Graph Colouring and Other Stories

Hannah Guggiari

Merton College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy in Mathematics

Trinity 2019

To Courtney

for always believing in me

Acknowledgements

I would like to start by thanking my supervisor, Professor Alex Scott, for all of his help and support over the past four years. Thank you for providing me with interesting and stimulating combinatorial problems to think about and for the useful discussions we've had during my time in Oxford. Under your excellent guidance, my ability to write and present complicated mathematics has also grown as is evidenced in part by this thesis.

I have been lucky to be part of the Oxford Combinatorics group, who have become both co-authors and friends. My DPhil has also given me opportunities to meet so many wonderful and interesting combinatorialists from around the world. A special mention goes to Alexander Roberts for his efforts to ensure that there is always cake after the seminars and for organising pie parties at my house.

Mathematical research can be lonely and frustrating at times and I am fortunate to have a large circle of friends to help me through the difficult times, in particular the bell ringers and Mertonians past and present. Thank you also to my brother, Sean Guggiari, for all of the little things – from the safe delivery of keys to the ironing of doilies – that have made my life easier over the past four years. It's been a pleasure spending time in Oxford with you.

I am grateful to my parents, John and Rosemarie Guggiari, for supporting me in everything that I do. You have always encouraged me to follow my heart and to pursue my dreams. Thank you for all of your encouragement throughout my DPhil.

Hopefully you now understand how someone can "research mathematics"!

My final thanks goes to my amazing husband, Courtney Spoerer, for his unending enthusiasm and support. At every step of the way in my DPhil journey, you have been by my side, celebrating with me every time a proof worked and encouraging me to keep trying whenever I hit brick walls with my thinking. Thank you for listening to me talk about more mathematics than you've ever wanted to know and for not complaining (too much!) when I pestered you with Python questions rather than googling the answers for myself. I could not have completed this thesis without you.

Abstract

This thesis investigates a variety of different problems within the field of Graph Theory. Half of these problems relate to edge-colourings of graphs; the other half are concerned with H-colourings, reconstruction, and pursuit and evasion games.

For graphs G and H, an H-colouring of G is a map $\psi : V(G) \to V(H)$ such that $ij \in E(G) \Rightarrow \psi(i)\psi(j) \in E(H)$. The number of H-colourings of G is denoted by $\hom(G, H)$. In Chapter 2, we prove the following: for all graphs H and $\delta \geq 3$, there is a constant $\kappa(\delta, H)$ such that, if $n \geq \kappa(\delta, H)$, the graph $K_{\delta,n-\delta}$ maximises the number of H-colourings among all connected graphs with n vertices and minimum degree δ. This answers a question of Engbers [20]. We also disprove a conjecture of Engbers [19] concerning the graph G that maximises the number of H-colourings when the assumption of the connectivity of G is dropped. In the final part of Chapter 2, we let H be a graph with maximum degree k and show that, if H does not contain the complete looped graph on k vertices or $K_{k,k}$ as a component and $\delta \geq \delta_0(H)$, then the following holds: for n sufficiently large, the graph $K_{\delta,n-\delta}$ maximises the number of H-colourings among all graphs on n vertices with minimum degree δ. This partially answers another question of Engbers [20].

In Chapter 3, we show that, in every 3-edge-colouring of the complete graph on $\mathbb{N}$, there exists a partition of $\mathbb{N}$ into at most 2 connected monochromatic subgraphs. Erdős, Gyárfás and Pyber [22] attribute this result to Nagy and Szentmiklóssy but, to our knowledge, a proof of it has never previously appeared in print.

For every $n \in \mathbb{N}$ and $k \geq 2$, it is known that every k-edge-colouring of the complete graph on n vertices contains a monochromatic connected component of order at least $\frac{n}{k-1}$. For $k \geq 3$, it is known that the complete graph can be replaced by a graph G with $\delta(G) \geq (1-\varepsilon_k)n$ for some constant ε_k. In Chapter 4, we show that the maximum possible value of ε_3 is $\frac{1}{6}$. This disproves a conjecture of Gyárfas and Sárközy [32].

