hypothesis we know that $\Pr(\mathcal{R}\overline{\mathcal{P}}) = n^{-\omega(1)}$, so we only need to show

$$\Pr(\mathcal{R}\mathcal{P}Q(x_1, \ldots, x_r)) < n^{-\omega(1)}.$$

We also note that if $X$ is good then,

$$|y : X \cup \{y\} \text{ good}| = \Omega(n). \tag{2.17}$$

To ensure that the edges between $x_r$ and $V\backslash X$ are independent of the initial conditioning, we use a relaxed form of $\mathcal{R}$, $\mathcal{R}_X$, which we say is satisfied if it satisfies part (a) of $\mathcal{R}$, whenever $A = \{a_1, \ldots, a_r\}$, $\psi(a_i) = x_i$ $(i \in [r])$ and $E' \subseteq (\mathcal{H} - A)$.

As for the graph case, if $\mathcal{RP} \wedge \{X \text{ good}\}$ holds, but $\mathcal{Y}(x_1, \ldots, x_r)$ does not, then there must be some $J = 2^{k_\mathcal{H}}$, with $k_\mathcal{H}$ an integer not exceeding $n \log n$ (the magnitude of the log of the number of $\mathcal{H}$-factors in the complete multigraph), such that with $\mathcal{Z}$ good, we have the following, (noting that throughout this chapter, wherever we choose sets of vertices from $V - X$, we choose them from partition sets that do not contain vertices of $X$),

(a) $\mathcal{R}_X$ holds;

(b) There are at least $\Omega(n)$ $y$'s in $V\backslash X$ for which we have $\mathcal{Y}(x_1, \ldots, x_r, y)$ (by (2.17)), and lastly,

(c) $\mathcal{Y}(x_1, \ldots, x_r)$ does not hold.

We note, that for a given $J$, the first two properties, depend only on $G' := G - X$. Since the number of possibilities for $J$ is at most $n \log n$, it is enough to show that for any $J$ and $G'$ satisfying (a) and (b) (with respect to $J$),

$$\Pr(\overline{\mathcal{Y}}(x_1, \ldots, x_r)|G') = n^{-\omega(1)}.$$

Given this fixed $G'$, we can express $|S(x_1, \ldots, x_r)|$ as a multilinear polynomial in terms of the indicator variable for edges between $x_r$ and other vertices of $\mathcal{H}(n,p)$, i.e.

$$t_u := \mathbf{1}_{\{x_r u \in E(\mathcal{H}(n,p))\}} \; u \in V\backslash X.$$

giving the polynomial,

$$|S(x_1, \ldots, x_r)| = g(t) := \sum_U \alpha_U t_U,$$

where $U$ ranges over $d_r$ subsets of $V\backslash X$, with the vertices taken from the correct partition sets, corresponding to the edges from $a_i = \phi^{-1}(x_i)$ in $\mathcal{H}$, and $\alpha_U$ is the

number of copies $\psi$ of $K := \mathcal{H} - \{a_1, \ldots, a_r\}$ in $G'$ with the induced subgraph,

$$\psi(N_r) = U$$

and

$$\psi(\{a_{r+1}, \ldots, a_{k_\mathcal{H}}\}) \cup X \in \mathcal{Z}.$$

We now apply a concentration result from Section 2.6.1, namely Theorem 2.17. To apply this, we require a normal polynomial, so we normalise, and consider

$$f(t) = \alpha^{-1} g(t),$$

where $\alpha$ is the maximum of the $\alpha_U$'s. The hypothesis requires that $\mathbb{E}f = \omega(\log n)$ and $\max_{1 \le j \le d-1} \mathbb{E}_j f \le n^{-\epsilon} \mathbb{E}f$, and will allow us to say that it is close to expectation. We rewrite:

$$g(t) = \sum_{y \in V \setminus X} \sum \{t_{\phi(N_r)} : \phi \in S(x_1, \ldots, x_r, y)\}.$$

Since we know that the indicator variables, $t_u,\ u \in V \setminus X$, are independent of $G'$, which determines our sets $S(x_1, \ldots, x_r, y)$, and using property (b) our situation, we have

$$\mathbb{E}g = p^{d_r} \sum_{y \in V \setminus X} |S(x_1, \ldots, x_r, y)| = \Omega(p^{d_r + \cdots + d_{k_\mathcal{H}-1}} n^{k_\mathcal{H} - r}). \tag{2.18}$$

Noting that if $d_r = 0$, then there are no random edges to consider, and we have that $|S(x_1, \ldots, x_r)|$ will equal

$$\sum_{y \in V \setminus X} |S(x_1, \ldots, x_r, y)| = \Omega(p^{d_r + \cdots + d_{k_\mathcal{H}-1}} n^{k_\mathcal{H} - r}),$$

as required, we will now assume that $d_r > 0$.

We now set $\mathcal{H}' = \mathcal{H} - \{a_1, \ldots, a_{r-1}\}$, and so $d_r + \cdots + d_{k_\mathcal{H}-1} = e(\mathcal{H}')$ and that $k_\mathcal{H} - r = v(\mathcal{H}') - 1$. This gives us that the right hand side of the expectation of $g$, is $\Omega(p^{e(\mathcal{H}')} n^{v(\mathcal{H}')-1})$. Using that $p > n^{-1/m(H)}$ and that $\mathcal{H}$ contains no subgraphs of density $m(H)$, we have

$$\mathbb{E}g = \begin{cases} \omega(\log n) & \text{if } r = 1 \\ n^{\Omega(1)} & \text{if } r > 1. \end{cases}$$

We will now show that for the normalised polynomial $f(t)$, that

$$\mathbb{E}f = \omega(\log n) \tag{2.19}$$

and

$$\max\{\mathbb{E}_T f : T \subseteq V\backslash X, 0 < |T| < d_r\} = n^{-\Omega(1)}\mathbb{E}f. \tag{2.20}$$

With these conditions, the concentration theorem will tell us that $f$, and therefore $g$ is close to its expectation, which implies $\overline{\mathcal{Y}}$, as required. To prove the two conditions, we will find it easier to consider the partial derivatives of $g$ rather than $f$. We use $t_e = \mathbf{1}_{e\in E(\mathcal{H}(n,p))}$, $t_S = \prod_{e\in S} t_e$ and $t = (t_e : e \in E(KM_{V\backslash X}))$, where $KM_{V\backslash X}$ is the multigraph induced by the 'complete' multigraph $KM_n$ on the the vertex set $V\backslash X$.

Since we are only interested in establishing upper bounds on the partial derivatives of $g$, we may now disregard the second requirement on $\alpha_U$, namely

$$\psi(\{a_{r+1}, \ldots, a_{k_{\mathcal{H}}}\}) \cup X \in \mathcal{Z}.$$

Therefore we are left with

$$p^{-(d_r-l)}\mathbb{E}_T g \leq \tau(t) := \sum_{\phi} t_{\phi(E(K))},$$

where we sum over $\phi$, injections such that

$$\phi : V(K) \to V\backslash X \text{ with } \phi(N_r) \supseteq T.$$

We set $E^*$, as before to be $E^* = \max\{\mathbb{E}_L\tau : L \subseteq E(KM_{V\backslash X}), |L| < |E(K)|\}$. We will show that there is a positive constant $\varepsilon$ (depending only on $\mathcal{H}$), such that (for large enough $n$),

$$p^{d_r-l}\mathbb{E}^* < n^{-\epsilon}\mathbb{E}g. \tag{2.21}$$

This will give us the two requirements for our concentration theorem, as follows. To prove (2.19), we need to show that

$$\alpha^{-1}\mathbb{E}g = \omega(\log n).$$

We apply (2.21) with $T = U$, a $d_r$-subset of $V/X$. We consider the two possible conditions of $\mathcal{R}_X$, which gives two separate cases, firstly if $\mathbb{E}g \geq n^{\varepsilon/2}$, (i.e. $\mathbb{E}^* \geq n^{-o(1)}$) then $\mathcal{R}_X$ tells us that

$$\alpha_U = \mathbb{E}_U g < n^{\varepsilon/4} \max\{1, \mathbb{E}^*\} \leq n^{-\varepsilon/4}\mathbb{E}f.$$

In the other case we are left with $\mathbb{E}^* < n^{-\varepsilon/2}$, and hence $\mathbb{E}^* = n^{-\Omega(1)}$. We know that $\mathbb{E}g = \omega(\log n)$, and hence we choose our $\beta(n) = \omega(1)$ (from $\mathcal{R}$) such that $\beta(n)^{-1}\mathbb{E}g = \omega(\log n)$. Since this does not depend on our choice of $U$, we have (2.19) as required.

To prove the other requirement, we need $\mathbb{E}_T g = n^{-\Omega(1)}\mathbb{E}g$ for any $T$, as we defined in (2.20). We again apply (2.21), noting that we can decrease the $\varepsilon$ without violating the equation, and use this observation to assume that $\varepsilon < 1/m(H)$. This gives us

$$\mathbb{E}_T g < p^{d_r - l} n^{\varepsilon/2} \max\{1, \mathbb{E}^*\} \leq n^{-\varepsilon/2}\mathbb{E}g$$

as we required.

We now return to prove (2.21). We fix $L \subseteq E(KM_{V\setminus X})$ and let $h_l = |E(K)| - |L|$, (recalling that $K = \mathcal{H} - \{a_1, \ldots, a_r\}$). We know that $\mathbb{E}_L \tau = p^{h_l} N_L$, where again $N_L$ is the number of $\phi$, as defined just before (2.21), with $\phi(E(K)) \supseteq L$. Each $\phi$ satisfies $\phi(V(K)) \supseteq I := T \cup V(L)$, where as earlier, $V(L) \supseteq V\setminus X$ is the set of vertices incident to the edges of $L$. We let $I = \{i_1, \ldots, i_s\}$, then we have $N_L = \sum N_L(b_1, \ldots, b_s)$, where $(b_1 \ldots, b_s)$ range over $s$-tuples of distinct elements of $V(K)$ and we sum over the number of $\phi$'s as above, with $\phi(b_j) = i_j$ for each $1 \leq j \leq s$. We only have $\mathcal{O}(1)$ choices for the $b_j$'s and hence the result will follow if we can show that for any such choice,

$$p^{d_r - l + h_l} N_L(b_1, \ldots, b_s) = n^{-\Omega(1)}\mathbb{E}g.$$

Given a fixed choice of $b_i$'s, let $\mathcal{H}'' = \mathcal{H}[\{a_r, b_1, \ldots, b_s\}]$. We know that $N_L(b_1, \ldots, b_s) < n^{k_\mathcal{H} - r - s} = n^{k_\mathcal{H} - r - (v(\mathcal{H}'') - 1)}$, and that $h_l \geq l + d_{r+1} + \cdots + d_{k_\mathcal{H} - 1} - e(\mathcal{H}'')$, since $|E(K)| = d_{r+1} + \cdots + d_{k_\mathcal{H} - 1}$ and $E(\mathcal{H}'')$ contains $\phi^{-1}(L)$ and at least $l$ edges joining

$a_r$ to $V(K)$. Therefore we have

$$p^{d_r-l-h_l}N_L(b_1,\dots,b_s) < p^{d_r+\cdots+d_{k_{\mathcal{H}}-1}}n^{k_{\mathcal{H}}-r}[n^{v(\mathcal{H}'')-1}p^{e(H'')}]^{-1}.$$

Noting that since $p > n^{-1/m(H)}$ and that $\mathcal{H}$ contains no subgraphs of density $m(H)$, we have that the expression in the square brackets is $n^{\Omega(1)}$. Combined with our earlier bound on $\mathbb{E}g$, from (2.18), we have the bound required above and hence (2.21), completing the Lemma.

### 2.6.6 Proof of Lemma 2.25

We now begin the final step of the proof, namely that $\Pr(\mathcal{A}\mathcal{R}\overline{\mathcal{C}}) = n^{-\omega(1)}$. We maintain our use of notation from the previous section with $G = \mathcal{H}(n,p)$, and $\mathcal{R}$ and $\mathcal{A}$. In Section 2.5.2, we have shown that $\mathcal{A}\mathcal{R}\overline{\mathcal{C}}$ results in the following;

There exist $x$ and $y$; vertices in the same partition set of $G = \mathcal{H}(n,p)$ and a set $Y$ with a single vertex from each of the remaining partition sets with $R = Y \cup \{x\}$ and $S = Y \cup \{y\}$, such that there exist, $a, b \in range(w_y)(= range(w'))$, such that $J := w_y^{-1}([a,b])$ satisfies

$$|J| > \Omega(|\mathcal{H}(y, G-R)),$$

and

$$w_y(J) > 0.7w_y(\mathcal{H}(y,G-R)) = 0.7w(R).$$

With, $J' = w_x^{-1}([a,b])$, we also have,

$$w_y(J) > 0.7w(R),$$

and

$$w_x(J') \le w(S) < 0.5w(R).$$

All that remains is to show that the probability of this event is $n^{-\omega(1)}$.

Once again, we can express $w_y(J)$ and $w_x(J')$ as evaluations of a multi-linear polynomial in variables $\{t_u : u \in W\}$, once we have conditioned on the value of $G[W]$, in the following way. Given a set $U \subseteq W$ with $|U| \le k_{\mathcal{H}} - 1$ and at most one vertex from each partition set, let $G_U^*$ be the graph obtained from $G[W]$ by

adjoining a vertex $w^*$ say, with neighbourhood $U$. We let $\mathcal{K}_U$ be the set of copies of $\mathcal{H}$ in $G^*_U$, containing $\{w^*u : u \in U\}$, and

$$\alpha_U = \sum \{w'(V(K)\backslash\{w^*\}) : K \in \mathcal{K}_U, w'(V(K)\backslash\{w^*\}) \in [a,b]\}.$$

Our polynomial now becomes,

$$g(t) = \sum_{U \subseteq W} \alpha_U t_U,$$

and $w_y(J)$ and $w_x(J')$ are simply $g$ evaluated at the point $t' := \mathbf{1}_{\{z \in W : yz \in E(G)\}}$ and $t'' := \mathbf{1}_{\{z \in W : xz \in E(G)\}}$.

$\mathcal{R}$ tells us that $D_G(y) = \Theta(n^{k_\mathcal{H}-1}p^{h_\mathcal{H}})$ and therefore, considering that at most $o(n^{k_\mathcal{H}-1}p^{h_\mathcal{H}})$ copies of $H$ lie in $G$ and contain $y$ and meet $R$, we have $|\mathcal{H}(y, G-R)| = \Theta(n^{k_\mathcal{H}-1}p^{h_\mathcal{H}}) = \omega(\log n)$.

We know from our conditioning of $J$, that it satisfies

$$|J| = \Theta(n^{k_\mathcal{H}-1}p^{h_\mathcal{H}}),$$

and

$$w_y(J) = \Theta(bn^{k_\mathcal{H}-1}p^{h_\mathcal{H}}). \tag{2.22}$$

We now apply a concentration result, namely Corollary 2.20. To apply it, we require a bound on the expectation of $f$, namely, that if $Ef \leq A$, where

$$A \geq \omega(\log n) + n^\epsilon \max_{l \neq \emptyset} E'_L f.$$

then the corollary gives us $\Pr(f > (1+\epsilon)A) = n^{-\omega(1)}$. We fix $T \subseteq W$, with $|T| = l < k_\mathcal{H}$, and as always, each vertex from a different partition set. For $d = l, \ldots, k_\mathcal{H} - 1,$, and $t$ as $t = (t_e : e \in E(KM_W))$, we consider the polynomial,

$$h_d(t) = \sum_z \sum_\phi t_{\phi(E(\mathcal{H}-z))},$$

Where $z$ ranges over vertices of $\mathcal{H}$ of degree $d$ and $\phi$ over the injections $V(H)\backslash\{z\} \to W$ with $\phi(N_z) \supseteq T$. Then we know that

$$\alpha_T \leq b\ h_l(t).$$

Since $h_l(t)$ is the number of such sets, and we know that their values are bounded by $b$, this also gives us

$$\mathbb{E}'_T g \leq b \sum_{d>l} p^{d-l} h_d(t),$$

where $\mathbb{E}'_T$ is the non-constant part of the partial derivative.

We let

$$\mathbb{E}^*_d = \max\{E_L h_d : L \subseteq E(KM_W), |L| < h_{\mathcal{H}} - d\}.$$

As in the previous section, similarly to (2.21) we can assert that there is a positive constant $\epsilon$ dependent only on $\mathcal{H}$, such that, for each $d$,

$$p^{d-l}\mathbb{E}^*_d < n^{-\epsilon} n^{k_{\mathcal{H}}-1} p^{h_{\mathcal{H}}}.$$

This follows from the proof of (2.21) in the previous section, by noting that in our definition of $h_d$, there are only finitely many $z$, and the inner sum, is bounded by the polynomial $\tau$, used in (2.21), with $r = 1$, $a_1 = z$ and hence $d_r = d$.

We are now able to apply the concentration results we require. We consider the polynomial $f = \alpha^{-1} g$, where $\alpha = \max_U \alpha_U$. Then we have, using (2.22),

$$f(t') = \alpha^{-1} w_y(J) = \Theta(\alpha^{-1} b n^{k_{\mathcal{H}}-1} p^{h_{\mathcal{H}}})$$

and using the same arguments as we used at the end of the previous section, we can show

$$f(t') = \omega(\log n)$$

and

$$\max\{\mathbb{E}'_T f : T \subseteq W, T \neq \emptyset\} = n^{-\Omega(1)} f(t').$$

For the final step, we first summarise that we have shown $\mathcal{AR}\overline{\mathcal{C}}$ implies that there exist $Y,\ x,\ y$ and $a,\ b \in range(w')$ for which we have the above two conditions, and

$$f(t'') < 0.8 f(t').$$

However, we know that for any given choice of $Y,\ x,\ y,\ a$, and $b$, $f$ depends only on $G[W]$. Equally, if $G[W]$ is fixed, then $t$ and $t'$ are independent random variables, with 'law' $Bin(W, p)$ (i.e. $t = (t_w : w \in W)$ has 'law $Bin(W, p)$' if each $t_w$ is an

independent Bernoulli with mean $p$.) We finish with the final claim of [37], which can be applied directly to our result here, without generalisation, and completes the proof.

**Claim 1**

*For any $\epsilon > 0$ and $d$ the following holds. If $f$ is a multilinear, normal polynomial of degree at most $d$ in $n$ variables, $\zeta(n) = \omega(\log n)$, and $t', t''$ are independent , each with law $Bin([n], p)$, then*

$$\Pr(f(t') > \max\{\zeta(n), n^{\epsilon} \max_{T \neq \emptyset} E_T^{'} f, (1+\epsilon) f(t'')\}) = n^{-\epsilon}.$$

*Because there are only polynomially many possibilities for $Y, x, y, a$ and $b$, this gives us Lemma 2.25.*

*Proof of claim.* Set

$$A = \frac{1}{2} \max \left\{ \zeta(n), n^{\epsilon} \max_{T \neq \emptyset} E_T^{'} f. \right\}.$$

If $\mathbb{E}f \leq A$ then Corollary 2.20 gives

$$\Pr(f(t') > A) = n^{-\omega(1)};$$

otherwise, by Theorem 2.19,

$$Pr(f(t') > (1+\epsilon) f(t'')) < \Pr(\max\{|f(t') - \mathbb{E}f|, |f(t'') - \mathbb{E}f|\} > (\epsilon/3)\mathbb{E}f) = n^{-\omega(1)}.$$

## 2.7 Theorems 2.2 and 2.3 (Factors in directed graphs)

We note that in our proof of Theorem 2.2 we do not require the collapsed graph to be a multigraph. Throughout, we only require that the graph does not contain any subgraphs of density $m(H)$, and this follows, in our main result, from the vertex collapsing technique, but as in [37], it can also follow from strict balance of $\mathcal{H}$, or equivalently $H$. Since our partitions were fixed only by the partial embedding of subgraphs of density $m(H)$, in the strictly balanced case, we can simply choose our vertex partitions freely, and then continue with the proof.

Lastly, to prove Theorem 2.3, we use Theorem 2.2. Partition the vertices of $G(n,p)$ as usual into $v_H$ equal sets, either as a result of vertex collapsing dense subgraphs, or freely in the strictly balanced case. In the former case, embedding the partial factors of dense directed subgraphs, requires a modification of Theorem 2.4, which follows the same arguments as the original but for directed graphs. We prove this result below, first recalling the definition of $D(n,p)$.

**Definition 2.5 ($D(n,p)$)**

We let $D(n,p)$ be the random directed graph on n vertices, where each unordered pair of vertices is present with probability $p$ independently of each other edge, and is directed with probability $1/2$ either way. This is equivalent to taking $G(n,p)$ and randomly directing it's edges. $D(n,1)$ is therefore a random directing of the complete graph on $n$ vertices.

We begin with a technical lemma.

**Lemma 2.28**

*Fix a directed graph $H$ with $v$ vertices and $e$ edges. Let $X_H$ be the number of copies of $H$ in $D(n,1)$. Then, a.a.s.*

$$X_H = \Theta(n^v).$$

Proof We make use of a martingale concentration result, which can be found as Corollary 2.27 in [36]. We state it below

**Corollary 2.29**

*Let $Z_1, \dots, Z_N$ be independent random variables, with $Z_k$ taking values in a set $\Lambda_k$. Assume that a function $f : \Lambda_1 \times \Lambda_2 \times \cdots \times \Lambda_N \to \mathbb{R}$ satisfies the following Lipschitz condition for some numbers $c_k$:*

*If two vectors $z, z' \in \prod_1^N$ differ only in the kth co-ordinate, then $|f(z) - f(z')| \leq c_k$.*

*Then the random variable $X = f(Z_1, \dots, Z_N)$ satisfies, for any $t \geq 0$,*

$$\mathbb{P}(|X - \mathbb{E}X| \geq t) \leq \exp\left(-\frac{t^2}{2\sum_1^N c_k^2}\right).$$

We apply this result to the complete graph, with our $Z_i$ being the direction of edge $i$, for each $1 \leq i \leq \binom{n}{2}$, and our function $f = X_H$. It is clear that if we change the direction of a single edge, we at most change the value of $f$ by the number of copies of $H$ that edge could have been present in, which is at most $\Theta(\binom{n-2}{v-2}) = \Theta(n^{v-2})$. In other words, our function satisfies the Lipschitz condition with $c_k = cn^{v-2}$ for some constant $c$. We note that the $\mathbb{E}(X_H) = \binom{n}{v} v!/2^v \mathrm{Aut}(H) = \Theta(n^v)$. We aim to show that $X_H$ is within a constant multiple of expectation, and so with $t = \frac{1}{2}\mathbb{E}(X_H)$, corollary 2.29 gives us,

$$\begin{aligned}
\mathbb{P}(|X_H - \mathbb{E}X_H| \geq \frac{1}{2}\mathbb{E}(X_H)) &\leq \exp\left(-\frac{\mathbb{E}(X_H)^2}{2\sum_{i=1}^{\binom{n}{2}}(cn^{v-2})^2}\right) \\
&= \exp\left(-\frac{\mathbb{E}(X_H)^2}{2c^2\binom{n}{2}n^{2v-4}}\right) \\
&\leq \exp\left(-\Theta\left(\frac{n^{2v}}{n^{2v-2}}\right)\right).
\end{aligned}$$

This clearly tends to 0 and hence, we have as $n \to \infty$, that a.a.s. $X_H = \Theta(\mathbb{E}(X_H)) = \Theta(n^v)$ as required. ∎

We also have the following corollary, the proof of which follows in an identical manner to the above.

**Corollary 2.30**

*Fix $H'$, a subgraph of $H$ with $v'$ vertices. Given a copy of $H'$ in $D(n,1)$, the number of copies of $H$ in $D(n,1)$, using that fixed copy as a subgraph is $\Theta(n^{v-v'})$.*

We will also require the following Theorem, again from [36], where it can be found as Theorem 2.18(ii).

**Theorem 2.31**

*Let $\Gamma_p$ be a binomial random set of size $N$, where each element of $\Gamma = [N]$ is present with probability $p$. Let $\mathcal{S}$ be a family of non-empty subsets of $\Gamma$ and for each $A \in \mathcal{S}$ let $I_A$ be the indicator function for the event $\{A \in \Gamma_p\}$. Let $X = \sum_{A\in\mathcal{S}} I_A$, i.e. $X$ is the number of sets in $\mathcal{S}$ contained in $\Gamma_p$. We have the following*

$$\mathbb{P}(X = 0) \leq \exp\left(-\frac{\mathbb{E}(X)^2}{\sum\sum_{A\cap B\neq 0}\mathbb{E}(I_A I_B)}\right) \tag{2.23}$$

We are now ready to tackle our required result, which is a digraph generalised form of Theorem 3.9 from [36].

**Theorem 2.32**

*Let $H$ be a directed graph on $v$ vertices and $e$ edges and $X_H$ be the number of copies of $H$ in $D(n,p)$. Let $\Psi_H = \Psi_H(n,p) = \min\{\mathbb{E}(X_{H'}) : H' \subseteq H, e_{H'} > 0\}$. Then for any sequence $p = p(n) < 1$, we have the following,*

$$\mathbb{P}(D(n,p) \not\supset H) \leq \exp\{-\Theta(\Psi_H)\}.$$

Proof We first direct the complete graph on $n$ vertices, and then we apply Theorem 2.31 to $D(n,p)$ where $\Gamma_p$ is the set of these directed edges from $D(n,1)$, each present with probability $p$. We then let $\mathcal{S}$ be the set of copies of $H$ in $D(n,1)$, and hence $X$ is the number of such copies in $D(n,p)$. We consider the denominator of the exponent of the inequality in Theorem 2.31, this can be reformulated as

$$\sum_{K\subseteq H, e_K>0} \sum_{H',H''\in\mathcal{S}} \sum_{H'\cap H''=K} \mathbb{E}(I_{H'}I_{H''}) = \sum_{K\subseteq H, e_K>0} \sum_{H',H''\in\mathcal{S}} \sum_{H'\cap H''=K} p^{2e-e_K}.$$

By Corollary 2.30, applied to the graph formed by the second copy of $H$, without the $K$ contained in $H'$, in the innermost sum, and by Lemma 2.28 applied to $H$ in the 2nd sum, we have that this is equal to,

$$\sum_{K\subseteq H, e_K>0} \Theta(n^v n^{v-v_K} p^{2e-e_K}) = \sum \Theta(n^{2v} p^{2e} n^{-v_K} p^{-e_K}) = \Theta((\mathbb{E}(X))^2/\Psi_H).$$

Substituting back into (2.23), we have the inequality as required. ∎

In light of this we have the following

**Corollary 2.33**

*For any $\epsilon > 0$ there exists $C > 0$ such that for $p > C(n^{-1/m(H)})$, a.a.s. $D(n,p)$ contains a partial factor covering at least $(1-\epsilon)n$ vertices.*

Proof Suppose false, then there exists a subset of at least $\epsilon n$ vertices that does not a contain a copy of $H$. The probability that this happens is, by union bound and Theorem 2.32, at most,

$$\binom{n}{\lceil \epsilon n \rceil} \mathbb{P}(D(\lceil \epsilon n \rceil, p) \not\supset H) \leq 2^n e^{-c\Psi_H(\lceil \epsilon n \rceil, p)}. \tag{2.24}$$

Where the $c$ in the exponential depends only on $H$ and $\epsilon$. Letting $p_d = p/2$, note that

$$\begin{aligned}\Psi_H &= \min\{\mathbb{E}(X_{H'}) : H' \subseteq H, e_{H'} > 0\} = \min_{H' \subseteq H} n^{v'} p_d^{e'} \\ &\geq \min n^{v'-1} p_d^{e'} = \min(n p_d^{(e'/v'-1)})^{v'-1} \geq \min(n p_d^{m(H)})^{v'-1},\end{aligned}$$

and this implies that for $p > C(n^{-1/m(H)})$, $\Psi_H \to \infty$, and in particular $c\Psi_H(\lceil \epsilon n \rceil, p) > n$, and so the right hand inequality of (2.24) tends to 0, giving us a contradiction, as required. ∎

We can use this result to embed the partial factors of the collapsed dense subgraphs of $H$, as in the undirected case, and continue as before, partitioning $D(n,p)$ into vertex sets corresponding to the vertex of $H$ they are covered by in the partial factors we have embedded.

We again consider the edges between the partition sets, only where they correspond to edges in $H$. Now however, we can consider only the edges that are in the direction we require, which are distributed uniformly and independently at random, but with edge probability $p/2$, discounting those edges in the wrong direction. At this point, the edges of the random digraph we are considering are equivalent to the undirected random graph $\mathcal{H}(n, p/2)$, and we apply Theorem 2.2 directly, providing the directed factor as required.

## 2.8 Conclusion, balanced $H$ and further work

With these results, Conjecture 1 is now proven for both strictly balanced graphs, and all non-vertex-balanced graphs. The methods used here also can be used to prove that the conjecture holds for a wide range of vertex-balanced $H$.

We can prove the threshold for a variety of 'necklace' and related graphs formed of copies of a dense graph linked by a 'supergraph' of equal or lower density.

As an example, in the graph pictured in Figure 2.5, the threshold for finding a copy of this 'triangle necklace' must be at least that of a triangle factor, and Conjecture 1 suggests it should be equal. We prove this by first embedding a triangle

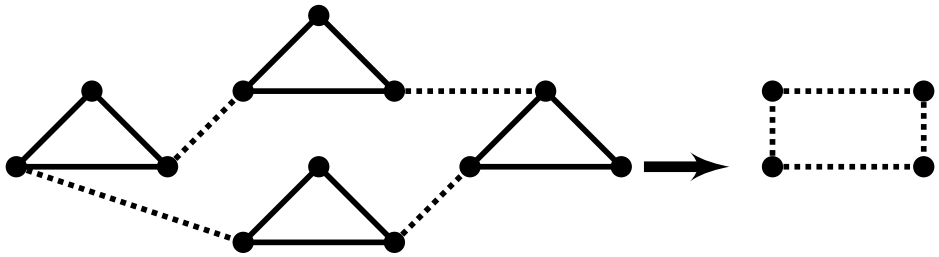


Figure 2.5: A necklace graph.

factor, and then use Theorem 2.2 to embed the dotted edges that form a less dense cycle in the supergraph. Provided the resulting graph or multigraph from the collapsing process is less dense than $m(H)$, we will always be able to apply Theorem 2.2 to find this.

The supergraph cannot ever be denser than $m(H)$, as this implies a subgraph of density greater than $m(H)$, which is a contradiction. So, all that remains to consider are graphs where we will be left with a supergraph of equal density, after collapsing all subgraphs. We know this supergraph must be strictly balanced, or we would have continued to collapse its subgraphs and so we can apply Theorem 2.2. However, the supergraph may have fewer edges than each of the collapsed subgraphs, and certainly has fewer than the original graph. This may force it to require a higher threshold, by a constant power of log to embed, and so will not prove the conjecture.

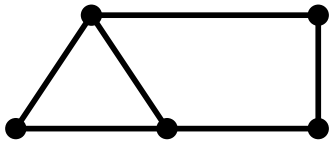


Figure 2.6: A balanced graph.

For graph in Figure 2.6, the conjectured threshold is $n^{-2/3}(\log n)^{1/5}$, but using the methods within this paper, we are only able to embed it at $p = \omega(n^{-2/3}(\log n)^{1/3})$, since collapsing the triangle will leave us with another triangle, requiring the higher log term to embed.

In light of this, we see that we can prove Conjecture 1 for all $H$, except for

those mentioned above, for which we are still within a constant power of log of the conjectured bound.

Formally we have the following. For a vertex-balanced but not strictly balanced $H$ we begin by labelling each vertex with $s_v$ which we recall is defined by

$$s_v = \min\{e(H') : H' \subseteq H, v \in V(H'), d(H') = m(v, H)\}.$$

We now apply the collapsing process, at each stage only collapsing strictly balanced strict subgraphs of $H$ (or subsequently strict subgraphs of the resulting graph after collapse) of density $m(H)$ and choosing the subgraph with the least number of edges each step. Label the newly formed vertex after each step with the smaller of either: the number of edges collapsed into the vertex or the maximum $s_v$ taken over the vertices collapsed to form it. This process terminates when the remaining (multi)graph after collapse is itself strictly balanced and we denote it by $\mathcal{H}$ and define $e_{\mathcal{H}}$ to be the number of edges it contains. Let $S_M = \min\{\max_{i \in V(\mathcal{H})}\{s_i\}, e_{\mathcal{H}}\}$. With these terms defined, we have,

**Theorem 2.34**

*If $H$ is vertex-balanced but not strictly balanced then,*

$$\mathrm{th}_H(n) = \mathcal{O}\left(n^{-1/m(H)} (\log n)^{1/S_M}\right).$$

Proof Consider a subgraph $H'$, which we collapse in the process of generating $\mathcal{H}$. This subgraph has density $m(H)$ and as such we can embed a partial factor of it for values of $p = \omega\left(n^{-1/m(H)}\right)$. Equally, if required we can embed a full, partitioned factor of $H'$ for $p = \omega\left(n^{-1/m(H)} (\log n)^{1/|e(H')|}\right)$. Consider the vertex formed by collapsing $H'$ and a strictly balanced subgraph $H''$ (or graph if the collapsing process has terminated) that contains it. We can now embed a factor of the vertices that both form and have been collapsed into $H''$ by first finding a partial factor of $H'$ and then finding a partitioned factor of $H''$. This requires $p$ to satisfy $p = \omega\left(n^{-1/m(H)} (\log n)^{1/|e(H'')|}\right)$.

The only issue is if all of the other vertices of $H''$ were also formed by the collapse of a subgraph. In this case, we can embed partial factors for all but one of the

subgraphs but will then be required to find a complete factor of the final subgraph on the remaining vertices of $G(n,p)$. We can choose which collapsed subgraph to embed last and hence we choose $H'$ to be the subgraph with maximum number of edges. As such we can embed the collapsed subgraphs at $p = \omega\left(n^{-1/m(H)}(\log n)^{1/|e(H')|}\right)$ and hence the full subgraph can be embedded at $\omega\left(n^{-1/m(H)}(\log n)^{1/\alpha}\right)$, where $\alpha = \min\{|e(H')|, |e(H'')|\}$.

If $H''$ is the graph resulting from the termination of the collapsing process, then we are done, as $\alpha = e_M$ in this case. If not and the process continues, then we can use the same argument as above to iteratively arrive at the same result. Ultimately we will be left with a single strictly balanced final graph, with each vertex $i$ labelled by the value of $s_i$ for which a factor of the vertices collapsed into it can be embedded at $\omega\left(n^{-1/m(H)}(\log n)^{1/s_i}\right)$. In this way we continue as above, partial factors are embedded for all but one vertex and a full factor is embedded for the vertex for which the value of $s_i$ is maximum. Lastly the edges of the final graph $\mathcal{H}$ are embedded which requires $\omega\left(n^{-1/m(H)}(\log n)^{1/e_{\mathcal{H}}}\right)$ and hence the threshold is bounded above by the smaller of these two values where the larger of the inverted values in the log exponent is precisely the definition of $e_M$ as required. ∎

For any $H$, $s_M$ will always satisfy $s_M \leq s(H)$ for the $s(H)$ that appears in the log exponent of Lemma 2.8 and hence since both appear inverted in the exponent, this bound is (as required) at least the size of the lower bound which is conjectured to be the correct threshold. This is clear as $s(H)$ is the maximum over all $s_v$ and $s_M$ is chosen by choosing amongst the smaller of the maximum of values of subsets of $s_v$ and the size of the final graph and so cannot exceed the maximum $s_v$ which is precisely $s(H)$. For many graphs these values will however coincide, as for example in Figure 2.5, after the collapsing process terminates, each vertex will be labelled with a value of 3 and the final supergraph has 4 edges and as such $s(H) = s_M = 3$ as required.

Chapter 3

# *Complexity on Eulerian Circuits*

### 3.0.1 Introduction

An Eulerian circuit of a graph $G$ is a trail (a sequence of distinct edges from $E(G)$; $\{(v_{1,1}, v_{1,2}), (v_{2,1}, v_{2,2}) \dots\}$ with $v_{i,2} = v_{i+1,1}$ for all $i$) in which every edge of $G$ appears exactly once, starting and ending at the same vertex, and is one of the oldest concepts in graph theory.

Euler proved that a necessary condition for a graph to be Eulerian, i.e. to permit an Eulerian circuit, is for it to be connected and that every vertex must have even degree [45] and stated that this condition was also sufficient, but this was not formally proven until 1873 by Carl Hierholzer when it was published posthumously after he had described his work to a colleague before his death [34].

Having defined the Eulerian circuits in this chapter's title, we now move on to introducing complexity. Informally, we call a well defined question with specified inputs, for which there is a yes or no answer, such as, "can the graph $G$ be properly coloured using only $k$ colours?", a decision problem. We define the set P to be the set of such problems, for which we can find a solution in polynomial time, in terms of the size of the problem's inputs (for example, $k$ and the number of edges or vertices in $G$).

By a solution in polynomial time, we mean an algorithm that will deterministically provide a solution to all instances of the problem such that the number of steps taken by this algorithm to solve any instance of the problem is bounded above by a polynomial function in the input parameters of that instance.

A reason why polynomial time algorithms are of interest, is that they scale well with increasing the size of the problem. The advent of computing means that tasks

and calculations that would have been tedious or impossible to compute by hand can be tackled by computers. If a problem has a polynomial time solution, then when applying such an algorithm, we know that there is a limit to how badly the time taken to solve it will scale up as we increase the size of the inputs. Conversely, if all we have at our disposal are exponential time algorithms, increasing the scale of the problem will very quickly lead to solution times that spiral out of control.

If for any possible solution to a decision problem, if it can be verified as a correct solution in polynomial time, then we say that the problem is part of the set NP. The problem of whether P=NP remains open (famously so), but there is a class of problems for which it has been shown that a polynomial time solution for any one of them would allow you to solve any problem in NP in polynomial time (and hence P would equal NP). This class is called NP-Hard, and also includes problems that are not decision problems, such as "what is the minimum number of colours needed for graph $G$ to be properly coloured?". We call a decision problem NP-complete if it lies in both NP and NP-Hard.

The first problem shown to be NP-complete, was the Boolean satisfiability problem, usually referred to as SAT, which was proven by Cook in his seminal paper "The Complexity of Theorem-proving Procedures" [12], which formalised the concepts of polynomial time reduction and NP-completeness. This result was also independently discovered by Levin in 1973 [46]. Following on from these initial results, Karp demonstrated 21 problems that were NP-complete [40], by demonstrating that if they were solvable in polynomial time, the original SAT problem would also become solvable, and hence these new problems must also be NP-complete. This approach is the model for almost all NP-completeness proofs today, namely to demonstrate a "polynomial time reduction" from a solution to the problem to be tackled to a solution to a known NP-complete problem, thus proving that it is also NP-complete.

Given an Eulerian graph, it is well known that an Eulerian circuit can be found in polynomial time. By simply following a trail around the graph, choosing a free edge (one not already used in the trail) at the starting or current vertex whenever

available and following this edge to the next vertex and repeating the process, you will eventually find a closed trail. Supposing this trail is not an Eulerian circuit, some vertex in the trail must have free edges not used in the trail, and since an even number of edges are used by any trail at each vertex, this vertex (and in fact all vertices) will have an even number of free edges remaining. Beginning again at this vertex with free edges remaining, and connecting the new trail to the first will rapidly cover the graph's remaining edges through iteration of this process. This approach is essentially what is known as Hierholzer's algorithm [34].

In sharp contrast to this structure is the Hamiltonian cycle, a closed cycle that visits every vertex of the graph exactly once. Although superficially similar, there are no known simple and easy to check conditions for the existence of a Hamiltonian cycle and finding such a cycle was one of Karp's original 21 NP-complete problems [40].

Finding a Hamiltonian cycle can be seen to be an example of the travelling salesman problem, i.e. finding the shortest cycle in an edge weighted graph, that visits each vertex at least once. By setting the distance between any two adjacent vertices to one, a solution of length $n$ to the travelling salesman problem must correspond to a Hamiltonian cycle.

Finding an Eulerian circuit is a special case of the Chinese Postman, or route inspection, problem, in which the shortest trail, that uses every edge at least once, is sought after in a (usually weighted) graph. In an Eulerian graph, this is always any Eulerian circuit, and in a general undirected graph, finding the minimum is computable in polynomial time [15].

This might lead one to wonder whether any problem related to Eulerian circuits is NP-complete and in fact the Chinese Postman Problem does become so, for graphs that contain both directed and undirected edges, (Listed as ND25 in [28]). This however is for non-Eulerian graphs, looking for the shortest circuit that may use edges several times.

Limiting ourselves to Eulerian circuits, we can see that weighting the edges or

vertices will have no impact, since all Eulerian circuits use each edge and vertex the same number of times, with the only choices in forming an Eulerian circuit are the order in which you follow them. This suggests a number of possible problems to consider to take into account these transitions and we consider two of them in this chapter.

## 3.1 The turn-costed Eulerian circuit problem

The first such problem we consider is that of the turn-costed Eulerian circuit problem (ETCP). We begin with an Eulerian graph $G$ and at each vertex of $G$ we assign turning costs (which we define below), with the aim of finding an Eulerian circuit with either minimum total cost or with total cost less than some fixed value (or whether such a circuit exists for the decision version of the problem). In defining the turning cost at a vertex, we consider the half edges incident to it formed by considering each edge as two half edges, so a loop at $v$ would create two distinct half edges, both at $v$, while each other edge would consist of one half edge at each of the end points of the edge.

**Definition 3.1 (Transition systems)**

Let $G$ be a graph, and $v \in V(G)$ be a vertex of $G$. A *pairing* at $v$ is a set $\{e, f\}$ where $e$ and $f$ are distinct half edges at $v$. A *transition system*, or *configuration* of $v$, denoted by $T(v)$, is a partitioning of the half edges at $v$ into pairings so that each half edge appears in exactly one pair.

**Definition 3.2 (Turning costs)**

For each possible pairing $\{e, f\}$ at $v$ we assign a non-negative rational number, which we call the *turning cost* of the pairing, denoted by $w_v(e, f)$. We call the set of turning costs at $v$ to be the turning costs at $v$, and for each transition system, we call the sum of the costs of the pairs in the configuration the total cost of the configuration at $v$, and denote this by $w(T(v))$.

If you consider an Eulerian circuit on a graph, starting at an arbitrary point and following the circuit in either direction will generate a transition system at each

vertex, by pairing the half edges that enter and then leave the vertex. Conversely, fixing a transition system at each vertex will induce a set of circuits on the graph, but these may not form an Eulerian circuit, even on an Eulerian graph. As an example, consider the transitions at a cut vertex, which separates the graph into two components if removed. Any transition system in which the half edges from one component are entirely paired among themselves cannot possibly induce an Eulerian circuit, since no trail within this component will ever leave and hence cannot span the edges of the graph.

Given an Eulerian circuit $C$, we denote the transition system determined by this circuit at a vertex $v$ by $T_C(v)$. For such an Eulerian circuit, the *cost* of $C$, denoted by $w(C)$, is the sum of the costs of the pairings that it determines:

$$w(C) = \sum_{v \in V(G)} w(T_C(v)).$$

We can now state the optimisation problem formally.

**Problem 1**

*Given an Eulerian graph $G$ equipped with a set of turning costs at each vertex, find an Eulerian circuit $C$ with the minimum cost $w(C)$.*

Since this problem is clearly not a decision problem (no yes/no answer), we can also state the corresponding decision form of Problem 1;

**Problem 2**

*Given an Eulerian graph $G$ equipped with a set of turning costs at each vertex, and a non-negative constant $c$, determine if there is an Eulerian circuit $C$ with a cost $w(C) \leq c$.*

We call both forms of this problem the turn-costed Eulerian circuit problem or ETCP. Verifying a possible solution (a given Eulerian circuit on $G$) to Problem 2 is clearly possible in polynomial time, since this requires only verifying the induced turns at each vertex, which can be carried out by following the path of the circuit and recording the edge sequence, and then summing the total costs of these turns and hence Problem 2 lies in NP. If a polynomial time algorithm for finding a solution

to Problem 1 were known, then it could be used to solve Problem 2 immediately, by finding the minimum and comparing it to the value of $c$ specified, and hence it would lie in P. We, however, will demonstrate that both problems are, in general, intractable (assuming P$\neq$NP).

It has been pointed out to us by Graham Brightwell, that this result is implied by the result of Bent and Manber on A-trails in [8]. In this paper they demonstrate that verifying the problem "Given a graph drawn in the plane, is there an Eulerian circuit in which successive edges always belong to a common face?" is NP-complete. This implies the NP-completeness of Problem 2 by taking the plane graph to be considered and setting turn costs that lie along a common face to 0 and those that do not to 1. A 0 cost turn costed Eulerian circuit on this plane graph implies the existence of such an Eulerian circuit as required. However in the following sections we also prove that several restrictions of the problem, such as bounding the maximum degree of the graph, do not make the problem tractable and for this we require our original result as the methods of Bent and Manber are not applicable here, and so include it. We also demonstrate a number of restrictions for which we do however produce a polynomial time algorithm.

## 3.2 Motivation

Both forms of ETCP we discuss, and the related problems we discuss later, arise from biomolecular computing and a design strategy problem in DNA self-assembly via origami folding, which involves finding an optimal route for a scaffolding strand of DNA through a targeted structure.

Self-assembly is the physical process by which structures form from disordered components without outside direction, based on the local chemical and physical properties of the materials used. It is an important property in nanoscale constructions, where the creation of tiny structures can be extremely difficult to direct and control.

DNA naturally possesses properties that lends itself to self-assembly, since bond-

ing of DNA strands is dependent on the base pairs within the strands, following rules that are well understood. This naturally leads to the double-helix structure that would be familiar to many people today. DNA self-assembly design problems are those in which synthetic DNA strands are designed such that the chosen base pairs will result in the formation of a particular desired construct. Initial uses of DNA self-assembly were little more than proofs of concept, producing 'artistic' 2D structures, such as a smiley face, but a range of possible applications have been proposed such as self-destructing drug delivery systems.

In 'DNA origami', a single scaffolding strand of DNA traces a construct exactly once, and then short helper strands, called staples, bond to this strand to fold and lock it into the desired configuration (see [35], [51], and [56] for a recent survey).

DNA origami has initially been applied to constructing a range of 2-complexes (solid 2D) structures, and later 3-complexes (solid 3D structures). A logical next step is adapting this technique to 1-complexes, or graph-theoretical structures, such as the skeletons of polyhedra. Such graph-theoretical structures (such as cubes [11]; truncated octahedra [67]; rigid octahedra [60]; tetrahedra, dodecahedra, and buckyballs [33]; and a 3D crystalline lattice [68]) have already been assembled via a different method known as branched junction molecules.