Chapter 5 concerns edge-colourings of $\vec{K}_{\mathbb{N}}$, the complete symmetric digraph on the positive integers. Answering a question of DeBiasio and McKenney [16], we construct a 2-edge-colouring of $\vec{K}_{\mathbb{N}}$ in which every monochromatic path has density 0. We then ask what happens if we restrict the length of monochromatic paths in one colour and show that, in every 2-edge-colouring that does not have a directed path with r edges in the first colour, there is directed path in the second colour with density at least $\frac{1}{r}$. We also extend this result to more colours and prove a related stability result.

The deck of a graph G is the multiset of (unlabelled) subgraphs $\{G-v : v \in V(G)\}$. The subgraphs $G - v$ are referred to as the cards of G. Brown and Fenner [8] recently showed that, for $n \geq 29$, the number of edges of a graph G can be computed from any deck missing 2 cards. In Chapter 6, we improve this result by showing that, for sufficiently large n, the number of edges can be computed from any deck missing at most $\frac{1}{20}\sqrt{n}$ cards.

In Chapter 7, we consider the pursuit and evasion game, Cat and Mouse, which is played on a connected graph G with n vertices. The mouse moves to vertices $m_1, m_2, \ldots$ of G where m_i is in the closed neighbourhood of m_{i-1} for $i \geq 2$. The cat tests vertices $c_1, c_2, \ldots$ of G without restriction and is told whether the distance between c_i and m_i is at most the distance between c_{i-1} and m_{i-1}. The mouse knows the cat's strategy, but the cat does not know the mouse's strategy. We show that the cat can determine the position of the mouse up to distance $O(\sqrt{n})$ within finite time and that this bound is tight up to a constant factor. This disproves a conjecture of Dayanikli and Rautenbach [13].

Statement of Originality

This thesis contains no material which has been submitted for any other degree.

All of the results presented in this thesis are entirely my own work with the following exceptions:

- Chapter 2 is based on joint work with Alex Scott [29].
- Chapter 4 is based on joint work with Alex Scott [30].
- Chapter 5 is my own work and based on the manuscript [27] with the exception of Section 5.7, which is based on joint work with Carl Bürger, Louis DeBiasio and Max Pitz [9].
- Chapter 6 is based on joint work with Carla Groenland and Alex Scott [26].
- Chapter 7 is based on joint work with Alexander Roberts and Alex Scott [28].

Contents

List of Figures

Chapter 1

Introduction

Graphs are a mathematical way of showing connections between objects. In its abstract form, a *graph* $G = (V, E)$ consists of a set of *vertices* V and a multi-set of *edges* $E \subseteq \{\{u, v\} : u, v \in V\}$. An edge is a pair of vertices and we normally insist that these vertices are distinct (i.e. there are no loops). For convenience, we will write uv for the edge $\{u, v\}$.

A graph is called *simple* if there is at most one edge between each pair of vertices. All graphs in this thesis are simple and without loops unless otherwise indicated. The *order* of a graph is the number of vertices $|V|$ and the *size* is the number of edges $|E|$.

Although graphs are an abstract concept, their applications are rich and diverse because the vertices can represent any object and the edges any connection between them. Graphs were first introduced by Euler in the 18^{th} century to solve the infamous "Seven bridges of Königsberg" problem: by representing the islands as vertices and the bridges as edges, Euler showed that it was impossible to find a walk through the city which crossed each of the seven bridges exactly once. Since then, graph theory has repeatedly proven itself to be a useful tool in areas ranging from computing to linguistics. Here are just a few examples:

- Sociology: when exploring how rumours are spread through a community, a graph can be used to model the social network with the vertices representing people and the edges connecting people who are friends with each other.

- Transport: in order to minimize the costs of transporting goods between various towns, we may consider the graph which represents the road network - each town is assigned a vertex and two towns are connected by an edge if there is a direct road between them. We then assign a weight to each edge of the graph which represents the costs of travelling along that road.

- Biology: to investigate how to stop the spread of TB in cattle, a graph can be used to model the location of different farms. Each farm is represented by a vertex and two farms are connected if they have cattle in adjacent fields.

1.1 H-Colourings

Although graphs are very versatile structures, sometimes we need to do more than just consider the connections between vertices in order to solve a problem. One technique that can be helpful is colouring the vertices. A *proper vertex-colouring* is a way of assigning a colour to each vertex in such a way that vertices which are connected by an edge receive different colours. Given a graph G, a common graph theory problem is to colour G using as few colours as possible. The minimum number of colours needed to properly colour G is known as the *chromatic number* of G and denoted by $\chi(G)$.

Vertex-colourings are typically used in scheduling problems. Consider a school which needs to schedule end-of-year exams for its pupils. Obviously, it is not possible for a pupil to sit two exams at the same time. In order to minimise cheating, the school would like all pupils doing a particular exam to sit it at the same time. This problem can be modelled as a graph with each exam being a vertex and two exams being connected by an edge if a student needs to sit both of them. Any proper vertex-colouring of this graph represents a viable exam timetable with the colour classes giving the different exam timeslots. Two exams with the same colour can be scheduled at the same time because no student needs to take both. The chromatic number of this graph represents the smallest number of timeslots needed to schedule all of the exams.

Given two graphs G and H, a *graph homomorphism* from G to H is a map $\psi : V(G) \to V(H)$ such that $ij \in E(G) \Rightarrow \psi(i)\psi(j) \in E(H)$. An *$H$-colouring* of G is a graph homomorphism from G to H. Another way to view a proper vertex-colouring of G using q colours is as a graph homomorphism from G to K_q and so finding H-colourings of G is simply an extension of finding proper vertex-colourings of G. We denote the number of H-colourings of G by $\hom(G, H)$

Given a class of graphs $\mathcal{G}$ and a fixed graph H, it is natural to ask which $G \in \mathcal{G}$ has the most H-colourings. Various classes of graphs have been considered (see Cutler [11] for a survey).

A number of authors have studied the class of all δ-regular graphs for fixed δ. Galvin [25] conjectured that, for any H, if G was a δ-regular graph on n vertices, then

$$\hom(G, H) \leq \max\left\{\hom(K_{\delta,\delta}, H)^{n/2\delta}, \hom(K_{\delta+1}, H)^{n/(\delta+1)}\right\}.$$

If this were true, it would mean that, whenever $2\delta(\delta+1)|n$, either $\frac{n}{2\delta}K_{\delta,\delta}$ or $\frac{n}{\delta+1}K_{\delta+1}$ maximises the number of H-colourings among all δ-regular graphs on n vertices (there may be other extremal graphs). Galvin's conjecture was shown to be false by Sernau [52], who produced an infinite family of counterexamples as follows: fix δ and any simple loopless graph H with no $(\delta+1)$-clique. Take any connected δ-regular graph G on $n < 2\delta$ vertices with $\hom(G, H) > 0$. Sernau proved that there exist $k \in \mathbb{N}$

such that

$$\hom(G,kH) > \max\left\{\hom(K_{\delta+1},kH)^{n/(\delta+1)}, \hom(K_{\delta,\delta},kH)^{n/2\delta}\right\}$$

and hence that Galvin's conjecture was false.

Loh, Pikhurko and Sudakov [40] took a different direction and investigated the class of all graphs with n vertices and m edges. They found that, for each $q \geq 4$ and sufficiently large $m < \frac{n^2}{q \log q}$, the graphs with the most q-colourings are complete bipartite graphs minus the edges of a star together with some isolated vertices. For $q = 3$ and sufficiently large $m \leq \frac{n^2}{4}$, the graphs with the most q-colourings are either as for $q \geq 4$ or complete bipartite graphs with one extra vertex and edge.

In Chapter 2, we consider the class of all graphs with minimum degree at least δ. This class was studied by Engbers [19, 20] who raised a number of questions and conjectures. We answer two of these questions and provide a partial answer to a third.

1.2 Edge-Colourings

Rather than colouring the vertices of a graph, we can instead colour the edges. An *edge-colouring* is an assignment of a colour to each edge of a graph. If all edges which have a vertex in common receive different colours, then it is a *proper edge-colouring*. The minimum number of colours needed to produce a proper edge-colouring of a graph G is known as the *edge chromatic number* and denoted by $\chi'(G)$.

Sometimes it is more helpful to consider improper edge-colourings. It is a well known fact that, among any 6 people, there are either 3 mutual strangers or 3 mutual acquaintances. This can be modelled as a complete graph where each vertex represents a person. If two people are strangers, then the edge between them is blue; if they are acquaintances, then the edge between them is red. A simple lemma from

Ramsey Theory tells us that, in any red and blue edge-colouring of this graph, there is a monochromatic triangle (red means 3 mutual acquaintances and blue means 3 mutual strangers). This type of edge-colouring problem is well-studied and there are many variations.