DNA origami is known to have some practical advantages over branched junction methods (the scaffolding strands and staples are easier to produce than branched junction molecules, for example), so it is of interest to try to assemble these and similar structures from DNA origami. However, the design strategies for 'filled' constructions, such as the stars and smiley faces of [56] or the 3D solid bricks and honeycombs and modularly assembled icosahedron of [13], are different from those needed for graph-theoretical structures such as 1-complexes.

The design process for using DNA origami folding to produce self-assembling molecular structures involves finding a route for a scaffolding strand to take through the desired structure. When aiming to produce a construct with the structure of a 1-complex or graph embedded in 3-space, for example a polyhedral skeleton, then

the route of the scaffolding strand must correspond to an Eulerian circuit through the graph, or through some augmentation of the graph (if it is not Eulerian, for example).

In general since DNA bonding of matched pairs is energetically favourable, a system will likely maximise the number of matches naturally, according to the laws of thermodynamics. However, other physical properties and behaviour of DNA strands may interfere and resultantly, at each vertex there may be a preferred route for the scaffolding strand, for example, following a face of the structure rather than weaving through the vertex. This gives us the graph theoretic associated problem of finding an Eulerian circuit with minimum turning cost, outlined above as Problems 1 and 2.

Our result, in proving that in general finding the minimum weight Eulerian circuit is NP-Hard and remains so for bounded degree structures, produces an immediate implication to using DNA origami in biomolecular computing. Graph invariants are properties of graphs which remain unchanged under isomorphism (such as the number of edges or vertices of the graph, or the existence of a Hamilton cycle). Verifying these properties is known to be NP-Hard in many cases (Hamilton cycles and graph colourability being two well known examples). Being NP-Hard does not mean that advances in tackling the problems cannot be made and biomolecular strategies have been developed for solving a number of such problems.

Biomolecular computing is the process of encoding a problem directly into structures, such as DNA, and taking advantage of the massive amount of data intrinsically encoded into its structure, and the ability of chemical processes to effectively run massively efficient and parallel computations at speeds that modern day silicon based computing cannot hope to match. The idea of nano-computation is widely credited as originating with Richard Feynman and his 1959 talk "There's plenty of room at the bottom" [22], where he proposed structural arrangement on a molecular scale to tackle problems that were outside the scope of the scientific community of the day, such as the production of nanoscale machines and computers. The first proof

of concept of such computation came in 1994 when Adleman [2] demonstrated that a biological system could be used to tackle the (NP-Hard) directed Hamilton cycle problem, actually managing to encode a 7 node example into a DNA structure. This approach has continued to attract interest and further results have been developed, for example, see [38] for strategies for tackling 3-SAT and vertex 3-colourability.

It is hoped that DNA origami might well be a sensible approach for encoding graph problems into biological processes, but the NP-Hardness results shown here demonstrate that caution is required, since the initial self-assembly design process of assembling a graph through DNA origami to tackle difficult problems may itself already require solving an NP-Hard problem.

Despite this gloomy outlook, a positive result comes from work with our co-authors; Joanna A. Ellis-Monaghan, Iain Moffatt and Greta Pangborn in [19], where we showed that ETCP can be transformed into a Travelling Salesman Problem (TSP). While the TSP is also in general NP-Hard, it is a problem that has attracted a huge range of research, for its wide ranging applications and uses (see [61] for a detailed survey). These results and work on TSP, while obviously not producing polynomial time algorithms, still provide computational approaches and software that are far beyond the machinery available for most NP-Hard problems and this progress, as a consequence of this result, can be brought to bear on finding optimal routes for a scaffolding strand for DNA origami assembly of graph-theoretical structures.

It is unclear at this point what these results say for the relative feasibility of using DNA origami for biomolecular computing. The NP-Hard result suggests that the complexity of the initial input makes the approach suboptimal and that single strand DNA origami methods may not be suitable as a generic starting point for efficient biomolecular computing of graph invariants. We previously mentioned branched junction molecules as an alternative method of producing self-assembling structures, and for this method, there are several provably optimal design strategies for a few common classes of graphs (see [18, 1]), but at this point no-one has looked at the

general computational complexity of the design problem for these methods, and it may be that they too are intractable in the general case.

### 3.2.1 Related problems

The ETCP problem at first glance may appear very similar to the 'Mill Routing Problem', in which a (possibly non-Eulerian) graph is assigned turning costs and the minimum weight tour that covers all the edges is sought.

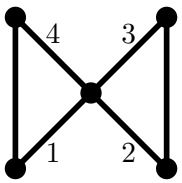
The Mill Routing Problem derives from discrete thin mill routing, based on the physical scenario of a router cutting out patterns from a solid surface of material. For mass-production, it is ideal for the router to go as quickly as possible, and as such turning at a grid point will be slower than going straight, and reversing may be extremely costly or impossible depending on the physical design of the router and as such this can be modelled as a turning cost problem.

This is a very similar set-up to Problem 1, with the only difference being that we insist that the covering tour is Eulerian. This problem of finding a minimum tour, rather a minimum Eulerian circuit was tackled in [7], where they show that the problem is in general NP-Complete.

At first glance, the change of insisting on an Eulerian cover may not seem significant and a naive view might suggest that they are equivalent on Eulerian graphs, but in practice, this changes the problem considerably, requiring a different approach to analyse the complexity of the problem. In general, subtle changes to similar problems can dramatically alter the complexity of the problem, as we already mentioned with subtle changes to the Chinese Postman problem moving it from polynomial solvable to NP-Completeness (see [16] and [23, 24] for overviews).

Relaxing the restriction on an Eulerian tour, even on an Eulerian graph, allows for doubling back or repeating edges, which should be avoided in DNA origami. The methods of [7] are not applicable to our restriction, as they in fact prove that in finding a minimum weight covering tour, even the number of turns is NP-Hard to decide, while this is fixed in our restriction of the problem.

An extremely simple turn costed Eulerian graph, for which the optimal covering tour is not Eulerian can be seen by taking the graph consisting of two triangles, intersecting at a single shared vertex.




We assign a cost of 0 to all pairs at the degree 2 vertices and those at the degree 4 vertex which stay within a single triangle (i.e. $\{1,4\}$ and $\{2,3\}$), and to one of the four transitions that pair edges from different triangles (say $\{1,2\}$), with a weight of 1 on the remaining 3 pairs ($\{1,3\}$ and $\{2,4\}$ and $\{3,4\}$) at the centre.

An optimal Eulerian circuit must use two distinct pairings at the centre vertex that cross from one triangle to another (either $\{1,2\}$ and $\{3,4\}$ or $\{1,3\}$ and $\{2,4\}$), and so will have total cost at least 1 even if it uses the 0 cost transition as one of its two choices, while in the mill routing case, the tour can back track on itself, using the same 0 cost crossing pair at the centre vertex twice to cross back and forth between the triangles and cover all the edges with a total cost of 0.

In light of this, the results of [7], which show that the mill routing problem is NP-Hard, cannot be applied here. Our results, however, show that the discrete thin mill routing problem remains NP-Hard in the restricted case that the desired tour must be an Eulerian circuit. Therefore, even if a set of repeated edges is specified ahead of time, which can be modelled by doubling edges, or if, in general, augmenting edges are added to make the graph Eulerian, then the general discrete thin mill routing problem remains intractable if the desired covering tour must also be an Eulerian circuit.

## 3.3 Finding an Eulerian circuit with minimum turning cost is NP-Hard

We now move on to the main sections of this chapter, the proofs that Problem 1 is NP-Hard and Problem 2 is NP-Complete and the consideration of the tractability of various restrictions of these problems.

We tackle this proof by demonstrating that Problem 2, namely that checking the existence of an Eulerian circuit with total turning cost less than or equal to some value, can be used to solve an arbitrary 3-SAT problem. Formally we prove that an arbitrary 3-SAT instance is polynomial time reducible to an ETCP problem. This means that a polynomial time solution to a general ETCP problem could be used to produce a polynomial time solution to 3-SAT, which is known to be NP-Complete [12], and hence ETCP is also NP-Complete, since it clearly lies in NP. As mentioned earlier, a polynomial time solution to Problem 1 implies one for Problem 2, since comparing the weight of the minimum circuit to the chosen value yields an immediate answer, and hence Problem 1 is therefore also NP-Hard.

A 3-SAT problem is a logical evaluation problem, involving a Boolean logic expression involving Boolean variables $x_1, \ldots, x_n$ in $m$ logical clauses, and the output is whether or not there is a configuration of the $x_i$ (to either true or false) such that the complete expression evaluates to true. As an example, a 3-SAT problem looks something like

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_4 \vee x_5) \wedge \cdots \wedge (x_1 \vee \neg x_3 \vee x_4).$$

Formally, each parenthetical clause in the Boolean expression is a collection of 3 distinct boolean variables, $x_i \in \{x_1, \ldots x_n\}$ in either positive ($x_1$) or negative form ($\neg x_1$, where $\neg$ is the logical 'not' operator), connected by logical 'or' ($\vee$) operators. The instance is formed of these clauses connected by logical 'and' operators ($\wedge$). A 3-SAT instance containing $m$ clauses in $n$ variables, therefore has input parameters $n$ and $m$. We refer to $x_i$ and $\neg x_i$ collectively as the *literal* corresponding to $x_i$, with $x_i$ being referred to as a positive literal and $\neg x_i$ as a negative literal.

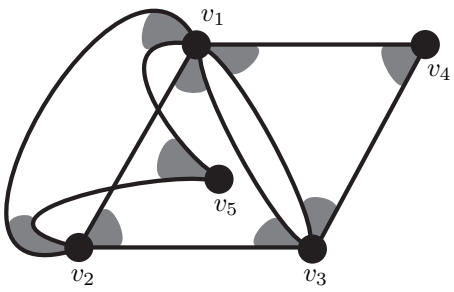



Figure 3.1: Triangles for 3-SAT formula: $(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (x_1 \vee x_2 \vee \neg x_5)$.

### 3.3.1 Constructing an Eulerian graph with turning costs from a given 3-SAT instance

We begin with an arbitrary 3-SAT instance, $I$ and for each variable $x_i$ we construct a vertex $v_i$ to model this variable. To help visualise this process, we embed each $v_i$ in a plane.

Analysing the 3-SAT instance from left to right, we add edges to the graph to represent each clause. Supposing the first clause contains $x_i$, $x_j$ and $x_k$, in either positive or negative forms (for example $(x_i \vee x_j \vee \neg x_k)$), we connect vertices $v_i$, $v_j$ and $v_k$ with a single edge between each, forming a triangle.

We do not require the graph to be planar, so we do not care whether the edges are crossing, but we do require each of the 3 pairs of half-edges to be consecutive in a clockwise cyclic ordering of the half-edges at each vertex, i.e. the edges $(x_i, x_j)$ and $(x_i, x_k)$ must appear consecutively in a clockwise ordering of the edges adjacent to $x_i$.

It does not matter in which order these half edges appear within their pair, nor in what order the pairs themselves appear, as long as the half-edges within each pair are adjacent.

Again, purely for ease of visualisation and description, we shade a small region between the pair of half-edges within each triangle to record this property, as seen in Figure 3.1.

As this process continues, it is likely that multiple edges will be added between

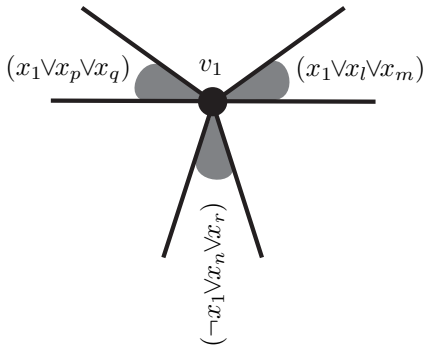



Figure 3.2: Initial neighbourhood of a vertex $v_1$.

various pairs of vertices. If one wishes to restrict this problem to simple graphs, each multiple edge $e, e' = (u, v)$ can be replaced with edges $(u, x)$ and $(x, v)$ with a weight of 0 assigned to the only possible transition at $x$. Clearly this does not affect the existence of an Eulerian circuit of any given weight, and since the number of added vertices and edges is less than the number of edges in the original multigraph, it makes no difference to the complexity of the problem.

Each triangle is labelled with the clause it represents. Observing the structure of the vertex $v_i$, at this point, observe that each vertex is of even degree, with pairs of adjacent edges alternately bounding either unshaded regions or shaded regions, labelled with a clause from the 3-SAT instance containing either a negative or positive occurrence of the variable $x_i$.

We now add an extra vertex $u$, above the plane. We call $u$ the *apex vertex*. We will be adding edges from each $v_i$ to $u$, having them leave $v_i$ within the unshaded regions, and lying in the plane for a small neighbourhood of $v_i$ before rising to $u$ as seen in figure 3.3.

The number of edges we embed in each unshaded region depends on whether the bordering shaded regions are labelled with clauses that contain $x_i$ in positive or negative form. For each unshaded region around $v_i$, if both adjacent labelled triangles contain $x_i$ in positive form, or both contain $x_i$ in negative form, we add two edges from $v_i$ to $u$, emerging from this region. If however the literal is positive

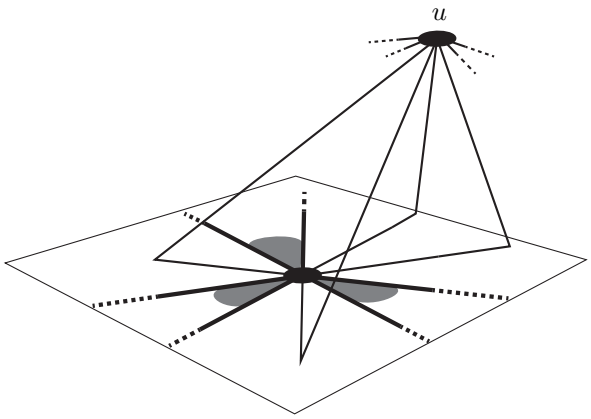



Figure 3.3: Adding the apex vertex $u$.

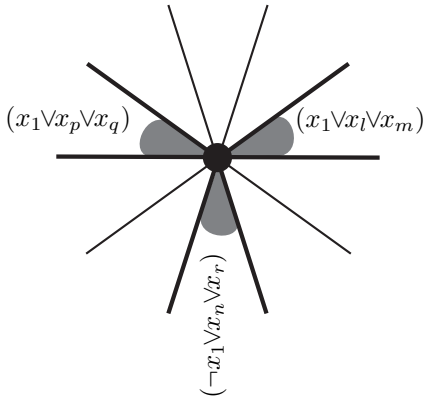



Figure 3.4: Final neighbourhood of a vertex $v_1$.

in one clause and negative in the other, we add a single edge in this region as seen in figure 3.4.

We observe that each vertex is still of even degree. This can be seen by considering the sequence of bounded regions and the form of the literal $x_i$ in each one. Between instances of $x_i$ that are both positive or both negative, we add two edges. We add one edge only when the 'polarity' of the instance switches, and again when it switches back. Since the edges are in a cyclic ordering, we must switch an even number of times, and hence add an even number of single edges. Since the number of edges from $v_i$ to $u$ is even at every $v_i$, the degree of $u$ must clearly also be even.

At this point we have constructed an Eulerian graph, since every vertex is of

even degree and connected to $u$ and hence the entire graph is connected.

All that remains in the construction is to add turning costs at each vertex. This is done according to a simple scheme. All possible turning configurations at the apex vertex, $u$ are assigned a cost of 0. At each $v_i$ we assign a weight of 0 to any edge pairing that appears consecutively in the clockwise orientation and a weight of 1 otherwise.

This is the Eulerian graph with turning costs associated to the given 3-SAT instance.

### 3.3.2 Equivalence of 3-SAT to finding an Eulerian circuit with minimum turning costs

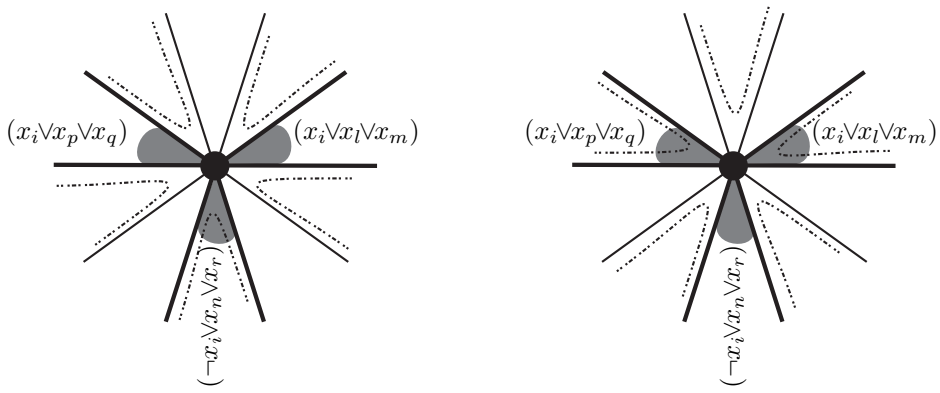
We now demonstrate checking the satisfiability of a given 3-SAT instance is equivalent to finding the minimum weight Eulerian circuit of the associated graph we constructed in subsection 3.3.1. Formally we prove the following;

**Theorem 3.1**

*There is a solution to a given 3-SAT instance $I$ if and only if there is a zero weight solution to the turn costed Eulerian circuit problem (ETCP) on the associated Eulerian graph $G_I$, with turning costs, as constructed in Subsection 3.3.1, i.e. an Eulerian circuit through the associated graph with total weight zero.*

Proof We first suppose that a 0 weight solution to the ETCP exists on the associated graph, $G_I$. Since the total cost is 0, the configuration at each vertex must also have total weight 0, and hence each pairing in the circuit has cost 0. This implies that at each $v_i$, each edge in the circuit is paired with a consecutive edge in the clockwise ordering of the adjacent edges (since these are the only pairs assigned a cost of 0), and hence these edges appear consecutively in the Eulerian circuit itself.

Choosing any edge at $v_i$ and examining which of its neighbours it is paired with in the 0 cost circuit, must then fix the choices for each of the edges at $v_i$, since the consecutive edge not paired with the initial edge we examine must therefore be paired with its only other neighbour, which then fixes the next edges in the cyclic




$x_i$ is true. $x_i$ is false.

Figure 3.5: These are the only two possible zero-cost transitions at a vertex $v_i$. Dotted lines in the left image represent the Eulerian circuit configurations at the vertex $v_i$ corresponding to variable $x_i$ set to true, hence connecting $\neg x_i$ triangles. The image on the right represents $v_i$ when $x_i$ is set to false.

ordering, and so on. This means that there are only two 0 total weight configurations at each vertex.

Due to the way in which we added single or double edges from $v_i$ to the apex vertex $u$, these two configurations behave in a fixed manner with regards to the shaded regions corresponding to the clauses that $x_i$ appear in. If two edges adjacent to a shaded region corresponding to a clause where $x_i$ appears in positive form are paired together, then all such pairs bordering positive instance of $x_i$ will also be paired up. Conversely all edges adjacent to regions corresponding to negative instance of $x_i$ will have the edges paired up with edges going from $x_i$ to the apex vertex. If instead two edges adjacent to a region with a negative instance of the clause are paired, all such negative regions will have their bordering edges at $x_i$ paired and the edges bordering the positive instances will be paired with the edges going to $u$.

In simpler terms, the two 0-cost configurations at $v_i$ will either pair up all edges incident to $v_i$ that are part of a triangle bordering regions corresponding to clauses containing positive instances $x_i$ or all edges incident to $v_i$ that are part of a triangle bordering regions corresponding to instances of $\neg x_i$ but never a mixture of the two.

This dual configuration set-up allows the states of $v_i$ to be linked to the boolean value of the literal $x_i$. Somewhat counter-intuitively, if the configuration at $v_i$ pairs up the edges correspond to $x_i$, we say the vertex is in the 'false' configuration, while if it connects up the edges corresponding to $\neg x_i$, we say $v_i$ is in the 'true' configuration.

We claim that examining the Eulerian circuit and recording the configuration of each vertex and then assigning each variable the value corresponding to its representative vertex's state produces a solution to the original 3-SAT problem.

Suppose that this process does not produce a solution to the 3-SAT problem, then at least one of the clauses in the 3-SAT instance is false. We examine the triangle corresponding to this clause. Since the clause is false, each $x_i$ in the clause must be set to true if $\neg x_i$ appeared in the clause and false if $x_i$ appeared there. If one of the $x_i$ is set to false, when the clause this triangle is labelled with contains $x_i$, then the edge configuration at $v_i$ is set to its false configuration, which means that the edges forming this triangle appear sequentially at $v_i$ in the Eulerian circuit. Equally if $x_i$ is set to true, and the clause contains $\neg x_i$ then the vertex is in its true configuration which will also link up the edges at $x_i$ in this triangle.

The clause evaluating to false therefore implies that all three edges of the triangle are paired up at each vertex, and hence form a closed loop. This cannot occur in an Eulerian circuit and hence we have a contradiction, since we derived the proposed solution to the 3-SAT instance from a solution to the turn costed Eulerian circuit problem.

For the converse, we must show that if a solution exists for the 3-SAT instance, then this graph and associated turning cost problem has a zero weight solution.

We begin by examining the boolean value of each $x_i$ and assigning each vertex the corresponding configuration state. By design, either configuration we can assign at each vertex will have weight 0 and any configuration at the apex vertex will have weight 0 also. All that remains is to check that this 0 weight configuration does form an Eulerian circuit.

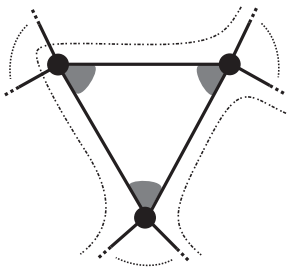


Figure 3.6: Subpaths in an Eulerian circuit around a triangle, recalling that the Euler circuit follows turns with cost zero, i.e. using consecutive edges about a vertex.

As in the previous argument, we can see that we will not have any closed triangles, as this would correspond to an unsatisfied clause which is a contradiction to our assumption of a solution for the 3-SAT instance. The edges around each triangle, therefore form up to 3 paths of length at most 3 which terminate at edges going to the apex vertex.

Every edge from the apex vertex, $u$ is either paired up with another edge to $u$, forming a path of length 2, or to a path bordering a triangle, which also continues to the apex. Since any configuration at $u$ has weight 0, we can simply concatenate these paths in any cyclic order at $u$ which must form an Eulerian circuit with total weight 0 as required. ∎

Before we move on, we make the following observation that will be useful later and is best understood with the construction and equivalence in mind. This observation essentially notes that in the graph constructed above, we could take any vertex representing a literal, and the apex vertex and add two edges between them without impacting Theorem 3.1, namely the equivalence of a solution to the 3-SAT instance and the existence of a 0 cost Eulerian circuit in the graph. As such, since the construction already ensures that the vertices have even degree, we can also assume that every vertex has a degree not divisible by 4.

**Observation 1**

*Given a 3-SAT instance $I$, we may assume without loss of generality that any vertex $v_i$, representing the literal corresponding to $x_i$ in $G_I$, the Eulerian graph with turning*

*costs of $I$ that is used in the proof of Theorem 3.1, has a degree that is not divisible by 4.*

Proof If the degree of $v_i$ is divisible by 4, then modify the graph by adding two additional copies of any edge $(v_i, u)$ from $v_i$ to the apex vertex, placing them immediately after $(v_i, u)$ in the cycling ordering of the edges around $v_i$ and making the region between them in the neighbourhood of $v_i$ unshaded. This increases the degree by two, without changing the parity of the number of edges in any unshaded region. Thus, we are still able to distinguish between the two possible transitions at the vertex $v_i$ (consider Figure 3.5 with the two additional edges added), so the additional edges do not affect the ways in which a zero-cost Eulerian circuit can follow the triangles corresponding to the clauses, and the construction of the graph is still polynomial time. Thus the proof of Theorem 3.1, still holds with the modified graph and does not increase the complexity of constructing the graph. ∎

In light of the above, all that remains is to demonstrate that the associated graph with turning costs can be constructed in polynomial time from the given 3-SAT instance.