An old remark of Erdős and Rado noted that, for any graph G, either G or its complement $\overline{G}$ is connected (see [22] for a mention of this remark). As this holds true for both finite and infinite graphs, it follows that, in any edge-colouring with 2 colours of the complete graph on n vertices, there is a monochromatic spanning tree. We also find that any edge-colouring with 2 colours of the complete graph on the positive integers contains a monochromatic subgraph containing all of the vertices.

A natural extension is to consider what happens for an edge-colouring with k colours. Erdős, Gyárfás and Pyber [22] made the following conjecture about finite complete graphs and showed that it was true for $k = 3$.

Conjecture 1.1. *Fix $k \in \mathbb{N}$. For any $n \in \mathbb{N}$ and any k-edge-colouring of K_n, there exists a partition of the vertices into at most $k-1$ connected monochromatic subgraphs.*

Tuza [54] proved this conjecture holds for $k = 4$ and $k = 5$. Although there has been a lot of focus on this question, it remains open for $k \geq 6$.

For the infinite case, the situation is a bit clearer. Hajnal, Komjáth, Soukup and Szalkai [34] showed that, for any edge-colouring with k colours of the complete countably infinite graph, there is a partition of the vertices into at most k monochromatic connected subgraphs. It is believed that Conjecture 1.1 also holds for infinite graphs. In Chapter 3, we show that this is indeed true for $k = 3$.

Another way in which Erdős and Rado's remark can be extended is by considering the guaranteed size of the largest monochromatic component in any k-edge-colouring of the complete graph. Gyárfás [31] proved that there always exists a monochromatic component of order at least $\frac{n}{k-1}$. A logical question to ask is whether it is necessary for the graph to be complete in order for this result to hold. Would a graph with large minimum degree suffice?

For two colours, Gyárfás and Sárközy [33] showed that we need a complete graph to guarantee that a monochromatic spanning subgraph exists. However, the situation for 3 or more colours is different. In this case, DeBiasio, Krueger and Sárközy [14] showed that it is possible to remove some edges from a complete graph and still obtain a monochromatic component of order $\frac{n}{k-1}$ in every k-edge-colouring. Gyárfás and Sárközy [32] suggested the following conjecture.

Conjecture 1.2. *Fix $k \geq 3$. Let G be a graph with n vertices and $\delta(G) \geq \left(1 - \frac{k-1}{k^2}\right) n$. If the edges of G are k-coloured, then there exists a monochromatic component of order at least $\frac{n}{k-1}$.*

In Chapter 4, we will disprove this conjecture for $k = 3$. We will also find the optimal bound for the minimum degree that guarantees a graph on n vertices will contain a monochromatic component of order $\frac{n}{k-1}$ in any k-colouring of its edges.

A third natural extension to consider is how dense a monochromatic path can be if we colour the edges of the countably infinite complete graph with 2 colours. Rado [48] showed that there is a partition of the vertices into 2 disjoint paths of distinct colours. Elekes, Soukup, Soukup and Szentmiklòssy [18] have recently extended this result to the complete graph on ω_1 where ω_1 is the smallest uncountable cardinal. For this setting, the notion of a path needs to be extended: a graph $P = (V, E)$ is a *path* if there is a well-ordering $\prec$ on V such that $\{w \in N_P(v) : w \prec v\}$ is $\prec$-cofinal below v for every $v \in P$ (i.e. for every $u, v \in P$ with $u \prec v$, there exists $w \in N_P(v)$ such that $u \preceq w \prec v$). It is noted in [18] that this definition coincides with the standard definition of a path for finite graphs.

For any $n \in \mathbb{N}$, we write $[n]$ for the set $\{1, \dots, n\}$. We define the *upper density* of a set $A \subseteq \mathbb{N}$ to be

$$\bar{d}(A) = \limsup_{n\to\infty} \frac{|A \cap [n]|}{n}$$

For a graph or digraph G with vertex set $V(G) \subseteq \mathbb{N}$, we define the *upper density* of G to be that of $V(G)$. This is a useful notion when describing the "length" of an infinite path. Note that, for us, an infinite path has only one infinite end. Erdős and Galvin [21] showed that if the edges of the complete countably infinite graph are coloured with 2 colours, then there exists a monochromatic infinite path with upper density at least $\frac{2}{3}$.

All of the graphs we have considered so far have been *undirected* because the order of the vertices in an edge has not mattered. It is often helpful to think about *directed* graphs (commonly referred to as *digraphs*) where the edges uv and vu are distinct for vertices u and v. For example, when modelling a road network where the edges are roads, we require directed edges if there are one-way streets. Pictorially, we represent directed edges by using arrows between the vertices instead of lines.

For digraphs, we can also consider the length of the longest monochromatic path. Here it makes sense to consider the complete symmetric digraph (between every pair of vertices there are two edges, one in each direction) and to look for directed monochromatic paths. In the finite case, Raynaud [49] showed that, in any 2-edge-colouring, the vertices of the complete symmetric digraph can be partitioned into two monochromatic directed paths.

In Chapter 5, we are interested in the density of monochromatic directed paths in the countably infinite complete symmetric digraph. We present a 2-edge-colouring in which every monochromatic path has density 0, which answers a question of DeBiasio and McKenney [16]. Assuming that the colours in a 2-edge-colouring are red and blue, we then show that, in every 2-edge-colouring with no red directed path of length r, there is a blue directed path with density at least $\frac{1}{r}$. Finally, we extend this result to more colours and prove a related stability theorem.

1.3 Reconstruction

Given a graph G and any vertex $v \in V(G)$, the *card* $G - v$ is the subgraph of G obtained by removing the vertex v and all edges incident to v. The multiset of all unlabelled cards of G is called the *deck*, $\mathcal{D}(G)$, and has size n.

One of the long-standing open problems in the field of graph theory is whether graphs are uniquely determined by their deck of cards. Clearly this is not true if we choose a graph with just 2 vertices because both graphs on two vertices have the same subgraphs (and hence their decks are the same). However, no one has yet found an example of two non-isomorphic graphs with at least 3 vertices which have the same deck. Kelly and Ulam [37, 38, 55] proposed the following *Reconstruction Conjecture.*

Conjecture 1.3. *For $n > 2$, two graphs G and H of order n are isomorphic if and only if $\mathcal{D}(G) = \mathcal{D}(H)$.*

The Reconstruction Conjecture is still open, although it is known to be true for certain classes of graphs (for example trees [37]). Moreover, almost every graph can be reconstructed [3, 42, 43]. For more background, see [1, 4, 5, 39, 45].

A potentially easier problem is to determine which parameters of a graph can be calculated from its deck; such parameters are said to be *reconstructible.* Given a full deck of cards $\mathcal{D}(G)$, a number of parameters are known to be reconstructible including:

- The order of G: each card in $\mathcal{D}(G)$ is created by deleting one vertex of G. Therefore $|\mathcal{D}(G)| = |G|$.

- The size of G: the card $G - v$ is missing exactly $d(v)$ edges. By summing over the edges in all of the cards, we find that

$$
\begin{aligned}
\sum_{v\in V(G)} e(G-v) &= \sum_{v\in V(G)} (e(G)-d(v)) \\
&= |G|e(V) - \sum_{v\in V(G)} d(v) \\
&= (|G|-2)e(G). && \text{[Handshaking Lemma]}
\end{aligned}
$$

- The degree sequence: the card $G-v$ has $e(G-v)=e(G)-d(v)$. As $e(G)$ is reconstructible, it is straightforward to calculate the degree sequence once the size of G has been determined.

- Whether G is connected: if at least two cards of G are connected, then G must also be connected. Conversely, if G is connected, then it contains a spanning tree with at least two leaves. Deleting a leaf from G produces a connected subgraph.

In fact, some parameters are reconstructible even if there is not a full deck of cards. For example, Bowler, Brown, Fenner and Myrvold [7] showed that we only need $\left\lfloor \frac{n}{2} \right\rfloor + 2$ cards to determine whether the graph is connected. Myrvold [44] also found that the degree sequence (and hence the size of the graph) is reconstructible from $n-1$ cards.

In Chapter 6, we focus on reconstructing the size of a graph from an incomplete deck of cards. Brown and Fenner [8] recently proved that, for $n \geq 29$, only $n-2$ cards are needed. We improve upon this result by showing that, for n sufficiently large, the number of edges can be reconstructed if at most $\frac{1}{20}\sqrt{n}$ cards are missing from the deck.