**Proposition 3.2**

*For a given 3-SAT instance, the associated graph with turning costs, $G_I$ as described in section 3.3.1 can be constructed in polynomial time.*

Proof To demonstrate that the construction of the graph $G_I$ can be completed in polynomial time, we first recall that our inputs are $n$, the number of literals, and $m$, the number of clauses in our 3-SAT instance. We aim to show that the graph can be constructed in $\mathcal{O}(nm^2)$ time.

Firstly, we note that to generate the vertex set of $G_I$, all we require is to know the value of $n$. This can be found by simply reading each clause in turn and recording the highest index seen through the whole process.

Equally, the pairs of edges we add to form the triangles corresponding to each clause, can be added in any order, as long as they are added in a clockwise ordering

around each vertex, and so can be added in as the clauses are read from left to right. There are 3 literals in each clause, corresponding to $3m$ edges added to the graph and at most $2m$ at a single $v_i$ vertex.

Once the edges between $v_i$ have been added, the only edges remaining are those that are required between the $v_i$ and the apex vertex $u$. This involves going through the cyclic ordering of edges around each $v_i$ and adding 1 or 2 edges for each pair of triangle forming edges, depending only on the labelling we added to the region between the existing edges. As such this process adds at most $2m$ edges to the graph at each vertex, and therefore requires $\mathcal{O}(mn)$ steps to complete.

Assigning the turn cost transitions requires us to list each possible pair of edges at a vertex. Since no vertex has degree greater than $4m$, this is at most $\binom{4m}{2} = \mathcal{O}(m^2)$ pairs at each vertex. We do not need to list the transitions at $u$ since all transitions at $u$ (which has degree at most $3m$) have weight 0 and so do not contribute to the total weight. Therefore, at most we require $\mathcal{O}(nm^2)$ steps to record the possible turning costs.

After these steps, the graph with turning costs are completed, and so the overall time required is $\mathcal{O}(nm^2)$ which is polynomial in $n$ and $m$ as required. ∎

**Corollary 3.3**

*Problem 2 is NP-Complete.*

Proof Problem 2 is clearly in NP, since the cost of any Eulerian circuit can easily be calculated in polynomial time. In light of Theorem 3.1 and Proposition 3.2, we have shown that 3-SAT is reducible to Problem 2. 3-SAT is well known to be NP-Complete, and hence Problem 2 must also be NP-Complete as required. ∎

**Corollary 3.4**

*Problem 1, which is the optimisation form of Problem 2, namely, finding a minimum cost Eulerian circuit is NP-Hard.*

Proof Problem 2 is clearly solvable in polynomial time if Problem 1 can be used as an oracle. ∎


Figure 3.7: The three transition systems of a vertex $v$ in a face two colored 4-regular plane graph.

## 3.4 Tractability of restricted versions of ETCP

As we have mentioned previously, it can often be the case that restrictions of an intractable problem can be solved in polynomial time, and that is indeed the case here. Some obvious restrictions to consider are often found by limiting graphs to planarity, bounded degree or adding additional conditions to the structure to be sought. We consider several variants and combinations of these restrictions. In addition to finding polynomial time variants, we demonstrate that for several restrictions, the problem remains NP-Hard. This is interesting as it provides insight into what properties are important in making the problem difficult to solve and a lot of research is dedicated to finding the thresholds of restriction at which problems switch from NP-complete or NP-Hardness to polynomial time solvability.

As a starting point, in [19] along with our co-authors, we demonstrated that for 4-regular plane graphs, with the added restriction of not allowing 'crossing' transitions at a vertex (for a given plane embedding of the graph), the problem becomes polynomial time solvable and produced an algorithm to solve it. It is fortuitous that this case is tractable, as many likely graph structured targets for DNA origami assembly, for example lattice subsets and cages, are planar, while requiring that a scaffolding strand and staples follow faces without crossing over one another respects the physical constraints of DNA.

We include this proof despite it being superseded by later results because it uses an interesting and different approach that may be of interest for related restrictions.

### 3.4.1 4-regular plane graphs with no crossing transitions

If $v$ is a vertex in a 4-regular plane graph, then it has 3 transition systems, determined by the embedding in the plane. Figure 3.7 illustrates a degree 4 vertex with its 3 configuration states, the last being a 'crossing' transition. If an Eulerian circuit does not use crossing transitions, then all of its edges follow the boundary of one of the two faces that the previous edge in the circuit also bordered. In this section, we will assume the crossing transition systems are prohibited. (This can be done by assigning the pairs that comprise them large turning costs, in particular, larger than the sum of all the turning costs of other non-crossing transition systems.).

More generally, if $G$ is an Eulerian graph embedded in some surface, then an Eulerian circuit in which consecutive edges in the circuit, $(v_{i-1}, v_i)$ and $(v_i, v_{i+1})$ say, are adjacent in the cyclic ordering of the edges incident to $v_i$, then we call this circuit an *A-trail* (or a *non-intersecting Eulerian circuit*) of $G$. Using this terminology, an Eulerian circuit of a 4-regular plane graph that has no crossing transitions is an A-trail.

In [42], Kotzig proved that every 4-regular plane graph contains an A-trail. However, Bent and Manber, in [8], showed that dropping the 4-regularity requirement results in a problem that is NP-complete, i.e., the problem of deciding if an Eulerian plane graph contains an A-trail is NP-complete. This remains the case even when restricted to simple, 3-connected graphs with only 3-cycles and 4-cycles as face boundaries (see [5]), although a polynomial-time algorithm for finding A-trails in simple 2-connected outerplane (a planar graph in which all of the vertices lie on the boundary of a single shared, unbounded face.) Eulerian graph was given in [6]. Andersen, Bouchet and Jackson [4] characterised all 4-regular plane graphs that have two orthogonal A-trails, where two A-trails of $G$ are *orthogonal* if the two trails have different transitions at every vertex of $G$. Furthermore the complexity of the related problem of finding Eulerian Petrie walks on 4-regular plane graphs has been studied by Žitnik in [69]. Eulerian Petrie walks are Eulerian circuits in which the non-crossing transition to either the left or right of the edge on which the walk

arrives at the vertex, is selected alternately at each vertex in the walk.

In light of these results, it is non-trivial to determine the minimal cost A-trail when turning costs are assigned to the graph. We demonstrate below that it can be accomplished in polynomial time for 4-regular plane graphs.

**Theorem 3.5**

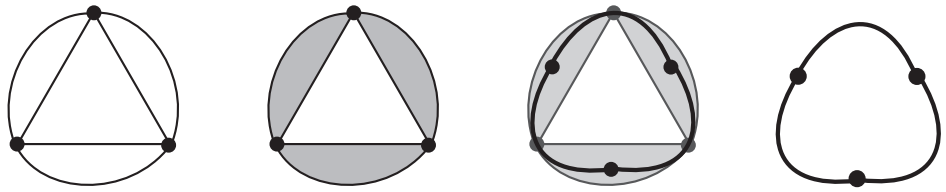
*If $G$ is a 4-regular plane graph with a set of turning costs such that the crossing transitions are prohibited, then an optimal Eulerian circuit may be found in polynomial time.*

Proof For any connected plane graph $G$, one can form its medial graph by taking a vertex for each edge of $G$ and connecting two such vertices in the medial graph with an edge for each face of $G$ in which their original corresponding edges in $G$ appear adjacently (i.e. consecutively in a clockwise cyclic ordering of the edges forming the face).

It should be noted that medial graphs depend on the planar embedding used for $G$ and as such are not unique under isomorphism.

For each edge in $G$, at each of its endpoints, the corresponding vertex in the medial graph will gain two half-edges, either one for each other edge it appears consecutively with (either clockwise or anti-clockwise) in a face, or if the original edge is a loop to itself, it will gain one loop and two half edges to the other adjacent edges. In this way the medial graph generated will also be 4 regular, with a number of vertices equal to the number of edges of $G$, which is clearly of polynomial size in terms of the size of $G$.

It is known that every 4-regular plane graph $G$ is the medial graph of its Tait graph (or blackface graph), as in Figure 3.8 (see, for example, [20] for details). The Tait graph, $F$, is constructed by face 2-colouring $G$ using the colours black and white such that the unbounded region is coloured white, and placing a vertex of $F$ in the interior of each black face. (Note that $G$ is face 2-colourable as it is plane and 4-regular.) There is an edge between two vertices in $F$ whenever the two regions corresponding to the vertices have a shared vertex of $G$ on their boundary. The edge



A 4-regular plane graph $G$. | A face 2-colouring of $G$. | Forming $F$. | The Tait graph $F$.

Figure 3.8: Forming Tait graphs.

is drawn between the two vertices of $F$, passing through this shared vertex of $G$. Thus, there is a one-to-one correspondence between the edges of $F$ and the vertices of $G$, and if $v$ is a vertex of $G$, we label the corresponding edges of $F$ by $e_v$.

The face 2-colouring of $G$ allows us to distinguish the two non-crossing transition systems at each vertex as either a black smoothing or a white smoothing, as in Figure 3.7. (The term smoothing derives from standard terminology in knot theory.) It is well-known that there is a one-to-one correspondence between spanning trees of $F$ and Eulerian circuits of $G$. The correspondence identifies an edge $e_v$ in a spanning tree of $F$ with a white smoothing at $v$ in the Eulerian circuit of $G$, and an edge $e_u$ not in the spanning tree with a black smoothing at $u$ in the Eulerian circuit. Again, see Figures 3.7 and 3.8.

Suppose for each vertex $v$ in $G$, the cost for the white smoothing is $a_v$, while the cost for the black smoothing is $b_v$. Then we assign the value $a_v - b_v$ to the edge $e_v$ in $F$. Now suppose $C$ is an Eulerian circuit without crossing in $G$, and let $I$ be the set of vertices of $G$ which have a white smoothing in $C$. We can see that the total cost of the Eulerian circuit will be,

$$\sum_{v \in I} a_v + \sum_{v \notin I} b_v = \sum_{v \in I} (a_v - b_v) + \sum_{v \in V} b_v.$$

However, because of the correspondence between Eulerian circuits of $G$ and the spanning trees of $F$, the set of edges $\{e_v \mid v \in I\}$ is a spanning tree of $F$. Since we have assigned the value of $a_v - b_v$ to the edge $e_v$ in $F$, the summand $\sum_{v \in I} (a_v - b_v)$ on the righthand side is the weight of this spanning tree.

Thus, a minimum weight spanning tree in $F$ corresponds to a minimum cost Eulerian circuit in $G$. Since it is well known that minimum weight spanning trees may be found in polynomial time (see for example [44] or [54]) it follows that optimal Eulerian circuits without crossings may be found for 4-regular plane graphs in polynomial time. ■

### 3.4.2 Graphs of bounded degree.

In light of Theorem 3.5, a natural approach was to see if it was possible to relax one of the three conditions needed for that result, namely planarity, 4-regularity or the restriction on crossing transitions and still find a polynomial time algorithm. Planarity is a powerful restriction on graphs and problems which are NP-Hard in general may be polynomial time solvable when restricted to planar graphs. As an example, the Max Cut Problem, one of Karp's 21 NP-complete problems [40], has a polynomial time solution on planar graphs [31].

The following observation however shows that the same is not true for ETCP.

**Observation 2**

*If ETCP is solvable in polynomial time on planar graphs of maximum degree $d$, then it is also solvable in polynomial time for general graphs of maximum degree $d$ (for $d > 2$).*

Proof Suppose we have an algorithm that solves ETCP on any planar graph of maximum degree $d$. Consider a non-planar Eulerian graph $G$ of maximum degree $d$, with turn costs assigned to each vertex. We can assume $G$ does not contain multiple copies of a single edge, as otherwise we can replace any such multi-edges with a path of length 2, which clearly has no impact on the complexity of ETCP. Embed the vertices of $G$ arbitrarily on the circumference of a circle in the plane. Embed all edges of $G$ into the plane as straight lines between their endpoints. Supposing that more than two edges intersect at a single point, replace one of the additional intersecting edges with an edge that curves around this intersecting point in a small circumference around the intersection point that does not intersect with any additional edges, but

that is unchanged elsewhere in the plane. Iteratively repeating this process leaves us with a graph that while not planar, does not have more than two edges intersecting at a single point.

Replace this graph $G$ with a new graph $G'$, that has the same vertices and turning costs on the circumference of the circle, but at each intersection point between two edges, add a vertex that bisects both edges, forming two new edges. This adds at most $\binom{|E(G)|}{2}$ vertices and edges to the graph, which is clearly still polynomial in terms of the graph size inputs. This new graph is now planar. At each new vertex we assign turning costs of zero to the pairings that connect up the two edges that previously formed a single edge, and a cost of $T$ to the other two configurations, where $T$ is equal to the total cost of all possible configurations of the vertices of $G$, plus 1.

By our assumption we can now solve ETCP on $G'$, since we have not increased the degree of any vertex and all new vertices are of degree 4, and hence it is of maximum degree $d$. Whether or not we can find it in polynomial time, an Eulerian circuit on $G$ must have total cost less than $T$. Note also that any Eulerian circuit on $G$ implies the existence of one on $G'$ of equal cost, by taking the zero cost transitions at each added vertex, which traces the same path in the plane as the Eulerian circuit in $G$, and therefore covering every edge of $G'$ as required. Therefore any minimal Eulerian circuit in $G'$ must not use of the transitions with cost $T$, which equally implies the existence of an Eulerian circuit of equal cost in $G$. By these facts we can see that there is a one to one correspondence between minimal weight Eulerian circuits in $G$ and $G'$ and as such solving ETCP on $G'$ provides a solution to ETCP on $G$ as required. ∎

This observation in fact applies to a wide range of graph classes, demonstrating that planarity does not reduce the complexity of the problem in general, but note that this argument cannot be applied to the problem of considering graphs with crossing transitions forbidden, as we require crossing transitions in our construction of $G'$ to maintain the equivalence.

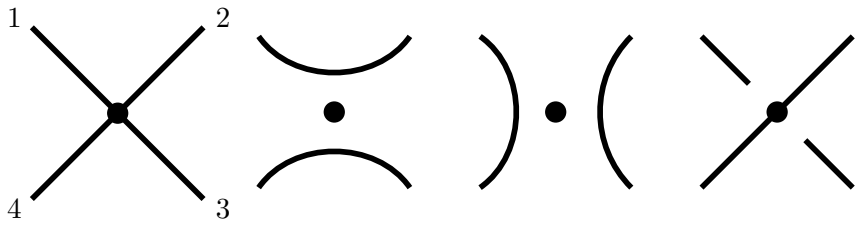



Figure 3.9: A degree 4 vertex and its 3 possible turning configurations.

Having attempted to drop planarity, we attempted to remove the restriction on crossing vertices and consider variants of planarity instead, but in a surprising result we found that the properties we required to solve ETCP in polynomial time on these graphs are actually inherited from all 4-regular graphs. As such we have the following result.

**Theorem 3.6**

*Solving Problem 1 can be completed in polynomial time when restricted to the class of* 4*-regular graphs.*

Proof We begin with several observations. Firstly recall that vertices of degree 4 have only 3 possible configurations, see figure 3.9.

Secondly, we note that given an Eulerian circuit in $G$, if we examine any degree 4 vertex and consider its configuration in this circuit, one of the other two configurations at this vertex would, all else remaining unchanged also form an Eulerian circuit, while the other would separate the circuit into two edge-disjoint circuits.

This can be seen by considering the circuit as it leaves the vertex by one of the edges. We label the edges 1-4 clockwise, with the edge in the top left, labelled 1. Without loss of generality, we can suppose that in our original circuit, the circuit leaves by edge 1, returning to the vertex next using edge 4 (on the lower left), and the vertex has the first of the three configurations. It will then leave by edge 3, before returning to edge 2, which connects back to edge 1, completing the circuit.

We observe that instead adopting the second configuration at this vertex means that the trail from edge 1 to 4 now forms a closed circuit when reaching the ver-

tex. In contrast, configuration 3 creates another Eulerian circuit, where the second half of the circuit is simply traversed in reverse from the direction it would with configuration 1.

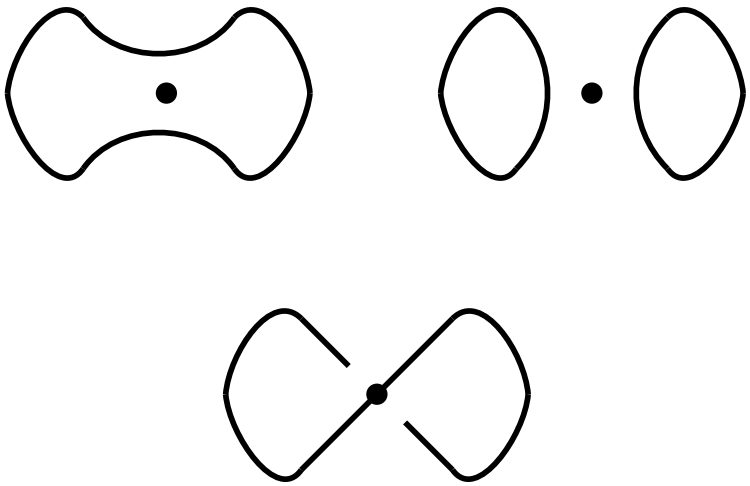


Figure 3.10: Two of the three transitions at a vertex form a single closed circuit.

This observation tells us that no minimum weight Eulerian circuit will need to use the most expensive configurations at any vertex, since supposing such a minimum weight circuit exists is an immediate contradiction, as we could replace the maximum weight configuration at a vertex with one of the other two cheaper configurations and maintain an Eulerian circuit. In the case where two or three configurations are of equal cost, this observation still holds, as one of the equally highest cost configurations can always be discarded.

Using these observations, we propose the following algorithm for finding the minimum weight Eulerian circuit. Firstly set each vertex to its configuration of minimum cost (if more than one are equally minimum, then choose one arbitrarily.). Simply by following edges sequentially through each configuration, we can find the number of disconnected circuits (which we will, in an abuse of notation, call components) this configuration forms, which we denote by $k$. We also record which vertices are in each component, noting that each vertex may lie in one or two components only.

We note that for any vertex which lies in two components, changing its configuration to either of its other configurations will connect these two components into a single circuit, forming a new single component, and reducing the number of components by 1. After $k-1$ repetitions of this process, we will be left with a single component, and hence an Eulerian circuit.

We claim that at each stage, choosing the vertex with least difference between its cheapest and next cheapest configuration, chosen from the set of vertices currently lying in two components, will produce an Eulerian circuit of minimum cost.

Firstly we note that any minimum solution must have exactly $k-1$ of its $n$ vertex configurations set to a cost differing from this initial lowest cost model (except for some number of vertices that may have two minimum cost configurations that do not spilt up the components they are in). Some of these $k-1$ may also be minimum cost configurations, if a vertex has two equal cost states, but the algorithm above would have also chosen these states, as they have a relative cost of 0 to switch to. Any alternative minimum cost configuration at a vertex that the algorithm would not have considered is irrelevant, as if it lies in two components, the switch would connect it, and hence the algorithm would have chosen it unless another 0 cost choice also connects those components. If it lies in one, then the switch either disconnects that component or leaves it connected, leaving the rest of the global structure unchanged, making it at best a neutral choice

Suppose we have a minimum cost Eulerian circuit with more than $k-1$ of its choices differing from a lowest cost configuration. For each vertex not set to a minimum cost configuration, we can observe the impact of replacing it with the configuration of minimum cost. We know that not all of these choices can create a new component, as there must be $k$ at the end, were we to change all of them. Supposing, however that such a change at a vertex did not increase the number of components, we can see that switching only that vertex to a lower cost would keep the minimum cost Eulerian circuit in a single component while reducing its cost, clearly a contradiction.

Equally, any minimum cost circuit cannot have less than $k-1$ differing configurations, since that would imply more than one component in the circuit.

Given these observations, we can see that a minimum cost circuit, consists of $n-k+1$ configurations set the cheapest value, and the remaining $k-1$ chosen to connect up the $k$ components from this cheapest base state.

Analogously to Kruskal's algorithm for minimum spanning trees, we can see that this circuit is minimum by induction. We suppose that at stage $t$ we have made $t$ changes from the cheapest base state and our inductive assumption is that there exist a set of $k-1$ changes including these $t$ that form a minimum Eulerian circuit.

This is clearly true at $t=0$, since we have seen that a minimum circuit that consists of $k-1$ changes must exist. We now suppose true for all $t < t_0 < k-1$. By our inductive hypothesis, we have a set of $t_0$ vertices, whose configurations have been switched from minimum, each one reducing the number of components by one, and some further $k-1-t_0$ changes will create a minimum weight Eulerian circuit. If the next choice of vertex to switch is contained in this minimum circuit, then by induction we are done, otherwise we suppose that the vertex $v_1$ we change next was not changed in this minimum circuit.

Taking this minimum circuit, and then additionally changing $v_1$ can either keep the circuit connected, or disconnect it into two components. Given the first scenario, there must exist some vertex $v_2$ that the algorithm would not have switched, but is switched within the minimum circuit, such that switching $v_1$ instead of $v_2$ keeps the circuit in a single component. We know that this new circuit must have the same weight as the minimum, otherwise the algorithm would have chosen $v_2$ instead of $v_1$. Therefore by induction our changes up to $t_0+1$ also lie within a minimum circuit.

If we are in the second scenario, then there exist two components that switching $v_1$ separates the minimum circuit into. In previous settings, where each change has taken place only in two disjoint circuits, each step can only have created a new component by merging two existing ones, and hence each component consists of a union of the components that form our base state. Here however, switching $v_1$,

which lies entirely within a component, could create two new components, where both new components may have edges from a single original component.

Since $v_1$ was chosen by the algorithm, we know its base state originally lay in two separate components. If we consider the transitions induced by changing all the vertices chosen by the algorithm, except for $v_1$, we are left with a graph with exactly two components, with $v_1$ lying in both. The minimum weight circuit must contain some vertex $v_3$ that lies in both these components, since it must connect the constituent components whose union forms these two. It must also not have been chosen by the algorithm, since $v_1$ would not have been selected as well as $v_3$. Even if the new components formed by switching $v_1$ are different from the original two, this vertex $v_3$ must still lie in both, since by splitting the graph into two new components, the switch at $v_1$ must be connecting two trails that each lay in different components originally, therefore some other vertex must be allowing these trails to connect across the two components, to become closed.

Switching $v_3$ therefore back to its original state, must reconnect the circuit, and hence taking the minimal circuit, except for switching $v_1$ instead of $v_3$ is also an Eulerian circuit. If this new circuit is not also minimal, then $v_3$ must have a lower cost than $v_1$, but if this is the case, then the algorithm would have selected $v_3$ over $v_1$, which is a contradiction, and hence this circuit is also of minimal cost. Since it contains all the choices of the algorithm, up to $v_1$, our inductive step is complete, and therefore by induction, our algorithm produces a minimal circuit.

This algorithm first requires sorting the weights of the configurations at each vertex. Since each vertex only has 3 configurations this can clearly be done in polynomial time. Setting the graph to a configuration where each vertex is at it's lowest configuration takes $n = |V(G)|$ steps and then determining the 'components' induced by these configurations requires simply following the edges until a closed walk is formed. This takes $|E(G)|$ steps, recording the component(s) each vertex lies in as the walks are traced.

At each step of the algorithm, for each vertex that lies in two components, the

differences between its two lowest costed configurations need to be recorded and then, this list sorted to find the vertex with the smallest difference. Generating this list takes at most $3n$ steps and merge sort (for example) can sort this list in $\mathcal{O}(n\log(n))$ time. Once the vertex whose configuration will be switched has been chosen, the components that vertex lies in will be merged and so updating the lists requires only checking each element of the list and updating the component for those affected, or removing those vertices who now lie in a single component. This again takes $n$ steps.

In light of this, each step of the algorithm takes at most $\mathcal{O}(n\log(n))$ time to complete, and there are $k \leq n$ steps before the algorithm completes. As such the algorithm clearly runs in polynomial time as required. ∎

The graphs that we construct in section 3.3.1 in order to prove Theorem 3.1 may have vertices of very high degree, up to linear in the number of vertices in the graph and for these graphs we can demonstrate NP-Completeness. In contrast, 4-regular graphs clearly have bounded degree, which leads to the obvious question of whether Problem 1 can be solved in polynomial time on all graphs of bounded degree. In the following theorem we show that this is not the case as we show that the restriction of only considering graphs with bounded degree vertices, does not, in general, change the complexity of Problem 1. In particular we can show that the problem remains NP complete for graphs of maximum degree 8. With very little modification to the below argument, it can also be shown for 8-regular graphs.

**Theorem 3.7**

*Solving Problem 1 remains NP-hard even restricted to the class of graphs of maximum degree 8.*

Proof Let $G = G_I$ be the Eulerian graph with turning costs associated with a 3-SAT instance $I$ as constructed in Section 3.3.1. By Observation 1, we may assume, without loss of generality, that if a non-apex vertex of $G$ has degree greater than 8, then its degree is not divisible by 4. To prove the theorem, we will construct, in polynomial time, an Eulerian graph with turning costs, $G'$, that has maximum

degree $\Delta(G') \leq 8$. Furthermore, $G'$ will have a zero-cost Eulerian circuit if and only if $G$ does. To construct $G'$, we 'blow-up' each high degree vertex of $G$, replacing it with a special graph that has maximum degree 8. We will need two types of blow-ups: one for the vertices $v_i$ arising from the variables of $I$, and one for the apex vertex $u$.

We denote the edges incident with a non-apex vertex $v_i$ of $G$ by $e^i_1, e^i_2, \ldots, e^i_{d(v_i)}$ and we assume that they appear in that cyclic order (with respect the orientation of the plane, as in the construction of $G$). Furthermore, if $d(u) = 2d$, we let $f_1, \ldots, f_{2d}$ denote the edges of $G$ that are incident with $u$. See Figures 3.11 and 3.12.

For the first type of blow-up, for a vertex $v_i$, we note that since $d(v_i)$ is even and not divisible by 4, then $d(x_i)/2 = 2k + 1$ for some $k$. We then form the structure $B_i$ which will act as the blow-up of $v_i$ as follows. Start with a plane $(2k + 1)$-cycle. Add two parallel copies of each edge in the cycle to the $(2k + 1)$-cycle. At each vertex, place two half-edges such that one lies in the bounded region, and one in the unbounded region formed by the cycle (these will correspond to, and represent the edges $e^i_j$ leaving vertex $v_i$ in the original graph). Label the half-edges by labelling an arbitrary edge $e^i_1$. Then, if $e^i_j$ has been assigned to some half-edge in the unbounded (respectively, bounded) face, travel round the outer face following the orientation of the plane and at the next vertex label the half-edge in the bounded (respectively, unbounded) face $e^i_{j+1}$, alternating between the half edges in the bounded and unbounded region as you follow the cycle around it's clockwise orientation. Continue until all half-edges have been labelled, which happens since $2k + 1$ is odd. See Figure 3.11. Assign pair costs to this graph as follows. Give a cost of zero to any consecutive pairing of half-edges with respect to the (plane) orientation about each vertex in $B_i$. All other pairings of half-edges have cost one. The resulting graph-like structure ($B_i$ would be a graph if degree one vertices were added to the end of each of the half edges.) with turning costs is $B_i$.

To *blow-up* a vertex $v_i$ in $G$, we replace it with $B_i$ as follows. Suppose an edge $e^i_j$ of $G$ has endpoints $v_i$ and $w$. Then we identify the unassigned endpoint of $e^i_j$ in

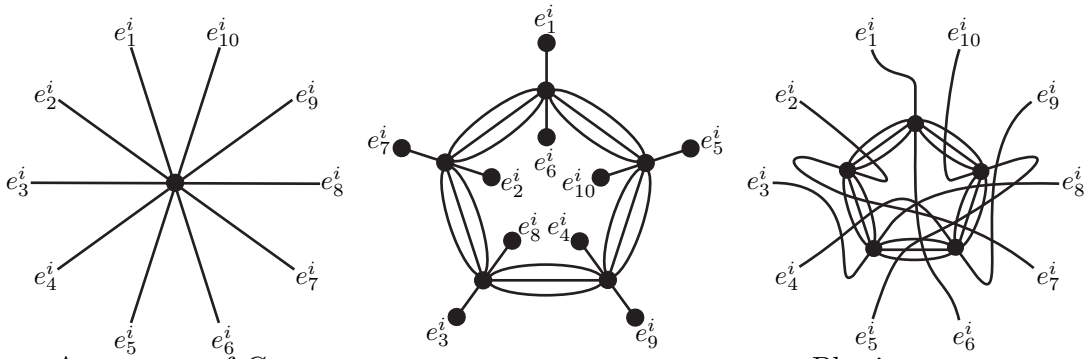
$B_i$ with the vertex $w$. We do this for each $e^i_j$, and then delete the vertex $v_i$ and its incident edges.

Examining a vertex in $B_i$, note that since only adjacent edges in the cyclic ordering have a zero cost pairing, the vertex only has two configurations that can have a total zero cost. Supposing the configuration is fixed for a single vertex, then consider either of the two vertices adjacent to it in the original $(2k+1)$-cycle used to form $B_i$. This adjacent vertex also only has two zero cost configurations but considering the three edges that connect these two vertices, we can see that two of these edges will have been paired up at the first vertex, and as such pairing them again at the second vertex would create a closed loop. This observation means that fixing the configuration of one vertex in $B_i$ will automatically fix the configuration of it's neighbours (assuming a zero cost configuration is desired).

For the blow-up of the apex vertex $u$, which has degree $2d$, we construct the structure $B_u$ by first creating a $d$-cycle on $d$ vertices. For each vertex in the cycle, we attach two additional half edges that will correspond to and represent the $2d$ edges incident to $u$. We assign pair costs of 0 to all configurations at each of these vertices.

The vertex $u$ is blown-up similarly to the $v_i$'s by identifying each half-edge leaving $B_u$ with an edge incident to $u$ in $G$, and then deleting $u$ and its incident edges. The choice of edges to assign to each half edge is irrelevant.

Observe that if a zero cost configuration is fixed at every vertex other than those within $B_u$, as long as a closed cycle is not formed elsewhere in the graph, the configuration at $B_u$ can always be set to form a zero cost Eulerian circuit. This follows from observing that fixing the configurations elsewhere, assuming a closed cycle is not formed, is equivalent to pairing up the half-edges leaving $B_u$ into full edges. Regardless of how this pairing is done, the resulting graph formed from $B_u$ and these pairings will be 4-regular and connected, and hence an Eulerian circuit exists (and will have 0 cost since all configurations at $u$ are zero costed) and can be found in polynomial time.




A vertex $x_i$ of $G$. The graph $B_i$. Blowing-up $x_i$.

Figure 3.11: The blow-up of $x_i$ into $B_i$.

Now, let $G'$ be the graph obtained from $G$ by blowing-up each vertex $v_i$ that has $d(v_i) > 8$ using $B_i$, and, if $d(u) > 8$, blowing-up $u$ using $B_u$. The turning costs of $G'$ are inherited from those of the $B_i$, $B_u$, and those of the vertices of $G$ with degree at most 8. Observe that $G'$ is constructed from $G$ in polynomial time in the number of edges and vertices. It remains to show that $G$ has a zero-cost Eulerian circuit if and only if $G'$ does.

First we suppose that a zero-cost Eulerian circuit exists in $G$. For each $v_i$, we examine the edge $e_2^i$ and record whether it is paired with $e_1^i$ or $e_3^i$ in the zero-cost configuration at $v_i$. Examining $B_i$, we observe the two configurations possible (as we observed earlier, each vertex in $B_i$ has two configurations, but fixing one of them, fixes it for all vertices in $B_i$, and hence $B_i$ as a whole has two configurations that provide a total zero-cost) and we see that this construction will mean that $e_2^i$ will be paired by a trail to one of the edges leaving $B_i$ from an adjacent vertex in the cycle forming $B_i$, but on the opposite region (bounded or unbounded by the cycle) to $e_2^i$ (see figure). By construction, these two possible edges are identified with the edges $e_1^i$ and $e_3^i$ in $G$, and as such, each $B_i$ allows the edge $e_2^i$ to be paired up with the same edge in $G'$ as it was in $G$. Equally the labelling of the edges leaving $B_i$ has been chosen such that, if examined purely in terms of outgoing edges, the two zero-cost configurations in $B_i$ are identical to those of $v_i$, i.e. the two configurations will pair up the outgoing edges in $B_i$ in the cyclic order they are labelled, which corresponds to the cyclic order they were embedded in $G$. In this way, since each $B_i$

pairs up the incoming and outgoing edges in the same ways that $v_i$ does, and uses every internal edge to do so, it is clear that setting each $B_i$ to the corresponding configuration of $v_i$ preserves the Eulerian circuit, as $v_i$ is blown up to $B_i$.

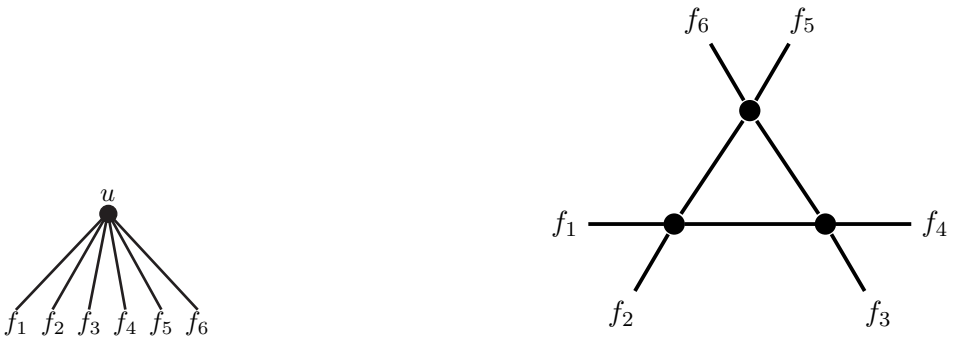
Given this, all that remains is to check that we can still find an Eulerian circuit at $u$, the apex vertex, once it has been blown up to $B_u$. Given that a zero-cost Eulerian circuit exists with the vertex $u$ rather than $B_u$, we can assume that the configurations elsewhere in the graph are fixed and just consider $B_u$. By our earlier observation, we note that this corresponds to pairing up the edges leaving $B_i$, replacing the trails between them with a single edge. This at most takes a number of steps equal to the number of edges in the graph, since we can simply follow the configurations around the graph, and as such can be done in polynomial time. Once this has been done, it is a simple step to find an Eulerian circuit in $B_u$ that uses all of the edges in $B_u$ and since all configurations here have a cost of 0, and all configurations external to $B_u$ have zero-cost configurations, this means that a zero-cost Eulerian circuit must exist for $G'$ as required.

Equally suppose a zero-cost Eulerian circuit exists in $G'$. As before, each $B_i$ has only two zero-cost configurations, which correspond to a pairing of the outgoing edges to the adjacent edges in a cyclic ordering relative to their labelling. Setting the configuration of $v_i$ in $G$ to the same as that of $B_i$ in $G'$ with regards to the pairing of the outgoing edges preserves the structure of $G'$ except for the collapse internally of $B_i$ to $v_i$. This clearly cannot introduce a closed loop.

At $u$, the task is even easier. Since any pairing of the outgoing edges is of zero-cost at $u$, one can pair them up in exactly the same way as they were paired at $B_u$. Again since we are collapsing trails to single edge transitions, which all have zero-cost at $u$, we cannot be eliminating the Eulerian circuit here.

Thus if there is a zero-cost Eulerian circuit in $G$ there is a to a zero-cost Eulerian circuit in $G$, completing the proof of the theorem ∎

With differing results for 4 and 8 regular graphs, there remains a clear area of interest in whether or not Problem 1 remains NP-Complete or becomes polynomial




The apex vertex $u$. The graph $B_u$ for $d = 3$.

Figure 3.12: The blow-up of $u$ into $B_u$.

time solvable (or neither of these two) for 6-regular graphs but as of yet, this problem has defied analysis using the techniques employed for its neighbouring cases but we would conjecture that it would be NP-complete.

### 3.4.3 Related problems

A problem that appears similar, and we thought may be related to the ETCP, is that of the Eulerian superpath problem.

This arises from another very different physical problem that also involves DNA which is that of DNA sequencing. This involves determining the precise ordering of the nucleotides, usually denoted by A, C, G, T and U, in a given strand of DNA or RNA. In practice, this usually involves cutting strands into smaller segments and producing large numbers of copies, or clones of each segment, and performing reads on these copies. By computationally identifying overlapping sequences, the full sequence is derived.

Traditional methods of solving this computational problem are relatively fast initially, but are known to introduce small errors, which take computationally large amounts of time to identify and correct.

In [52], [53], the authors proposed the Eulerian Superpath Problem (ESP) as a possible approach to solving the problem of DNA sequencing by hybridization. Before we state ESP, we first define an Eulerian superpath;

**Definition 3.3 (An Eulerian superpath)**
Given an Eulerian (multi)graph $G$, and a collection $\mathcal{P}$, of trails in $G$, we define an Eulerian superpath of $G$ with respect to $\mathcal{P}$, as an Eulerian circuit on $G$, such that the circuit contains all of the trails in $\mathcal{P}$.

We define the Eulerian superpath problem, for a given $G$ and $\mathcal{P}$, as the problem of determining whether such an Eulerian superpath exists, formally;

**Problem 3 (Eulerian Superpath Problem)**
*Given an Eulerian graph and a collection of trails $\mathcal{P}$ in this graph, find an Eulerian circuit that contains all the trails of $\mathcal{P}$ as subtrails, or determine that it is not possible.*

As the problem was originally formulated, the trails were specified by a sequence of vertices, with multiple edges not distinguished. This problem has since been shown to be NP-Hard in general, see [39] where the problem known as the De Bruijn Superwalk problem is shown to be NP-complete. This problem is ESP formulated on a specific class of graphs called De Bruijn graphs. Since it is NP-complete for these graphs, it must also be for the class of general graphs.

In [52] and [53], an algorithmic approach is provided that will return the solution to Problem 3 in polynomial time for a range of graphs, including those with no multiple edges and in cases where the prescribed trails do not overlap. In general the approach of Pevzner, Tang and Waterman in these papers appears to provide a quick solution to the types of graphs that are generated from the DNA sequencing problem it inspired, but clearly since the problem is NP-complete, they cannot be applied to all graphs or trail sizes.

Since a trail is prescribed to be followed in the Eulerian circuit, it can be seen that as soon as an Eulerian circuit begins following the first edge of the prescribed trail, it must continue on to the end, and hence this would be equivalent to removing the edges of the trail and following a single edge between the start and endpoints of the trail. The only issue is that the edges removed may have been present in other prescribed trails.

The methods employed in the approach of [52] and [53] essentially revolve around iteratively replacing one of the prescribed trails by a single edge in the base graph, and updating any other trails that overlap with the replaced trail by contracting this overlap, where possible, down into the new single edge. If this is definitely not possible, then no solution to the problem exists. If however the trails can be replaced consistently this process replaces the instance of the ESP problem with another ESP instance with one less prescribed trail such that a solution to one exists if and only if a solution exists for the other. If this iteration can be continued, one is eventually left with an Eulerian graph with no prescribed trails, for which ESP is equivalent to just finding an Eulerian circuit.

In general the difficulty arises when there are repeated edges, as it is not clear which of the multiple edges are used in a given trail covering them and hence updating these overlapping trails can have multiple options that may at each stage all appear valid choices, resulting potentially in exponential possibilities to check.

The case where each trail is a single edge is clearly just analogous to finding an Eulerian circuit and hence solvable in polynomial time. The case, however, where each trail is a path of length 2 appears more complex.

At first glance this problem appears to be quite similar to the turn costed Eulerian circuit problem. Each prescribed trail is essentially (in a graph with no multiple edges) a pairing at a vertex which is now fixed. If there are repeated edges, then a number of possible pairings are allowed at the vertex at the centre of the trail, but this still seems closely linked to Problem 1.

In light of this similarity, it seemed likely that ESP would remain NP-complete in the case where each prescribed trail is of length 2, and we attempted to use our results in Corollary 3.3 to prove this, but to our surprise, we actually found the opposite, and as such arrived at the following theorem.

**Theorem 3.8**

*If every element of $\mathcal{P}$ is of length 2, then the Eulerian superpath problem (ESP) can be solved in polynomial time.*

Proof We note that the Eulerian superpath problem can always be solved in polynomial time if $G$ is a simple graph (i.e. no multiplicity of edges). This follows from following a process of replacing each trail in $\mathcal{P}$ with a single edge and then updating each other trail which contained any of the edges replaced, with the new edge, wherever the two trails are "consistent". i.e. they overlap only at the beginning or end of each of the trails (or if one trail is contained entirely within another). If they are not consistent, then the superpath cannot exist.

Formally, if the trail $T = xyz$ exists in $G$ and $\mathcal{P}$ (we restrict ourselves here to paths of length 2 in explanation, but these arguments for simple graphs with no multiple edges are valid for paths of any length), then we replace the edges in the trail with a single edge $(x, z)$. If it does not exist then clearly there is no Eulerian circuit that uses $T$, since it is not even present in the graph. Clearly an Eulerian circuit on $G$ containing $T$ only exists if and only if an Eulerian circuit exists on the modified graph. Once we have replaced $T$ we examine any trails in $\mathcal{P}$ of the form $uxy$ or $yzv$ and replace them with new trails $uz$ and $xv$ respectively. Since a circuit following the trail $uxy$ must immediately follow on to $z$ if it also follows $T$, this maintains the equivalence of the superpath problem between the original instance of $G$ and $\mathcal{P}$ and this new modified instance.

The difficulty in the general problem comes from allowing multi-edges. In this case, we can still replace trails with edges, but when we come to update the remaining overlapping trails, we no longer have a single choice for each trail to use. If both $xyz$ and $yzv$ are present in $\mathcal{P}$, there is no requirement for the circuit to follow $xyzv$ if there are multiple copies of the edge $(y, z)$. The circuit may follow $xyz$ before returning to $y$ later in the circuit and using the second copy of $(y, z)$ to then follow $yzv$. From this difficulty, it is not clear, when replacing $xyz$ with $(x, z)$ whether we should update $yzv$, assuming it follows on from the first, or to leave it alone. Certainly we can not guarantee that making one choice or the other preserves the equivalence of the superpath problem on these two instances.

Looking at the case where all elements of $\mathcal{P}$ are length 2, we note that we have

3 cases to consider for any multi-edges. A multi-edge $(x, y)$, may be covered by

- No trails,
- Some number of trails of the form $xyz$ for some $z$ adjacent to $y$ (or $x$, but not both),
- Some number of trails of the form $xyz$ and $yxz'$ at once.

In the first case we do not need to consider this edge in any additional detail than in finding a normal Eulerian circuit.

Suppose we are in the second case and our edge $(x, y)$ is covered by some number of trails (w.l.o.g) of the form $xyz$. For each trail, we can replace one of the edges $(x, y)$ and (possibly one of) the edges $(y, z)$, with a single edge $(x, z)$ in $G$ and remove the trail from $\mathcal{P}$, since this single edge must be covered by any Eulerian circuit anyway. We can continue by replacing each other trail using the edge $(y, z)$ that we have replaced, and hence is of the form $yz\alpha$ by a new trail $xz\alpha$ unless $(z, \alpha)$ is also a multi-edge covered by multiple trails. In this case it is again unclear whether the trail must really be using the edge we have replaced. In this case, $(z, \alpha)$ must be an edge of the third type.

Therefore supposing we have only edges of the first two types, we have a polynomial time solution to whether a circuit exists. We replace the trails whenever possible, updating both the graph and the elements of $\mathcal{P}$. If not possible, the circuit does not exist. Once we are done, we are left with a graph and a series of trails of length 1. This is simply the problem of finding a normal Eulerian circuit, which is well known to be solvable in polynomial time.

For each edge of type 3 (or for ease of analysis, for every multi-edge) we take the following steps. We replace each such multi-edge $(x, y)$ with multiplicity $k$, with a new vertex $v$ and $k$ copies of $(x, v)$ and $(v, y)$ each. We replace any trails in $\mathcal{P}$ of the form $xyz$ with new trails $vyz$ and similarly, trails of the form $zxy$ with trails $zxv$. We now have no edges of type 3 and can hence solve the Eulerian superpath problem on this new graph and trail system which we denote by $G'$ and $\mathcal{P}'$.

Suppose a solution exists on the original graph, in other words an Eulerian circuit that respects all of the original trails. This implies a solution on our new graph by equating any instance in the circuit of a vertex progression of $xyx$ with $xvyvx$ in the new graph. Equally, we equate $yxy$, with $yvxvy$. We also equate $xyz$ with $xvyz$ and similarly in the other direction. It is clear that each edge in the original graph uses up one each of the two new edges, and so translates into an Eulerian circuit in the new graph.


Also, any trail in the modified graph $G'$ either derives from a trail in $G$ that did not cross a multi-edge and hence is unchanged or is of the form $vyz$ or $zxv$ where $v$ is a vertex added to $G'$ between some multi-edge $(x, y)$ and some trail existed in $G$ of the form $xyz$ or $zxy$. Since all trails in $\mathcal{P}$ were covered in $G$ by this assumed Eulerian circuit, we know, supposing without loss of generality that $xyz \in \mathcal{P}$ that $xyz$ is present as a subtrail in the circuit in $G$. However, by construction we equate the subtrail $xyz$ with $xvyz$ in generating the Eulerian circuit $G'$ which covers the new trail $vyz$ as required. Therefore the new $G'$ and $\mathcal{P}'$ Eulerian superpath instance is solved by this new circuit as required.

We also need to show the reverse, that we can construct an Eulerian circuit in $G$ that contains all of $\mathcal{P}$ from a solution to the superpath problem on $G'$ and $\mathcal{P}'$. Once we have applied the Eulerian superpath solution to $G'$, which we note contains no edges of type three, we must construct an Eulerian circuit in $G$ in polynomial time that still respects the original trails in $\mathcal{P}$.

We do so by using the same equating method as above for subtrails that use both $x$, $y$ and $v$. Any subtrail in the circuit that travels from $x$ to $v$ to $y$ or the

reverse is collapsed down to a single $(x, y)$ edge. We are left with one remaining case however to translate. When a trail is of the form $xvx$ (or equivalently $yvy$), it is not clear whether this can be translated directly into an Eulerian circuit in the original graph $G$.

To resolve this third case, we note firstly that since there are an equal number of edges from $v$ to each of $x$ and $y$, and trails of the previous types use an equal number of edges from each, there must also be an equal number of these third type on each side of $v$, i.e. for each trail of the form $k_1xvxk_2$, there is also one of the form $k_3yvyk_4$. See figure 3.13, bearing in mind that there may well be more than 2 sets of multiple edges between $x$ and $v$.