1.4 Pursuit and Evasion Games

As well as having many practical applications, graphs are also a good source of entertainment. There are a wide variety of different games which can be played on graphs. The most common games are of the “pursuit and evasion” kind. There are two players; the first player (the *pursuer*) is trying to find the location of the second player (the *evader*) who is trying to avoid being found. The players take alternate turns to choose a vertex of the graph which represents their current location. The evader is normally restricted to only moving along edges of the graph. The movement of the pursuer depends on the game, though they are usually given some information about the current location of the evader in order to help inform their next choice of vertex.

One of the first and most widely studied pursuit and evasion games is Cops and Robbers, which was introduced by Nowakowski and Winkler [46] and independently by Quilliot [47]. The pursuer controls a set of cops and the evader is a robber. At the start of the game, the cops are placed on vertices of the graph. The robber then chooses a vertex. The players take alternate turns with the cops playing first. On the cops’ turn, each cop moves to a vertex within the closed neighbourhood of their current position and multiple cops may occupy the same vertex. The robber plays in a similar fashion, moving to a vertex within its current closed neighbourhood. The cops win if one of them is on the same vertex as the robber after a finite period of time; otherwise the robber wins. In this game, the players have perfect information - they know the location of both the cops and the robber at all times. One of the main open questions is Meyniel’s Conjecture which states that, on any n-vertex graph, $O(\sqrt{n})$ cops can catch the robber. See [2] for a general account and [23, 41, 51] for the best current bounds.

Haslegrave [35] proposed a different game, Cat and Mouse, in which the cat wins by finding the mouse in finite time. As in Cops and Robbers, the players take alternate turns. However, in this game, the cat has no information about where the mouse is and, on its turn, it is not constrained by the edges of G and may test any vertex to see if the mouse is there. The mouse knows the cat's strategy in advance. The mouse is also required to move to a neighbouring vertex on each turn. These differences mean that the strategies employed by the players in Cat and Mouse differ greatly from those used in Cops and Robbers.

In Chapter 7, we will consider a variation of Cat and Mouse introduced by Dayanikli and Rautenbach [13]. The rules are as described above, except the mouse is not required to move on its turn. A time step consists of one turn for each player with the mouse playing first. After the mouse has moved, the cat chooses any vertex and is told whether the distance between itself and the mouse has increased compared with the previous time step. The cat wins if it is able to find the mouse up to a given distance d within finite time; otherwise, the mouse wins.

Clearly, if d is the diameter of the graph, then the cat can always win. Let the graph be any graph on n vertices. We will prove that the cat can always locate the mouse within finite time if $d = \sqrt{32n}$ and that the bound of $\sqrt{n}$ is tight up to a constant factor, disproving a conjecture of Dayanikli and Rautenbach [13].

Chapter 2

Maximising H-Colourings of Graphs

2.1 Introduction

Throughout this chapter, G will be a simple, loopless graph and H will be a simple graph, possibly with loops. We will adopt the same convention for vertex degrees as Engbers [20]: for any vertex $v \in V(H)$, we define $d(v) = |\{w \in V(H) : vw \in E(H)\}|$. In particular, adding a loop to a vertex in H increases the degree by one. All logarithms will be in base e.

A graph is *regular* if all of its vertices have the same degree. Recall from Chapter 1 the definition of a *graph homomorphism* from G to H (also called an H*-colouring* of G); this is a map $\psi : V(G) \to V(H)$ such that $ij \in E(G) \Rightarrow \psi(i)\psi(j) \in E(H)$. The number of H-colourings of G is denoted by $\hom(G, H)$.

In Section 2.2, we consider the set $\mathcal{G}$ consisting of all *connected* graphs on n vertices with minimum degree at least δ. For this $\mathcal{G}$ and any non-regular graph H, Engbers [20] showed that, for any fixed $\delta \geq 2$ and n sufficiently large, $\hom(G, H)$ is maximised uniquely by $G = K_{\delta,n-\delta}$. In this chapter, we will extend this result by showing that it holds for all $\delta \geq 3$ and for all graphs H. This answers a question posed by Engbers [20]. In the case where $\delta = 2$ and H is any graph, Engbers [19] showed that the number of H-colourings is maximised by one of $K_{2,n-2}$, $\frac{n}{3}K_3$ or $\frac{n}{4}K_{2,2}$ (depending on the structure of H).

An H-colouring of G requires that each component of G is mapped to a component of H. As we are only considering connected graphs G, each H-colouring of G maps G to a single component of H. We therefore begin with the case when H is connected.