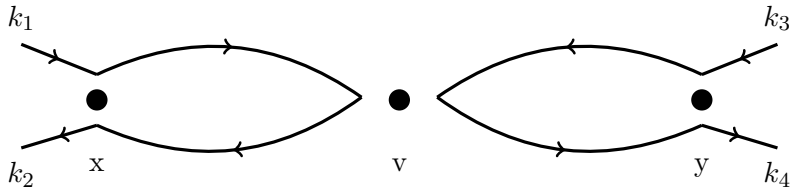



Figure 3.13: Subtrails of the Eulerian circuit in $G'$ that 'reflect' at $v$

We pair these 'reflected at $v$' trails up arbitrarily, and note that if this transition forms an Eulerian circuit (which by our assumption it does), then there must exist a trail that connects the trail leaving from $k_1$ to either $k_3$ or $k_4$ and a trail from $k_2$ to the other neighbour of $y$. Checking which pairs are connected will take less steps than the number of edges in the new graph (which is no more than twice the number of edges of $G$), and so can be done in polynomial time. The union of these trails may return to the $(x, v, y)$ edges but must cover all edges, since this forms an Eulerian circuit.

At this point we can apply the same logic we used in Theorem 3.6 when looking for turn-costed Eulerian circuits in 4-regular graphs. Although we do not necessarily have a degree 4 vertex at $v$, we do have an analogous situation, when considering just the 4 edges that enter $v$ in these two subtrail of the circuit. The important point is that we can switch this configuration of 'reflection' at $v$ with one of the

other two possible configurations for pairing these 4 edges at $v$ and we will still have an Eulerian circuit, assuming all other transitions are left unchanged.

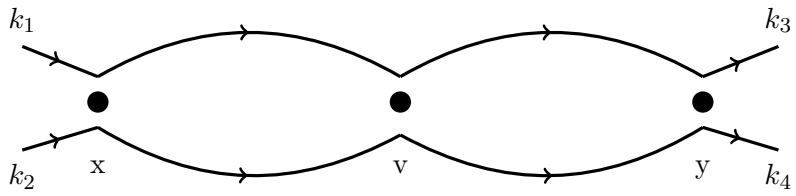



Figure 3.14: Modified subtrails that still form an Eulerian circuit.

We note that without loss of generality that supposing $k_1$ is linked to $k_4$ by some trail, we can replace this configuration by two trails $k_1xvyk_3$ and $k_2xvyk_4$ which will still be an Eulerian circuit. This change only affects trails in $\mathcal{P}'$ covering $xvy$ of which there are none so all trails in $\mathcal{P}'$ are still covered by this new Eulerian circuit. We can now equate these configurations back into an Eulerian circuit in $G$ using the same equivalence we used when going from $G$ to $G'$. All trails in $\mathcal{P}'$ are unchanged from those in $\mathcal{P}$ or of the form $vyz$ or $zxv$ for some multi-edge $(x, y)$ and as we saw for the reverse direction, this equivalence preserves the covering of these trails in the two circuits as required.

Based on this equivalence we can see that we solve ESP on $G$ and $\mathcal{P}$ if and only if we can solve it for $G'$ and $\mathcal{P}$. The process of transforming one solution into the other is polynomial time, since it at most requires to follow all the edges of the graph once for each duplicate edge in $G$, which is at most quadratic steps in $|E(G)|$ and we can solve ESP on $G'$ and $\mathcal{P}'$ in polynomial time, since there are no overlapping trails in $\mathcal{P}'$ by design.

With these facts it is clear that for any instance of the Eulerian superpath problem where all trails are of length 2, the problem can be solved in polynomial time as required.

## 3.5 Future work

The work in this chapter leaves a range of interesting questions still open. The most obvious is that of examining the complexity of Problem 1 (ETCP) on 6-regular graphs. Finding the precise boundaries at which a problem shifts from polynomial time solvability to being NP-complete gives great insights into the nature of the problem and where the difficulties arise in tackling it.

All attempts at using degree 6 vertices to form gadgets that model higher degree vertices have proved unsuccessful, and it seems unlikely that the methods used to tackle 8-regular graphs will apply. Equally however the methods used to find polynomial time solutions to the 4-regular case are very much intrinsic to degree 4 graphs, and cannot be used for higher degree vertices.

Another area of interest would be to tackle ETCP on graphs of bounded tree-width. Although some initial success seemed promising with 4-regular such graphs, this result made almost no use of the bounded tree-width of the graph and was superseded by the polynomial time approach for all 4-regular graphs. It seems very likely that ETCP will be solvable in polynomial time for all graphs of bounded tree-width. This seems likely because the key determining factor which prevents the choice of the lowest cost configuration at each vertex generating the lowest cost Eulerian circuit, is the presence of cycles in the graph, which can be closed by a greedy choice of configurations for the circuit. The tree decomposition of the graph can be generated in linear time and although we have not managed to find an algorithm using it to tackle the problem, it seems a promising approach.

The final open question this work presents is that of considering the Eulerian superpath problem with larger trails. The problem is known to be NP-complete in general, and we have shown that it is polynomial time solvable if all the trails are of length 1 (the standard Eulerian circuit problem) or 2. Considering instances of the problem in which all trails are of length 3 seems a natural next step and may already be NP-complete, but considering the more general case of bounded trail length would also be of interest and probably more applicable to DNA sequencing. Although it

seems very likely that this problem is NP-complete, finding faster algorithms to solve it would have direct and immediate applications. Currently algorithms are used (see [52] and [53]) which 'usually' (based on observed applications) run quickly, as the kind of difficult cases that induce super-polynomial time solvability in the general problem rarely occur in the natural processes used to generate the ESP instance. An algorithm with provable upper bounds on running times and good average-case performance would clearly be of interest.

Chapter 4

# *Adversarial resilience of matchings in bipartite random graphs*

## 4.1 Adverserial resilience

In the first chapter we talked about matchings and how they are one of the most fundamental structures in graph theory, both for their intrinsic properties and their applicability to both mathematical and applied problems.

As a reminder, a matching is a collection of disjoint edges, which can also be thought of as a pairing up of the vertices. A complete or perfect matching is one in which every vertex is covered by one of these edges, or in other words, every vertex has a unique partner it is paired up with.

Finding matchings in graphs is a well-studied problem and polynomial time algorithms are known to find maximum matchings, see for example [14].

Matchings can be important to find in a range of computer based situations such as pairing up tasks to processors or units capable of completing them or assigning unique hashed or encrypted keys to objects stored in a database. In these scenarios, in particular in those of computer security, a situation may arise in which some number of keys or encryption methods may face being compromised and the ability for a security scheme to remain secure in the face of such an adversary can be described as resilience.

Another area in which matchings have proved important is in the area of controllability which has become an increasingly active and interesting field in the light of recent papers, most prominently, the Nature paper by Liu, Slotine and Barabasi

[48]. Controllability refers to the ability, given a complex dynamical system, to control the overall state of the system, by modifying some subset of the system's components. One such model for this is known as network controllability in which each component of the system is modelled by a single vertex with some weight assigned to it. As time passes each vertex generates weight on its neighbours according to some weighted values assigned to its neighbouring edges. The aim is to find a minimum set of vertices, for which having complete control of the weights assigned to this control set, allows for complete control of the entire network, in particular, the ability to bring the network to any desired state in some finite time.

Liu, Slotine, and Barabási used a characterisation by Lin [47] of structural controllability to show how large matchings in bipartite graphs play a crucial role to obtain bounds on the number of nodes needed to control directed networks. They also estimated the fraction of edges in a matching drawn randomly from several classes of graphs in a non-rigorous way using the cavity method. Rigorous proofs for most of these results can be found in [10] and [59] . In particular the authors of these papers determine the size of a maximum matching in a random bipartite graph with a fixed degree distribution under some mild assumption on the degree distributions.

Several forms of resilience of graph properties have been considered for a range of different motivations and applications. In many ways random graph theory can be thought of as a kind of average case resilience argument, in which a large number of edges from the complete graph are removed, or have failed, and thresholds demonstrate the average failure rate required before a property is likely to no longer be present with high probability.

Instead of average case analysis, one can also consider worst case, or adversarial resilience. In this model, rather than allowing edges to be removed at random, we allow some adversary the ability to delete edges at will, with the aim of disrupting the desired property in the graph. This models both the ability of an attacker to disrupt a network, and also the worst possible case that could occur randomly.

In general allowing an adversary free reign in choosing which edges to destroy provides too much power, and almost any property can be disrupted. Such a result falls into the realm of extremal combinatorics, where properties are shown to exist for certain only when a very large number of edges are still present within a graph.

Limiting the abilities of an adversary provides more realistic results, since it is unlikely that any system an adversary would want to attack would be designed to give them such power over it. In these cases, two differing models present themselves. The first is the concept of global resilience, which includes the case where the adversary has freedom to choose. In this model, the adversary is given some fixed number of edges to destroy, anywhere in the graph, perhaps subject to some selection criteria. An example of such a selection criteria might be that at each stage the adversary may be presented with two edges selected at random, and may delete one of them. Such processes have attracted considerable interest, with the equivalent reverse process of adding an edge being the Achlioptas process, generating several interesting recent results (see, for example [55]).

An alternative model, and the one which we consider in this chapter, is that of local resilience. In this case, the adversary is presented with the ability to delete edges limited by local conditions, for example deleting half of the edges adjacent to each vertex. An excellent overview of local resilience and various results in this area can be found in the article "Local Resilience of Graphs" by Sudakov and Vu [63]. It is often surprising to discover that enforcing fairly simple and entirely local conditions can allow the adversary to eliminate completely global properties.

In this chapter we look at proving a local resilience result for matchings on a particular model of random bipartite graph. The use of this model was motivated by that used in a paper by Alan Frieze and Páll Melsted in a non-adversarial setting which has applications to the computer science problem of cuckoo-hashing [26]. In this paper, the authors demonstrated tight bounds on the relative size of the vertex sets in the bipartite graph for which a matching not only exists (with high probability) but can also be found with high probability by the Karp-Sipser algorithm.

This algorithm simply chooses an unclaimed edge incident to a degree one vertex where possible, otherwise choosing any unclaimed edge at random and adding it to the matching. Since this relatively simple algorithm has been shown to succeed for graphs on which a maximum matching exists, we were interested to see how granting the adversary the ability to isolate these degree one vertices would impact the bounds for the existence of a maximum matching. This led us to consider the case where the adversary is given the power to eliminate a single edge incident to each vertex in one of the partition sets of the bipartite graph.

## 4.2 Technical introduction

We are interested in the problem of finding the largest matching in a random bipartite graph after an adversary has deleted some edges. More precisely, we consider a random bipartite graph $G = (A \cup B, E)$ with $|A| = n$ and $|B| = (1 + \varepsilon)n$. Each vertex in $A$ is adjacent to $d$ neighbours chosen uniformly at random from $B$. We allow repetition, so this is a multigraph. An adversary is then able to remove a single edge adjacent to each vertex of $A$, with the aim of minimising the size of the largest matching.

We will show that for each $\eta > 0$ there exists a $d_0$ such that for all $d \geq d_0$, if $\varepsilon > (4+\eta)\log d/d^2$, then asymptotically almost surely the adversary cannot destroy all matchings of size $n$. If on the other hand $\varepsilon < (4-\eta)\log d/d^2$, then asymptotically almost surely the adversary can destroy all matchings of size $n$. (Asymptotically almost surely (a.a.s.) means with probability tending to 1 as $n$ tends to infinity.)

This problem is closely related to finding maximum strongly independent sets in $d$-uniform hypergraphs $\mathcal{H}^d_{(1+\varepsilon)n,n}$ on $(1 + \varepsilon)n$ vertices and $n$ edges. A strongly independent set in a $d$-uniform hypergraph is a set of vertices such that no hyperedge contains two or more vertices of this set. Here, the partition class $B$ corresponds to the vertices of the hypergraph and the vertices of $A$ correspond to the edges. Clearly, an adversary can isolate all vertices corresponding to a strongly independent set, and hence if the maximum size of a strongly independent set is $\beta$ then the adversery can

force the size of a maximum matching to be at most $(1+\varepsilon)n-\beta$. We will show that $\mathcal{H}^d_{(1+\varepsilon)n,n}$ a.a.s. contains a strongly independent set of size $(4-\eta)\log d/d^2 n$.

As a note on notation, asymptotically, we are interested in the probabilistic results as $n \to \infty$, and so by $o(1)$ we mean a function that tends to 0 as $n \to \infty$, but since we are also considering $d \to \infty$ (and at the same time $\varepsilon \to 0$), we may also require the use of little $o$ notation to denote the size of terms which do not depend on $n$, in which case we will label them $o_d(f(d))$ to indicate that the asymptotics depend on $d$ rather than $n$. As an example $\varepsilon = o_d(1)$, but neither depend on $n$ and hence are asymptotically constant in terms of $n$.

## 4.3 Upper bounds

We begin by finding an upper bound for the threshold for $\varepsilon$, such that for values of $\varepsilon$ above this bound the graph will, with high probability, contain a matching of size $n$. To prove this we consider Hall's theorem.

**Theorem 4.1**

*(Hall [32]) For a bipartite graph $G$ with partition sets $X$ and $Y$, a matching in $G$ of size $|X|$ exists if and only if,*

$$\forall X' \subseteq X, |\Gamma_Y(X')| \geq |X'|, \tag{4.1}$$

*where $\Gamma_Y(X')$ is the set of neighbours of $X'$ in $Y$.*

If there exists an $X' \subseteq X$ that does not satisfy (4.1), we call $X'$ a *witness* for violating Hall's condition.

We aim to give a bound on the probability of the adversary being able to restrict the size of a maximum matching. We do this by bounding the probability of the existence of a set that after deletion of edges by the adversary, could become a witness.

**Theorem 4.2**

*Let $d > 3$, and let $G$ be the random bipartite graph with partition sets $A$ and $B$ of size $n$ and $\tilde{n} = (1+\varepsilon)n$ respectively, and each vertex of $A$ chooses $d$ neighbours*

*uniformly at random from $B$. An adversary deletes a single edge incident to each vertex of $A$ to obtain $G'$. Then, for $\varepsilon > (4 + o_d(1))\frac{\log d}{d^2}$, a matching of size $n$ still exists in $G'$ with probability tending to 1 as $n \to \infty$.*

Proof Fix $\eta > 0$ and assume that $\varepsilon \geq (4+\eta)(\log d)/d^2$. Suppose that a matching of size $n$ does not exist in $G'$. By Hall's Theorem, at least one witness of Hall's condition exists. Consider a smallest such witness $S$, of size $s$, say. Its neighbourhood (again after deletion) must be of size $s-1$, or we could delete an element of $S$ and still have a witness of smaller size.

For two sets $S \subseteq A$ and $S' \subseteq B$ of sizes $s$ and $s-1$ respectively, if they form a witness and its neighbourhood after deletion, for each vertex of $S$, either all of the edges incident to that vertex in $G$ lie in $S'$, or exactly one lies outside (which the adversary then deletes).

The probability of a given edge incident to a vertex of $S$ also being incident to a vertex in $S'$ is $p := \frac{s-1}{\tilde{n}}$. Let $q := 1-p$. Given this, the probability that the $d$ edges incident to a vertex of $S$ satisfy this condition is $\mathbb{P}(\mathrm{Bin}(d,q) \leq 1) = (qd+p)p^{d-1}$.

The expected number $e_s$ of witnesses of size $s$ is therefore

$$e_s = \binom{n}{s}\binom{\tilde{n}}{s-1}\left((qd+p)p^{d-1}\right)^s. \tag{4.2}$$

We split the analysis in several cases depending on $s$ and $p$. For $s$ not too large, we use the inequality $\binom{n}{r} \leq \left(\frac{en}{r}\right)^r$ to bound

$$\binom{n}{s}\binom{\tilde{n}}{s-1} \leq \left(\frac{en}{s}\right)^s \left(\frac{e\tilde{n}}{s-1}\right)^{s-1} \leq \left(\frac{e\tilde{n}}{s-1}\right)^{2s} \leq \left(\frac{e}{p}\right)^{2s}.$$

Thus

$$e_s \leq (e/p)^{2s}\left((qd+p)p^{d-1}\right)^s = (e^2(qd+p)p^{d-3})^s. \tag{4.3}$$

**Case $s \leq \sqrt{n}$.** In this case $p \leq 1/\sqrt{n}$, so for $d > 3$, $e^2(qd+p)p^{d-3} \leq e^2 d/\sqrt{n}$. Hence

$$\sum_{s=1}^{\sqrt{n}} e_s \leq \frac{e^2 d/\sqrt{n}}{1 - e^2 d/\sqrt{n}} = o(1).$$

**Case $s \geq \sqrt{n}$ and $q \geq 2(\log d)/d$.** Note that the lower bound on $q$ implies an upper bound on $s$, say $s \leq s_0$, so we must also ensure we have covered all the cases

for $s$. In this case $e^2(dq+p)p^{d-3} \le e^2 d\exp(-q(d-3)) \le 1/2$ for large enough $d$. Thus

$$\sum_{s=\sqrt{n}}^{s_0} e_s \le \sum_{s=\sqrt{n}}^{\infty} \frac{1}{2^s} = 2^{-\sqrt{n}+1} = o(1).$$

**Case** $5/d \le q \le 2(\log d)/d$**.** Write the corresponding bounds on $s$ as $s_0 \le s \le s_1$. Note that $(qd+p)p^{d-1} = \mathbb{P}(\mathrm{Bin}(d,q) \le 1)$ is increasing as $q$ decreases. Thus

$$(qd+p)p^{d-1} \le (5+1)(1-5/d)^{d-1} \le 6e^{-\frac{5d-1}{d}}$$

which is less than $1/20$ for large enough $d$. For sufficiently large $d$ we also have $e^2/p^2 \le 10$, so $e^2(qd+p)p^{d-3} \le 1/2$ and

$$\sum_{s=s_0}^{s_1} e_s \le \sum_{s=s_0}^{\infty} \frac{1}{2^s} = o(1)$$

as in the previous case.

**Case** $1000(\log d)/d^2 \le q \le 5/d$**.** Write the corresponding bounds on $s$ as $s_1 \le s \le s_2$. From now on we will use the fact that

$$(qd+p)p^{d-1} \le 1 - \binom{d}{2}q^2p^{d-2} \le \exp\left(-\binom{d}{2}q^2p^{d-2}\right). \tag{4.4}$$

Note that $s-1 = p\tilde{n} = (1-q)\tilde{n}$ and in this case $\binom{n}{s} \le \binom{\tilde{n}}{s-1} = \binom{\tilde{n}}{q\tilde{n}}$. Thus

$$\begin{aligned} e_s &\le \binom{\tilde{n}}{q\tilde{n}}^2 \exp\left(-\tbinom{d}{2}q^2p^{d-2}\right)^s \\ &\le \left(\frac{e}{q}\right)^{2q\tilde{n}} \exp\left(-\tbinom{d}{2}q^2p^{d-2}\right)^s \\ &= \exp\left(-\tbinom{d}{2}q^2p^{d-2}s + 2q\tilde{n}\log(e/q)\right). \end{aligned}$$

Since $s \ge s-1 = p\tilde{n}$ we get

$$\sum_{s=s_1}^{s_2} e_s \le n\exp\left(-\left(\tbinom{d}{2}q^2p^{d-1} - 2q\log(e/q)\right)\tilde{n}\right).$$

Thus it suffices to show that $\binom{d}{2}q^2p^{d-1} - 2q\log(e/q)$ is bounded away from 0 independently of $n$. But this is clear as $p^{d-1} \ge 1/200$, $\log(e/q) \le 2\log d$, and $q \ge 1000(\log d)/d^2$.

**Case $q \le 1000(\log d)/d^2$ and $s \le (1 - e/d^2)n$.** In this case we have that $p^{d-1} = 1 - o_d(1)$. Note that

$$\tfrac{n-s}{n} = 1 - \tfrac{s}{n} \le 1 - \tfrac{s-1}{n} = 1 - p(1+\varepsilon) = q - p\varepsilon$$

and $q = 1 - p \ge 1 - n/\tilde{n} = \varepsilon/(1+\varepsilon) \ge e/d^2$. Hence by (4.2) and (4.4)

$$\begin{aligned} e_s &\le \binom{n}{n-s}\binom{\tilde{n}}{q\tilde{n}} \exp\left(-\tbinom{d}{2} q^2 p^{d-2}\right)^s \\ &\le \left(\frac{en}{n-s}\right)^{n-s} \left(\frac{e}{q}\right)^{q\tilde{n}} \exp\left(-\tbinom{d}{2} q^2 p^{d-2} s\right) \\ &\le (d^2)^{n(q-p\varepsilon)} (d^2)^{q\tilde{n}} \exp\left(-\tbinom{d}{2} q^2 p^{d-1} \tilde{n}\right) \\ &\le \exp\left(\left(-\tbinom{d}{2} q^2 p^{d-1} + 2(\log d)(2q - p\varepsilon)\right)\tilde{n}\right). \end{aligned}$$

Thus it suffices to show that $-\binom{d}{2} q^2 p^{d-1} + 2(\log d)(2q - p\varepsilon)$ is bounded away from 0 independently of $n$. But $(q - p\varepsilon)^2 \ge 0$ so

$$2(\log d)(2q - p\varepsilon) \le 2(\log d) q^2/p\varepsilon \le \tfrac{2}{4+\eta} d^2 q^2/p \le \left(1 - \tfrac{\eta}{10}\right)\tbinom{d}{2} q^2 p^{d-1}$$

for sufficiently large $d$.

**Case $q \le 1000(\log d)/d^2$ and $s \ge (1 - e/d^2)n$.** In this case we have $\binom{n}{n-s} \le \binom{n}{en/d^2}$. As above we have

$$\begin{aligned} e_s &\le (d^2)^{en/d^2} (d^2)^{q\tilde{n}} \exp\left(-\tbinom{d}{2} q^2 p^{d-1} \tilde{n}\right) \\ &\le \exp\left(\left(-\tbinom{d}{2} q^2 p^{d-1} + 2(\log d)(q + e/d^2)\right)\tilde{n}\right). \end{aligned}$$

Now $q \ge \varepsilon/(1+\varepsilon)$ so

$$2(\log d)(q + e/d^2) \le (2\log d) q (1 + \eta/10)$$

while

$$\binom{d}{2} q^2 p^{d-1} \ge d^2 q \varepsilon (1 - \eta/10)/2 \ge (2\log d) q (1 + \eta/2).$$

■

Thus the proof is complete.

## 4.4 Lower bounds

We now demonstrate a lower bound for the threshold, such that for values of $\varepsilon$ below this bound, with high probability, the adversary can ensure that a matching of size $n$ does not exist. We achieve this by demonstrating that a strategy for the adversary can succeed in isolating more than $\varepsilon n$ vertices from $B$ (which is of size $(1+\varepsilon)n$). This means that there will remain less than $n$ vertices in $B$ with neighbours in $A$ and so no matching of size $n$ can possibly occur.

Instead of looking at a matching in a bipartite graph we consider the random $d$-uniform hypergraph $\mathcal{H}^d_{\tilde{n},n}$ consisting of $\tilde{n} = (1+\varepsilon)n$ vertices and $n$ edges. The correspondence between these two models is simple. For each of the vertices of $A$, its neighbourhood consists of $d$ vertices in $B$, chosen uniformly and independently at random. Each neighbourhood of $d$ vertices in $B$ is chosen using an equivalent random process to the one that determines the hyperedges of $\mathcal{H}^d_{\tilde{n},n}$ and as such, treating these neighbourhoods as hyperedges on $B$, creates an auxiliary hypergraph, which is equivalent to $\mathcal{H}^d_{\tilde{n},n}$.

A small issue is that in the graph, the neighbourhood of a vertex in $A$ may not be of size $d$, since each vertex chooses uniformly at random from the vertices of $B$ for each of its $d$ edges, and hence (with small probability) may choose the same vertex twice, creating a multi-edge while the hypergraph model assumes all hyperedges are exactly of size $d$. In this case we can still generate a corresponding hyperedge in $\mathcal{H}^d_{\tilde{n},n}$ by simply choosing vertices at random to increase the size of the neighbourhood to $d$. Since every set of size $d$ is generated with equal probability by this method, the hypergraph generated this way and $\mathcal{H}^d_{\tilde{n},n}$ have the same probability distribution.

To address the equivalence in terms of translating our results, we firstly note that the number of vertices of $A$ whose incident edges contain a multi-edge (or equivalently, the number of vertices of $A$, whose neighbourhood is of size smaller than $d$) is asymptotically almost surely near to constant size.

To see this, consider that the probability that the edges incident to a single vertex in $A$ each have unique neighbours in $B$ is $\frac{\tilde{n}}{\tilde{n}}\frac{\tilde{n}-1}{\tilde{n}}\dots\frac{\tilde{n}-d+1}{\tilde{n}}$ and hence the probability

that a given vertex in $A$ has at least two incident edges incident to a single vertex in $B$ (or equivalently that the neighbourhood of the vertex in $A$ is smaller than $d$) is

$$1-\frac{\tilde{n}}{\tilde{n}}\times\frac{\tilde{n}-1}{\tilde{n}}\times\cdots\times\frac{\tilde{n}-d+1}{\tilde{n}}=1-\frac{\tilde{n}!}{(\tilde{n}-d)!(\tilde{n})^d}\leq 1-\left(\frac{\tilde{n}-d}{\tilde{n}}\right)^d$$
$$\leq 1-\left(1-\frac{d^2}{\tilde{n}}+\frac{d^3(d-1)}{2\tilde{n}^2}+\ldots\right)\leq\frac{d^2}{\tilde{n}}.$$

In light of this, if we allow $A_i$ to be the indicator variable which has value 1 if the neighbourhood of vertex $i\in A$ is of size less than $d$ and 0 otherwise, we have that $\mathbb{E}(A_i)\leq\frac{d^2}{\tilde{n}}$. Equally the variance of $A_i$ satisfies,

$$Var\left(A_i\right)\leq\frac{d^2}{\tilde{n}}-\frac{d^2}{\tilde{n}^2}+o\left(\frac{1}{\tilde{n}^2}\right).$$

The total number of vertices whose neighbourhood is of size less than $d$ is $\sum A_i$, and since the $A_i$ are independent, the expectation and variance of $\sum A_i$ are the sum of the expectations and variance respectively of the $A_i$s. This gives us

$$E\left(\sum A_i\right)\leq n\frac{d^2}{\tilde{n}}=\frac{d^2}{(1+\varepsilon)},$$

and

$$Var\left(\sum A_i\right)\leq n\left(\frac{d^2}{\tilde{n}}-\frac{d^2}{\tilde{n}^2}+o\left(\frac{1}{\tilde{n}^2}\right)\right)=\frac{d^2}{(1+\varepsilon)}+o\left(\frac{1}{n}\right).$$

By Chebyshev's inequality, the probability that $\sum A_i$ is more than a function $f(n)$ from its expected value is

$$\mathbb{P}\left(\left|\sum A_i-\mathbb{E}\left(\sum A_i\right)\right|\geq f(n)\right)\leq\frac{Var\left(\sum A_i\right)}{(f(n))^2}.$$

Letting $f(n)\to\infty$ as $n\to\infty$ however slowly we wish gives us that,

$$\mathbb{P}\left(\sum A_i\geq\frac{d^2}{(1+\varepsilon)}+f(n)\right)\leq\frac{d^2}{(1+\varepsilon)(f(n))^2+o\left(\frac{(f(n))^2}{n}\right)}\to 0,$$

as $n\to\infty$. In light of this, we can see that for any increasing function of $n$, the probability that the number of neighbourhoods containing multi-edges is larger than this function tends to 0, and such the number is asymptotically small.

For the lower bound we aim to find a strongly independent set in $B$ of size $(4 - o_d(1))\frac{\log d}{d^2}n$, which the adversary can then isolate since it corresponds to vertices in $B$, whose neighbourhoods consists of unique vertices in $A$. If some of the hyperedges that cover the elements of this set were generated from neighbourhoods containing multi-edges, then the vertex incident to the multi-edges cannot be isolated by the adversary (since this would require deleting two edges adjacent to a single vertex of $A$). However since there are a.a.s near to a constant number (and certainly $o(n)$) such multi-edges, this is less than the error term in the size of the independent set. The adversary can simply ignore any such vertices, and still isolate a sufficiently large set.

Formally, we aim to find a strongly independent set in $\mathcal{H}^d_{\tilde{n},n}$, that is, a set $I$ of vertices such that no two vertices of $I$ are contained in the same hyperedge. Clearly, given a strongly independent set $I \subseteq B$, the adversary can isolate the vertices of $I$ in the set $B$, since no two vertices in this set share a neighbour in $A$. The adversary can go through the vertices of $I$, eliminating their incident edges and never need to remove two edges incident to a single vertex in $A$. Hence if the maximum strongly independent set is of size bigger than $\varepsilon n$ then the adversary can ensure that a matching of size $n$ does not exist.

**Theorem 4.3**

*The maximum size of a strongly independent set in a random $d$-uniform hypergraph $\mathcal{H}^d_{\tilde{n},n}$ consisting of $\tilde{n}$ vertices and $n$ edges is asymptotically almost surely at least*

$$(4 - o_d(1))\frac{\log d}{d^2}n.$$

Proof We actually prove a stronger result by proving this result for the $d$-uniform hypergraph $\mathcal{H}^d_{\tilde{n},p}$ which has $\tilde{n}(= (1+\varepsilon)n)$ vertices and each subset of the vertices of size $d$ is a hyperedge with probability $p$ independently of the presence or absence of all other hyperedges. Here $p$ is such that the expected number of hyperedges equals $n$, that is,

$$\binom{\tilde{n}}{d}p = n. \tag{4.5}$$

This is indeed a stronger result. For example, suppose the maximum strongly independent set is a.a.s. of size at least $k$ in $\mathcal{H}^d_{\tilde{n},p}$. As this hypergraph has at least $n$ edges with probability bounded away from zero, the maximum size of a strongly independent set is a.a.s. at least $k$, even conditioned on the hypergraph having at least $n$ hyperedges. But then a.a.s. there is a strongly independent set of size at least $k$ in the hypergraph obtained from this graph by selecting at random $n$ hyperedges from this graph, which is distributed precisely as $\mathcal{H}^d_{\tilde{n},n}$.

We follow an argument of Frieze [27] suggested by Łuczak. We use a partition $P$ of the vertex set consisting of parts $P_1, \ldots, P_{n'}$ of size $m$ where $m$ grows asymptotically faster than $(\log d)^2$ but slower than $d^2/\log d$, that is, $m = o_d(d^2/\log d)$ and $m = \omega(\log^2 d)$. A set is a *$P$-set* if it is strongly independent and contains at most one vertex from each $P_i$ for $i \in \{1, \ldots, n'\}$. Let $\beta$ be the maximum size of a $P$-set. By Azuma's inequality, we have that

$$\mathbb{P}\left(|\beta - \mathbb{E}(\beta)| \geq t\right) \leq 2e^{-t^2/2n'} \tag{4.6}$$

using the martingale that exposes the hyperedges incident to $P_i$ at step $i$, $i = 1, \ldots, n'$, and noting that $\beta$ can change by at most 1 when we change hyperedges incident to a single $P_i$. This is because until we have exposed the hyperedges incident to $P_i$ we cannot know whether any vertex in $P_i$ can be added to our $P$-set and at most we can increase the size of the maximum $P$-set by 1 by adding a vertex from $P_i$ to an existing $P$-set formed of vertices from the already exposed $P_{i'}$.

Let $X_k$ be the random variable that counts the number of $P$-sets of size $k$. Using the second moment method we will now show that for $\eta = o_d(1)$, $\eta > 0$ and sufficiently large $d$,

$$\mathbb{P}(X_k > 0) > 2\exp\left(-2^{10}\left(\tfrac{\log d}{d}\right)^4 n\right) \tag{4.7}$$

where

$$k = (4-\eta)\frac{\log d}{d^2}n.$$

This implies the lower bound by setting $t = 2^6 \left(\frac{\log d}{d}\right)^2 n/\sqrt{m}$ as

$$\begin{aligned} t^2/2n' =& \frac{2^{12}n^2}{2n'm}\left(\frac{\log d}{d}\right)^4 = \frac{2^{12}n^2}{2\tilde{n}}\left(\frac{\log d}{d}\right)^4 \\ =& \frac{2^{12}n}{2(1+\varepsilon)}\left(\frac{\log d}{d}\right)^4 > 2^{10}\left(\frac{\log d}{d}\right)^4 n. \end{aligned}$$

From this it follows that the probability that $X_k > 0$ is larger than the probability that $\beta$, (which is the largest $k$ for which $X_k \neq 0$) lies further from the expectation than $t$. Thus $k$ must be at most $t$ greater than $\mathbb{E}(\beta)$. By Azuma's inequality, we know that $|k - \mathbb{E}(\beta)| < t$ with high probability and hence with high probability $\beta > k - 2t$. Hence if $m/(\log d)^2 \to \infty$, we have that for sufficiently large $d$,

$$\beta > (4 - \eta - 2^7(\log d)/\sqrt{m})\frac{\log d}{d^2}n \geq (4 - 2\eta)\frac{\log d}{d^2}n$$

as required.

We begin by observing that, by the second moment method,

$$\mathbb{P}(X_k > 0) \geq \frac{(\mathbb{E}X_k)^2}{\mathbb{E}(X_k^2)}. \tag{4.8}$$

We derive bounds on these values, firstly noting that

$$\mathbb{E}X_k = \binom{n'}{k} m^k (1-p)^{N_k}, \tag{4.9}$$

where $N_k$ is the number of $d$-sets in $\{1, \ldots, \tilde{n}\}$ intersecting a given $k$-set in at least two elements. Clearly,

$$N_k \leq \binom{k}{2}\binom{\tilde{n}-2}{d-2},$$

since this counts the number of pairs $(e, e')$ where $e$ is a $d$-set and $e' \subseteq e$ is a pair of elements from the given $k$-set.

For the second moment, consider two sets of size $k$ which overlap in a set of size $i$. We define $N_{k,i}$ as the number of $d$-sets which cover at least two vertices from each of these two $k$-sets. This can happen if at least two vertices from the overlap are covered by an edge or we cover one from the overlap and at least one from each of the two sets outside the intersection, or two in each set from outside the intersection.

Hence we have

$$\begin{aligned} N_{k,i} \leq & \binom{i}{0}\binom{k-i}{2}\binom{k-i}{2}\binom{\tilde{n}-4}{d-4} \\ & + \binom{i}{1}\binom{k-i}{1}\binom{k-i}{1}\binom{\tilde{n}-3}{d-3} \\ & + \binom{i}{2}\binom{k-i}{0}\binom{k-i}{0}\binom{\tilde{n}-2}{d-2}. \end{aligned}$$

Multiplying by $p$ and simplifying, we get,

$$\begin{aligned} pN_{k,i} &\leq n\Bigg(\binom{k-i}{2}^2 \frac{d(d-1)(d-2)(d-3)}{\tilde{n}(\tilde{n}-1)(\tilde{n}-2)(\tilde{n}-3)} \\ &\qquad + (k-i)^2 i \frac{d(d-1)(d-2)}{\tilde{n}(\tilde{n}-1)(\tilde{n}-2)} + \binom{i}{2}\frac{d(d-1)}{\tilde{n}(\tilde{n}-1)}\Bigg) \\ &\leq n\left(\frac{k^4d^4}{2\tilde{n}^4} + \frac{k^2id^3}{\tilde{n}^3} + \frac{i^2d^2}{2\tilde{n}^2}\right) \end{aligned}$$

as $(d-j)/(\tilde{n}-j) \leq d/\tilde{n}$ for $j \leq d < \tilde{n}$. Using that

$$\frac{kd}{\tilde{n}} \leq \frac{4\log d}{d},$$

we get

$$pN_{k,i} \leq n\left(\frac{1}{2}\left(\frac{4\log d}{d}\right)^4 + \left(\frac{4\log d}{d}\right)^2 \frac{id}{\tilde{n}} + \frac{i^2d^2}{2\tilde{n}^2}\right). \tag{4.10}$$

Now consider $\mathbb{E}(X_k^2)$. As $X_k^2$ counts ordered pairs of $k$-sets, both having at most one vertex in each $P_i$ and intersecting all hyperedges of $\mathcal{H}^d_{\tilde{n},p}$ in at most one vertex, we have

$$\mathbb{E}(X_k^2) = \sum_{i=0}^{k} \binom{n'}{k}\binom{k}{i}\binom{n'-k}{k-i} m^{2k-i}(1-p)^{2N_k - N_{k,i}}.$$

Indeed, there are $\binom{n'}{k}m^k$ ways of choosing the first $k$-set, $\binom{k}{i}$ ways of choosing the intersection of the $k$-sets, given this intersection has size $i$, and $\binom{n'-k}{k-i}m^{k-i}$ ways of choosing the remaining vertices of the second $k$-set. Also, there are $2N_k - N_{k,i}$ potential edges of $\mathcal{H}^d_{\tilde{n},n}$ that must be absent as there are $N_k$ such potential hyperedges intersecting each $k$-set in at least two vertices, and these two sets of hyperedges intersect in a set of size $N_{k,i}$.

We now return to bound (4.8), by first considering its reciprocal;

$$\begin{aligned}
\frac{\mathbb{E}(X_k^2)}{(\mathbb{E}X_k)^2} &= \sum_{i=0}^{k} \frac{\binom{n'}{k}\binom{k}{i}\binom{n'-k}{k-i} m^{2k-i}(1-p)^{2N_k-N_{k,i}}}{\binom{n'}{k}^2 m^{2k}(1-p)^{2N_k}} \\
&= \sum_{i=0}^{k} \frac{\binom{k}{i}\binom{n'-k}{k-i}}{\binom{n'}{k}} m^{-i}(1-p)^{-N_{k,i}} \\
&\le \sum_{i=0}^{k} \binom{k}{i} \left( \binom{n'-i}{k-i} / \binom{n'}{k} \right) m^{-i} \exp(pN_{k,i} + O(p^2 N_{k,i})) \\
&\le \sum_{i=0}^{k} \binom{k}{i} \frac{k(k-1)\dots(k-i+1)}{n'(n'-1)\dots(n'-i+1)} m^{-i} \exp(pN_{k,i} + o(1)) \\
&\le \sum_{i=0}^{k} \binom{k}{i} \left( \frac{k}{n'm} \right)^i \exp(pN_{k,i} + o(1)) \\
&\le 2 \sum_{i=0}^{k} \binom{k}{i} \left( \frac{k}{\tilde{n}} \right)^i \exp(pN_{k,i})
\end{aligned}$$

for sufficiently large $d$ and $n$. Here we have used the fact that $pN_{k,i} = O(n)$ by (4.10), so $p^2 N_{k,i} = O(np) = O(n^{1-d})$ as $n \to \infty$. We split the sum, dealing firstly with the case $i \le i_0$ where $i_0$ satisfies

$$\frac{i_0 d}{\tilde{n}} = \left( \frac{4 \log d}{d} \right)^2,$$

and hence (by (4.10))

$$pN_{k,i} \le 2n \left( \frac{4 \log d}{d} \right)^4.$$

Note that $(ek^2/i\tilde{n})^i$ is maximised at $i = k^2/\tilde{n}$ and hence

$$\begin{aligned}
2 \sum_{i=0}^{i_0} \binom{k}{i} \left( \frac{k}{\tilde{n}} \right)^i \exp(pN_{k,i}) &\le 2 \sum_{i=0}^{i_0} \left( \frac{ek^2}{i\tilde{n}} \right)^i \exp(pN_{k,i}) && (4.11) \\
&\le 2(i_0+1) \exp\left( \frac{k^2}{\tilde{n}} + 2n \left( \frac{4 \log d}{d} \right)^4 \right) \\
&\le \exp\left( 2n \left( \frac{4 \log d}{d} \right)^4 + O\left( \frac{(\log d)^2}{d^4} n \right) \right). && (4.12)
\end{aligned}$$

For $i \ge i_0$, (4.10) implies that

$$pN_{k,i} \le \frac{i^2 d^2}{2\tilde{n}} + \frac{24 i (\log d)^2}{d}.$$

We first consider the sum over $i > (1-\eta)k$. In this case write $j = k - i$. This gives us $j < \eta k$ and so, assuming $\eta$ is sufficiently small,

$$\begin{aligned}\exp(pN_{k,i}) &\le \exp(i^2d^2/2\tilde{n} + 24i(\log d)^2/d)\\ &\le \exp\big((k^2 - 2jk + j^2)d^2/2\tilde{n} + 24(k-j)(\log d)^2/d\big)\\ &\le \exp\big((k-2j)\big((1+2\eta^2)kd^2/2\tilde{n} + 25(\log d)^2/d\big)\big)\\ &\le \exp\big((k-2j)(1+3\eta^2)(4-\eta)(\log d)/2\big)\\ &\le (d^{2-\eta/3})^{k-2j} \le (\tilde{n}/k)^{k-2j} = (k/\tilde{n})^{k-2i}. \qquad (4.13)\end{aligned}$$

Hence

$$\begin{aligned}2\sum_{i\ge(1-\eta)k}\binom{k}{i}\left(\frac{k}{\tilde{n}}\right)^i\exp(pN_{k,i}) &\le 2\sum_{i=0}^{k}\binom{k}{i}\left(\frac{k}{\tilde{n}}\right)^{k-i} = 2\left(1+\frac{k}{\tilde{n}}\right)^k\\ &\le 2\exp(k^2/\tilde{n}) = \exp(O((\log d)^2 n/d^4)),\end{aligned}$$

which is much less than the contribution (4.12) above.

Now assume $i_0 \le i \le (1-\eta)k$. By applying (4.13) with $i = k$ we have

$$\exp\big(k^2d^2/2\tilde{n} + 24k(\log d)^2/d\big) \le (\tilde{n}/k)^k,$$

so by raising this to the power $i/k$ we have

$$\exp\big(ikd^2/2\tilde{n} + 24i(\log d)^2/d\big) \le (\tilde{n}/k)^i.$$

Hence

$$\left(\frac{k}{\tilde{n}}\right)^i\exp(pN_{k,i}) \le \left(\frac{k}{\tilde{n}}\right)^i\exp\big(i^2d^2/2\tilde{n} + 24i(\log d)^2/d\big) \le \exp(-(k-i)id^2/2\tilde{n}).$$

Hence it remains to bound the sum

$$2\sum_{i=i_0}^{(1-\eta)k}\binom{k}{i}\exp(-(k-i)id^2/2\tilde{n}).$$

For $\eta k \le i \le (1-\eta)k$, $(k-i)id^2/2\tilde{n} \ge \eta(\log d)k > k$, so $\exp(-(k-i)id^2/2\tilde{n}) < 2^{-k}$. Hence the sum over these terms is at most 1. For $i < \eta k$ we use that $\exp((k-i)d^2/2\tilde{n}) \ge \exp(3(\log d)/2) = d^{3/2}$. Thus

$$\binom{k}{i}\exp(-(k-i)id^2/2\tilde{n}) \le \left(\frac{ek}{id^{3/2}}\right)^i \le 1$$

for $i \geq i_0 > ek/d^{3/2}$. Thus the sum over $i_0 \leq i \leq (1-\eta)k$ contributes at most $O(i_0+1)$. Hence we obtain

$$\frac{\mathbb{E}(X_k^2)}{(\mathbb{E}X_k)^2} \leq \exp\left(2n\left(\frac{4\log d}{d}\right)^4 + O\left(\frac{(\log d)^2}{d^4}n\right)\right)$$

and so

$$\mathbb{P}(X_k > 0) \geq 2\exp\left(-2^{10}\left(\frac{\log d}{d}\right)^4\right)$$

as required. ∎

In light of this result, as mentioned before, we therefore immediately have the following,

**Corollary 4.4**

*Let $d > 3$, and let $G$ be the random bipartite graph with partition sets $A$ and $B$ of size $n$ and $\tilde{n} = (1+\varepsilon)n$ respectively, and each vertex of $A$ chooses $d$ neighbours uniformly at random from $B$. An adversary deletes a single edge incident to each vertex of $A$ to obtain $G'$. Then, for $\varepsilon < (4 + o_d(1))\frac{\log d}{d^2}$, there exists a strategy for the adversary to ensure, with probability tending to 1 as $n \to \infty$, that there is no matching of size $n$ in $G'$.*

In an earlier result using different methods, we showed a weaker threshold of $\varepsilon < (2 + o_d(1))\,\frac{\log d}{d^2}$ for the lower bound. Although this result has been superseded by the above, it is still of interest from an algorithmic point of view.

The above result provides an existence result of a set of edges the adversary can eliminate to reduce the maximum matching size, but we do not have a clear strategy for the adversary to pursue at each step, other than to identify the entirety of the independent set in advance. Although the result below is only proven for a weaker threshold, it does provide a clear algorithmic strategy for the adversary. In particular it proves that a greedy strategy of simply identifying any new vertex of roughly average degree, whose neighbourhood hasn't had any incident edges deleted already, to isolate at each step, can eliminate any matchings of size $n$. This algorithm is also producing a strongly independent set in the same manner as the previous

result but the gap in the thresholds demonstrates what is lost by pursuing a simple greedy approach of trying to isolate low degree vertices one at a time.

**Theorem 4.5**

*Let $G$ be the random bipartite graph with partition sets $A$ and $B$ of size $n$ and $\tilde{n} = (1+\varepsilon)n$ respectively, and each vertex of $A$ chooses $d$ neighbours uniformly at random from $B$. An adversary deletes a single edge incident to each vertex of $A$. For $\varepsilon < (2 + o_d(1)) \frac{\log d}{d^2}$, with probability tending to $1 - o_d(1)$ as $n \to \infty$, we provide a strategy for the adversary such that a matching of size $n$ will not exist after deletion.*

Proof Firstly we note that for every subset $S_B \subset B$ with $|S_B| = \gamma(1+\varepsilon)n$ for some $\gamma \in (0,1)$, the number of edges from $A$, incident to $S_B$ is

$$(1 + o_d(1))\gamma dn, \tag{4.14}$$

with probability tending to 1 as $n \to \infty$. This follows from observing that for each of the $nd$ edges from $A$, the probability that it is adjacent to a vertex in $S_B$ is $\gamma$, independently of each of the other edges. Hence the number of edges is binomially distributed with $nd$ trials and mean $\gamma nd$.