Theorem 2.1. *For every $\delta \geq 3$ and every connected graph H, there exists a constant $\kappa(\delta, H)$ such that the following holds: if $n \geq \kappa(\delta, H)$ and G is a connected graph on n vertices with minimum degree at least δ, then we have* $\hom(G, H) \leq \hom(K_{\delta,n-\delta}, H)$. *Further, if H is not a complete looped graph or a complete balanced bipartite graph, then we have equality if and only if $G = K_{\delta,n-\delta}$.*

Extending this result to all graphs H follows as an easy corollary. If H has h components $H_1, \ldots H_h$, then $\hom(G, H) = \hom(G, H_1) + \cdots + \hom(G, H_h)$ because G is a connected graph. For n sufficiently large, $G = K_{\delta,n-\delta}$ maximises $\hom(G, H_i)$ for each component H_i and so $G = K_{\delta,n-\delta}$ also maximises $\hom(G, H)$.

Corollary 2.2. *For every $\delta \geq 3$ and every graph H, there exists a constant $\kappa(\delta, H)$ such that the following holds: if $n \geq \kappa(\delta, H)$ and G is a connected graph on n vertices with minimum degree at least δ, then we have* $\hom(G, H) \leq \hom(K_{\delta,n-\delta}, H)$. *Further, if H has a component which is neither a complete looped graph nor a complete balanced bipartite graph, then we have equality if and only if $G = K_{\delta,n-\delta}$.*

We may identify a proper q-colouring of a graph G with a graph homomorphism from G to K_q. Therefore, counting the number of proper q-colourings of G corresponds to counting the number of proper graph homomorphisms from G to K_q. As K_q is a connected graph, the following corollary also follows immediately from Theorem 2.1. This answers another question posed by Engbers [20].

Corollary 2.3. *Fix $\delta \geq 3$ and $q > 2$. Then, for n sufficiently large, $K_{\delta,n-\delta}$ uniquely maximizes the number of proper q-colourings amongst all connected graphs on n vertices with minimum degree at least δ.*

A natural extension to Corollary 2.2 is to allow G to have more than one component. Here the picture is less complete.

If H is the graph consisting of a single edge with one of the vertices looped, then counting the number of H-colourings of a graph G is equivalent to counting the number of independent sets in G. Extending previous work on this topic, Cutler and Radcliffe [12] gave complete results for all values of n and δ. In particular, if $n \geq 2\delta$, then $K_{\delta,n-\delta}$ is the unique graph which maximises $\hom(G, H)$.

Engbers [19] considered the question for general H, but only when the order of G is sufficiently large. They asked which graph on n vertices with minimum degree δ maximises the number of H-colourings as the value of n increases.

For any graph H and $\delta = 1$ or $\delta = 2$, Engbers showed that $\hom(G, H)$ is maximised by one of $\frac{n}{\delta+1}K_{\delta+1}$, $\frac{n}{2\delta}K_{\delta,\delta}$ or $K_{\delta,n-\delta}$ (where the graph that maximises $\hom(G, H)$ depends on the structure of H). These results led them to make the following conjecture.

Conjecture 2.4 [19]**.** *Fix $\delta \geq 1$ and any graph H. Let G be a graph on n vertices with minimum degree at least δ. There exists a constant $c(\delta, H)$ such that, for $n \geq c(\delta, H)$, we have*

$$\hom(G, H) \leq \max\left\{\hom(K_{\delta+1}, H)^{\frac{n}{\delta+1}}, \hom(K_{\delta,\delta}, H)^{\frac{n}{2\delta}}, \hom(K_{\delta,n-\delta}, H)\right\}.$$

In Section 2.3, we will use similar ideas to Sernau to construct counterexamples to Conjecture 2.4 whenever $\delta \geq 3$.

On the other hand, we can show that Conjecture 2.4 does hold in certain circumstances. In Section 2.4, we will consider the case when the graph H is fixed and δ and n are sufficiently large. In particular, for each $k \in \mathbb{N}$, we consider the family $\mathcal{H}_k$

of all graphs with maximum degree k that do not contain the complete looped graph on k vertices or $K_{k,k}$ as a component. We will prove the following theorem.