To prove this statement formally, we first bound the probability of an arbitrary set, $S_B \subset B$ of size $\gamma(1+\varepsilon)n$ having a number of adjacent edges that differ significantly from (4.14). Using a Chernoff bound (see for example, Corollary 2.3 in [36]) and denoting the number of edges between two sets $X$ and $Y$ by $e(X,Y)$ we find that for any $\beta > 0$,

$$\mathbb{P}(|e(T, S_B) - \gamma nd| > \beta\gamma nd) \leq 2e^{-\frac{\beta^2}{3}\gamma nd}.$$

We now consider the number of such sets, and using $H(p)$, the binary entropy function and the following inequality;

$$\binom{n}{k} \leq e^{nH\left(\frac{k}{n}\right)} \text{ where } H(p) = -p\log p - (1-p)\log(1-p),$$

find that the number of sets is equal to

$$\binom{(1+\varepsilon)n}{\gamma(1+\varepsilon)n} \leq e^{(1+\varepsilon)nH(\gamma)}.$$

The expected number of sets $S_B$ for which $|e(A, S_B) - \gamma nd| > \beta\gamma nd$ is therefore less than or equal to,

$$2e^{n\left(-\frac{\beta^2}{3}\gamma d+(1+\varepsilon)H(\gamma)\right)}.$$

Noting that $(1+\varepsilon)H(\gamma)$ is bounded above by 2 and $\frac{\beta^2}{3}\gamma d$ is unbounded as $d \to \infty$, we can see that this tends to 0 as $d \to \infty$ for any $\beta > 0$.

In light of this, the expected number of sets that satisfy $|e(A, S_B) - \gamma nd| > \beta\gamma nd$ tends to 0 for any $\beta > 0$, and hence every set $S_B \subset B$ of size $\gamma(1+\varepsilon)n$ must satisfy $e(A, S_B) = (1 + o_d(1))\gamma dn$ as required.

We now outline a greedy approach for the adversary to isolate vertices from $B$. Our aim is to isolate more than $\varepsilon n$ vertices from $B$, which means a matching of size $n$ cannot occur.

We begin by taking a vertex at random from $B$, and for each of its neighbours in $A$, the adversary eliminates the incident edges from that neighbour. The expected number of vertices in $B$ that have two edges incident to a single vertex in $A$ is $o(d/n)$ and hence, the probability of choosing such a vertex at any step (as long as the number of iterations is $o_d(n)$) is small and hence we discount this possibility, discarding any randomly chosen vertex that has this property. We iteratively define the sets $S_B$ and $S_A$ as the vertices we have isolated in $B$ and their neighbours in $A$ respectively.

After one step, with high probability, there will be $d + o_d(d)$ vertices in $S_A$ and a single vertex in $S_B$. We now iteratively look for vertices in $B \setminus S_B$ whose neighbourhood lies entirely in $A \setminus S_A$, in other words, those vertices not yet chosen to be added to $S_B$ who have no neighbours in $S_A$.

This process terminates once we can no longer find a vertex in $B \setminus S_B$ which does not have a neighbour in $S_A$, at which point we will have $|S_A| = (1 + o_d(1))d|S_B|$. Treating the edges from $S_A$ as independent selections from $B$, this problem can be seen as analogous to a coupon collector problem. Each edge from $S_A$ is a draw of a random vertex from $B$, and our process ends when there are no more vertices 'undrawn'. The only complication is that we are removing vertices from considera-

tion by adding them to $S_B$, and this selection process is not random. At each step, we are choosing (non-randomly) one element of $A$ that is definitely 'undrawn' and which we add to $S_B$, and then the remaining choices in the coupon collector analogy are selected randomly, by considering the neighbours of the neighbourhood of this chosen vertex (all of which are distributed uniformly at random on $B$). In light of this, if we can show that the number of remaining vertices not adjacent to any vertex in $S_A$ is greater than the number of vertices in $S_B$ (which is the number of vertices removed from selection, non-randomly by this process), then regardless of the choices we made in adding to $S_B$, there must, with high probability, be choices remaining that we can add.

Since we aim to construct $S_B$ to be of size greater than $\varepsilon n$, we need to show that, with high probability, the set of vertices with no neighbours in $S_A$ is also of size greater than $\varepsilon n$. This calculation will give us a bound on the size of $S_A$, which in turn gives a bound on the size of $S_B$.

Since each vertex of $S_A$ has degree $d$, the number of edges incident to the set is $d|S_A|$, which is therefore the number of coupons drawn in the equivalent coupon collector problem that we consider. It is known (see, for example Lemma 2 in [50] with $s = 1$) that with $n_0$ distinct coupons, that after $\alpha n_0$ iterations of the coupon collector process, the number of coupons remaining unseen is $e^{-\alpha} n_0 + o(n_0)$ with high probability. In our case $n_0 = (1 + \varepsilon)n$ and

$$\alpha = \frac{|S_A|d}{(1+\varepsilon)n}$$

and as such with high probability, there will be

$$e^{\frac{|S_A|d}{(1+\varepsilon)n}}(1+\varepsilon)n + o(n)$$

vertices whose neighbours lie entirely in $A \setminus S_A$. Since we aim for this to be greater than $\epsilon n$, we require the following;

$$e^{\frac{|S_A|d}{(1+\varepsilon)n}}(1+\varepsilon)n > \varepsilon n + o(n).$$

Rearranging, we get,

$$-|S_A|d > (1+\varepsilon)n \log\left(\frac{\varepsilon}{1+\varepsilon}\right) + o(1)$$

and hence we require,

$$|S_A| < \frac{(1+o_d(1))n}{d}\left(\log\left(\frac{1}{1+\varepsilon}\right) + o_d(1)\right). \tag{4.15}$$

To this end, we recall that $\varepsilon$ satisfies

$$\varepsilon < (2+o_d(1))\,\frac{\log d}{d^2},$$

and so the right hand side of 4.15 is greater than,

$$\frac{(1+o_d(1))n}{d}\left(\log\left(\frac{d^2}{2\log d}\right) + o_d(1)\right) = \frac{(2+o_d(1))n}{d}\log d.$$

Since this is smaller than the bound we require for $S_A$, if we have,

$$|S_A| < \frac{(2+o_d(1))n}{d}\log d,$$

then we have shown that with high probability, a set of size $\epsilon n$ exists in $B$ where each vertex in this set has neighbours only in $A \setminus S_A$ as required, and hence the process of generating $S_B$ will not terminate until $S_A$ reaches this threshold.

All that remains is to show that $S_B$ is of sufficient size at this point. Since $S_B$ is constructed such that all of its neighbours are in $S_A$ and distinct, we can see that the size of $S_A$ is precisely equal to the number of edges leaving $S_B$. We can use our observation (4.14), on the number of incident edges from any subset of $B$ to note that the size of $S_B$ must therefore be equal to $|S_A|/(1+o_d(1))d$. This gives us that once $S_A$ is approaching the bound we calculated, we must have

$$|S_B| = \frac{(2+o_d(1))n}{d^2}\log d > \epsilon n,$$

as required. ∎

Given this, the adversary can therefore isolate all of the vertices of $S_B$, leaving strictly less than $n$ vertices remaining in $B$, and hence no matching of size $n$ can occur.

## 4.5 Conclusion

Although we have proven a threshold for the lower and upper bounds which are asymptotically equal as $d \to \infty$, it seems likely that a threshold should exist for each $d$. In other words, we conjecture that there exists constants $c_d$ for $d \geq 3$ such that for all $\eta > 0$, if $\varepsilon > c_d + \eta$ then with high probability a matching of size $n$ can be found, while for $\varepsilon < c_d - \eta$ there is with high probability a strategy for the adversary that reduces the size of the maximal matching below $n$. If this conjecture is true then we know $c_d = (4 + o_d(1))(\log d)/d^2$.

Although we have identified the threshold for which an adversary can and cannot destroy the complete matching, there still remain a number of interesting open questions.

One that is of interest, is that of allowing the adversary greater or differing powers in modifying $G$. We initially considered the case where the adversary was able to delete $n$ edges globally, but found that for any fixed values of $d > 0$, this allowed the adversary to easily isolate a linear proportion of the vertices, while equally, a matching still exists that covers a linear proportion of the vertices. In both cases a simple greedy algorithm provides fairly simple bounds, but finding the exact size of the largest remaining matching seems challenging and certainly would require further insight in tackling.

Another problem to consider would be the case when $d$ is small. Although our arguments here are asymptotically tight for large $d$, it would be interesting to be able to say something about the graph's behaviour for small, fixed values of $d$. The use of our graph model was motivated by its use in [26] which analysed the size of the maximum matching in the same model but without an adversary removing edges, for all values of $d$, and it would be interesting to know what the behaviour of the maximum matching becomes for small values of $d$ once an adversary is introduced.

Finally, it would be interesting for values of $\varepsilon$ between the two thresholds demonstrated in section 4.4, to identify an algorithmic approach for the adversary to pursue in eliminating the maximum matching.

# *Bibliography*


[1] New graph theory from and for nanoconstruct design strategies (2012) : https://sites.google.com/site/nanoselfassembly [online]. Available from: `https://sites.google.com/site/nanoselfassembly` [cited Cited 29 Aug 2013]. 69

[2] Adleman, L. M. Molecular computation of solutions to combinatorial problems. *Science 266*, 11 (Nov. 1994), 1021–1024. Available from: `http://dl.acm.org/citation.cfm?id=189441.189442`. 69

[3] Alon, N., and Yuster, R. Threshold functions for H-factors. *Combinatorics, Probability and Computing 2*, 02 (1993), 137–144. Available from: `http://dx.doi.org/10.1017/S0963548300000559`. 9, 12, 17

[4] Andersen, L. D., Bouchet, A., and Jackson, B. Orthogonal a-trails of 4-regular graphs embedded in surfaces of low genus. *Journal of Combinatorial Theory, Series B 66*, 2 (1996), 232 – 246. Available from: `http://www.sciencedirect.com/science/article/pii/S0095895696900179`. 83

[5] Andersen, L. D., and Fleischner, H. The np-completeness of finding a-trails in eulerian graphs and of finding spanning trees in hypergraphs. *Discrete Applied Mathematics 59*, 3 (1995), 203 – 214. Available from: `http://www.sciencedirect.com/science/article/pii/0166218X9580001K`. 83

[6] Andersen, L. D., Fleischner, H., and Regner, S. Algorithms and outerplanar conditions for a-trails in plane eulerian graphs. *Discrete Applied Math-*

*ematics 85*, 2 (1998), 99 – 112. Available from: http://www.sciencedirect.com/science/article/pii/S0166218X97001418. 83

[7] Arkin, E. M., Bender, M. A., Demaine, E. D., Fekete, S. P., Mitchell, J. S. B., and Sethia, S. Optimal covering tours with turn costs. In *IN PROC. 13TH ACM-SIAM SYMPOS. DISCRETE ALGORITHMS* (2005), pp. 138–147. 70, 71

[8] Bent, S. W., and Manber, U. On non-intersecting eulerian circuits. *Discrete Applied Mathematics 18*, 1 (1987), 87 – 94. Available from: "http://www.sciencedirect.com/science/article/pii/0166218X8790045X". 66, 83

[9] Bollobás, B., and Thomason, A. Random graphs of small order. In *Random graphs '83 (Poznań, 1983)*, vol. 118 of *North-Holland Math. Stud.* North-Holland, Amsterdam, 1985, pp. 47–97. 9

[10] Bordenave, C., Lelarge, M., and Salez, J. Matchings on infinite graphs. *Probability Theory and Related Fields 157*, 1-2 (2013), 183–208. Available from: http://dx.doi.org/10.1007/s00440-012-0453-0. 109

[11] Chen, J., and Seeman, N. C. Synthesis from dna of a molecule with the connectivity of a cube. *Nature 350*, 6319 (04 1991), 631–633. Available from: http://dx.doi.org/10.1038/350631a0. 67

[12] Cook, S. A. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 1971), STOC '71, ACM, pp. 151–158. Available from: http://doi.acm.org/10.1145/800157.805047. 62, 72

[13] Dietz, H., Douglas, S. M., and Shih, W. M. Folding dna into twisted and curved nanoscale shapes. *Science 325*, 5941 (2009), 725–730. Available from: http://www.sciencemag.org/content/325/5941/725.abstract. 67

[14] Edmonds, J. Paths, trees, and flowers. *Canad. J. Math. 17* (1965), 449–467. Available from: `www.cs.berkeley.edu/~christos/classics/edmonds.ps`. 108

[15] Edmonds, J., and Johnson, E. L. Matching, euler tours and the Chinese postman. *Mathematical Programming 5*, 1 (1973), 88–124. 63

[16] Eiselt, H. A., Gendreau, M., and Laporte, G. Arc routing problems, part i: The chinese postman problem. *Operations Research 43*, 2 (2014/06/24 1995), 231–242. Available from: `http://dx.doi.org/10.1287/opre.43.2.231`. 70

[17] Ellis-Monaghan, J., McDowell, A., Moffatt, I., and Pangborn, G. Dna origami and the complexity of eulerian circuits with turning costs. *Natural Computing* (2014), 1–13. Available from: `http://dx.doi.org/10.1007/s11047-014-9457-2`. i

[18] Ellis-Monaghan, J., Pangborn, G., Beaudin, L., Miller, D., Bruno, N., and Hashimoto, A. Minimal tile and bond-edge types for self-assembling dna graphs. In *Discrete and Topological Models in Molecular Biology*, N. Jonoska and M. Saito, Eds., Natural Computing Series. Springer Berlin Heidelberg, 2014, pp. 241–270. Available from: `http://dx.doi.org/10.1007/978-3-642-40193-0_11`. 69

[19] Ellis-Monaghan, J. A., McDowell, A., Moffatt, I., and Pangborn, G. DNA origami and the complexity of Eulerian circuits with turning costs. *ArXiv e-prints* (Sept. 2013). 69, 82

[20] Ellis-Monaghan, J. A., and Moffatt, I. *Graphs on surfaces: dualities, polynomials, and knots*. Springer, 2013. 84

[21] Erdős, P., and Rényi, A. On the existence of a factor of degree one of a connected random graph. *Acta Math. Acad. Sci. Hungar. 17* (1966), 359–368. 9

[22] Feynman, R. P. There's Plenty of Room at the Bottom. Dec. 1959. 68

[23] Fleischner, J., and Fleischner, H. *Eulerian Graphs and Related Topics*, vol. 45 of *Annals of discrete mathematics*. Elsevier Science, 1990. Available from: http://books.google.co.uk/books?id=Y9e4ASBxNBwC. 70

[24] Fleischner, J., and Fleischner, H. *Eulerian Graphs and Related Topics*. No. pt. 1, v. 2 in Annals of discrete mathematics. Elsevier Science, 1991. Available from: http://books.google.co.uk/books?id=cDktskC2vQcC. 70

[25] Frieze, A. Private communication. 2012. 32

[26] Frieze, A., and Melsted, P. Maximum Matchings in Random Bipartite Graphs and the Space Utilization of Cuckoo Hashtables. *ArXiv e-prints* (Oct. 2009). 110, 129

[27] Frieze, A. M. On the independence number of random graphs. *Discrete Math. 81*, 2 (Apr. 1990), 171–175. Available from: http://dx.doi.org/10.1016/0012-365X(90)90149-C. 119

[28] Garey, M. R., and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979. 63

[29] Gerke, S., and McDowell, A. Non-Vertex-Balanced Factors in Random Graphs. *ArXiv e-prints* (Apr. 2013). 22, 42

[30] Gerke, S., and McDowell, A. Nonvertex-balanced factors in random graphs. *Journal of Graph Theory* (2014), n/a–n/a. Available from: http://dx.doi.org/10.1002/jgt.21805. i

[31] Hadlock, F. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing 4*, 3 (1975), 221–225. Available from: http://dx.doi.org/10.1137/0204019. 86

[32] Hall, P. On representatives of subsets. *Journal of the London Mathematical Society s1-10*, 1 (1935), 26–30. Available from: http://jlms.oxfordjournals.org/content/s1-10/1/26.short. 5, 112

[33] He, Y., Ye, T., Su, M., Zhang, C., Ribbe, A. E., Jiang, W., and Mao, C. Hierarchical self-assembly of dna into symmetric supramolecular polyhedra. *Nature 452*, 7184 (03 2008), 198–201. Available from: http://dx.doi.org/10.1038/nature06597. 67

[34] Hierholzer, C., and Wiener, C. Ueber die mglichkeit, einen linienzug ohne wiederholung und ohne unterbrechung zu umfahren. *Mathematische Annalen 6*, 1 (1873), 30–32. Available from: http://dx.doi.org/10.1007/BF01442866. 61, 63

[35] Högberg, B., Liedl, T., and Shih, W. M. Folding dna origami from a double-stranded source of scaffold. *Journal of the American Chemical Society 131*, 26 (2009), 9154–9155. PMID: 19566089. Available from: http://pubs.acs.org/doi/abs/10.1021/ja902569x. 67

[36] Janson, S., Luczak, T., and Ruciński, A. *Random Graphs*. Wiley-Interscience, 2000. 8, 9, 10, 14, 54, 55, 56, 125

[37] Johansson, A., Kahn, J., and Vu, V. Factors in random graphs. *Random Struct. Algorithms 33*, 1 (2008), 1–28. Available from: http://dx.doi.org/10.1002/rsa.20224. vi, 4, 7, 8, 10, 13, 16, 20, 22, 24, 25, 26, 27, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53

[38] Jonoska, N., Karl, S. A., and Saito, M. Three dimensional {DNA} structures in computing. *Biosystems 52*, 1–3 (1999), 143 – 153. Available from: http://www.sciencedirect.com/science/article/pii/S0303264799000416. 69

[39] Kapun, E., and Tsarev, F. De bruijn superwalk with multiplicities problem is np-hard. *BMC Bioinformatics 14*, S-5 (2013), S7. Available from: `http://dblp.uni-trier.de/db/journals/bmcbi/bmcbi14S.html#KapunT13`. 99

[40] Karp, R. M. Reducibility among combinatorial problems. In *Complexity of Computer Computations* (1972), pp. 85–103. 62, 63, 86

[41] Kim, J. H., and Vu, V. Concentration of multivariate polynomials and its applications. *Combinatorica 20*, 3 (2000), 417–434. 33

[42] Kotzig, A. Eulerian lines in finite 4-valent graphs and their transformations. *Theory of graphs, Proceedings of the Colloquium, Tihany, Hungary* (1966), pp. 219–230. 83

[43] Krivelevich, M. Embedding spanning trees in random graphs. *ArXiv e-prints* (July 2010). 8

[44] Kruskal, Joseph B., J. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society 7*, 1 (02 1956), 48–50. Available from: `http://www.jstor.org/stable/2033241`. 86

[45] L., E. Solutio problematis ad geometriam situs pertinentis. (The solution of a problem relating to the geometry of position). *Commentarii academiae scientiarum Petropolitanae 8, 1741, pp. 128-140* (1741). Available from: `http://www.math.dartmouth.edu/~euler/docs/originals/E053.pdf`. 2, 61

[46] Levin, L. A. Universal sequential search problems. *Problemy Peredachi Informatsii 9*, 3 (1973), 115–116. 62

[47] Lin, C.-T. Structural controllability. *Automatic Control, IEEE Transactions on 19*, 3 (Jun 1974), 201–208. 109

[48] Liu, Y.-Y., Slotine, J.-J., and Barabasi, A.-L. Controllability of complex networks. *Nature 473*, 7346 (May 2011), 167–173. Available from: http://dx.doi.org/10.1038/nature10011. 109

[49] Luczak, T., and Ruciński, A. Tree-matchings in graph processes. *SIAM J. Discrete Math. 4*, 1 (1991), 107–120. 9

[50] Mahmoud, H. M. Gaussian phases in generalized coupon collection. *Advances in Applied Probability 42*, 4 (12 "2010"), 994–1012. Available from: http://dx.doi.org/10.1239/aap/1293113148. 127

[51] Nangreave, J., Han, D., Liu, Y., and Yan, H. {DNA} origami: a history and current perspective. *Current Opinion in Chemical Biology 14*, 5 (2010), 608 – 615. Nanotechnology and Miniaturization/Mechanisms. Available from: http://www.sciencedirect.com/science/article/pii/S1367593110000931. 67

[52] Pevzner, P. A., Tang, H., and Waterman, M. S. An eulerian path approach to dna fragment assembly. *Proceedings of the National Academy of Sciences 98*, 17 (2001), 9748–9753. Available from: http://www.pnas.org/content/98/17/9748.abstract. 98, 99, 100, 107

[53] Pevzner, P. A., Tang, H., and Waterman, M. S. A new approach to fragment assembly in dna sequencing. In *Proceedings of the Fifth Annual International Conference on Computational Biology* (New York, NY, USA, 2001), RECOMB '01, ACM, pp. 256–267. Available from: http://doi.acm.org/10.1145/369133.369230. 98, 99, 100, 107

[54] Prim, R. C. Shortest connection networks and some generalizations. *Bell System Technical Journal 36*, 6 (1957), 1389–1401. Available from: http://dx.doi.org/10.1002/j.1538-7305.1957.tb01515.x. 86

[55] Riordan, O., and Warnke, L. Achlioptas process phase transitions are continuous. *ArXiv e-prints* (Feb. 2011). 110

[56] Rothemund, P. W. K. Folding dna to create nanoscale shapes and patterns. *Nature 440*, 7082 (03 2006), 297–302. Available from: http://dx.doi.org/10.1038/nature04586. 67

[57] Ruciński, A. Matching and covering the vertices of a random graph by copies of a given graph. *Discrete Math. 105*, 1-3 (1992), 185–197. Available from: http://dx.doi.org/10.1016/0012-365X(92)90141-2. 10, 12

[58] Ruciński, A., and Vince, A. Balanced extensions of graphs and hypergraphs. *Combinatorica 8*, 3 (1988), 279–291. 15

[59] Salez, J. Weighted enumeration of spanning subgraphs in locally tree-like graphs. *Random Structures & Algorithms 43*, 3 (2013), 377–397. Available from: http://dx.doi.org/10.1002/rsa.20436. 109

[60] Shih, W. M., Quispe, J. D., and Joyce, G. F. A 1.7-kilobase single-stranded dna that folds into a nanoscale octahedron. *Nature 427*, 6975 (02 2004), 618–621. Available from: http://dx.doi.org/10.1038/nature02307. 67

[61] Shmoys, D., Lenstra, J., Kan, A., and Lawler, E. *The Traveling Salesman Problem*. Wiley Interscience Series in Discrete Mathematics. John Wiley & Sons, 1985. Available from: http://books.google.co.uk/books?id=EPFQAAAAMAAJ. 69

[62] Spencer, J. H. Threshold functions for extension statements. *J. Comb. Theory, Ser. A 53*, 2 (1990), 286–305. 14

[63] Sudakov, B., and Vu, V. Local resilience of graphs. *ArXiv e-prints* (June 2007). 110

[64] Tutte, W. T. The factorization of linear graphs. *Journal of the London Mathematical Society 1*, 2 (1947), 107–111. 5

[65] Vu, V. On the concentration of multivariate polynomials with small expectation. *Random Struct. Algorithms 16* (July 2000), 344–363. Available from: http://dl.acm.org/citation.cfm?id=349225.349231. 33

[66] Vu, V. Concentration of non-lipschitz functions and applications. *Random Structures. Algorithms 20*, 3 (2002), 262–316. Available from: http://dx.doi.org/10.1002/rsa.10032. 33

[67] Zhang, Y., and Seeman, N. C. Construction of a dna-truncated octahedron. *Journal of the American Chemical Society 116*, 5 (1994), 1661–1669. Available from: http://pubs.acs.org/doi/abs/10.1021/ja00084a006. 67

[68] Zheng, J., Birktoft, J. J., Chen, Y., Wang, T., Sha, R., Constantinou, P. E., Ginell, S. L., Mao, C., and Seeman, N. C. From molecular to macroscopic via the rational design of a self-assembled 3d dna crystal. *Nature 461*, 7260 (09 2009), 74–77. Available from: http://dx.doi.org/10.1038/nature08274. 67

[69] Zitnik, A. Plane graphs with eulerian petrie walks. *Discrete Math. 244*, 1-3 (Feb. 2002), 539–549. Available from: http://dx.doi.org/10.1016/S0012-365X(01)00061-9. 83