Theorem 2.5. *Fix $k \in \mathbb{N}$. For every graph $H \in \mathcal{H}_k$ and every $\delta \geq \delta_0(H)$, the following holds: there exists a constant $n_0(\delta, H)$ such that, if $n \geq n_0(\delta, H)$ and G is a graph on n vertices with minimum degree at least δ, then* $\hom(G, H) \leq \hom(K_{\delta,n-\delta}, H)$. *Equality holds if and only if $G = K_{\delta,n-\delta}$.*

The graph $K_{\delta,n-\delta}$ need not maximise the number of H-colourings if H has maximum degree k and contains either the complete looped graph on k vertices or $K_{k,k}$ as a component (i.e. $H \notin \mathcal{H}_k$). This is discussed in more detail in Section 2.5.

2.2 Proof of Theorem 2.1

The following definition was introduced by Engbers [19]. We will use it in the proof of Theorem 2.1 as well as in Section 2.4.

Definition. For any graph H with maximum degree k and $\delta \geq 1$, we define $S(\delta, H)$ to be the collection of all δ-tuples $(v_1, \ldots, v_\delta)$ with $v_i \in V(H)$ such that $v_1, \ldots, v_\delta$ all have the same k neighbours. Note that some or all of $v_1, \ldots, v_\delta$ can be identical. We define $s(\delta, H) = |S(\delta, H)|$. As H has at least one vertex of degree k, we have $s(\delta, H) \geq 1$.

We will need the following theorem of Erdős and Pósa (see [17] for the statement and proof of this theorem).

Theorem 2.6. *There is a function $f : \mathbb{N} \to \mathbb{R}$ such that, given any $d \in \mathbb{N}$, every graph contains either d disjoint cycles or a set of at most $f(d)$ vertices meeting all its cycles.*

Throughout this chapter, P_r will be a path with r vertices and the length of a path will be the number of vertices in it. We will frequently use the following lemma of Engbers [19].

Lemma 2.7. *Suppose H is not the complete looped graph on k vertices or $K_{k,k}$. Then, for any vertices $i, j \in V(H)$ and any $r \geq 4$, there are at most $(k^2-1)k^{r-4}$ H-colourings of P_r that map the initial vertex of that path to i and the terminal vertex to j.*

We will also need the following simple observation.

Proposition 2.8. *Let G and H be graphs with G connected and $X \subseteq V(G)$. Suppose the vertices of X have already been mapped to vertices of H. The remaining vertices of G can be mapped into $V(H)$ in such a way that there are at most $\Delta(H)$ choices for each vertex of $V(G)\backslash X$.*

Proof. Because G is connected, there is a shortest path from each vertex of $V(G)\backslash X$ to X. We order the vertices of $V(G)\backslash X$ by increasing distance from X. Each vertex $v \in V(G)\backslash X$ either has a neighbour in X or a neighbour before it in the ordering. Therefore, when we come to colour v, one of its neighbours has already been coloured so there are at most $\Delta(H)$ choices for v. □

Proof of Theorem 2.1. Let $\delta \geq 3$ be fixed and let H be a connected graph with maximum degree $k \in \mathbb{N}$. We have $|V(H)| \geq k$. There are two special cases to look at before we consider a general H.

1. *H is the complete looped graph on k vertices.*

   If G is any graph on n vertices, we find that $\hom(G, H) = k^n$ because any vertex of G can be mapped to any vertex of H. Hence, as any graph on n vertices with minimum degree δ maximises the number of H-colourings, we have $\hom(G, H) \leq \hom(K_{\delta,n-\delta}, H)$ as required.

2. $H = K_{k,k}$.

   H is bipartite so $\hom(G,H) \neq 0$ if and only if G is bipartite. For any connected bipartite graph G on n vertices, $\hom(G,H) = 2k^n$. This means that any connected bipartite graph on n vertices with minimum degree δ maximises the number of H-colourings and hence $\hom(G,H) \leq \hom(K_{\delta,n-\delta},H)$ as required.

As the theorem is true in these two cases, we may assume that H is not the complete looped graph on k vertices or $K_{k,k}$. We may also assume that $k \geq 2$ as we have already dealt with the cases when H is a single looped vertex and when $H = K_{1,1}$. Hence we may apply Lemma 2.7 when required.

Let G be a graph on n vertices with minimum degree δ that has the maximum number of H-colourings. We know that H has at least one vertex v of degree k. When considering H-colourings of $K_{\delta,n-\delta}$, we can map the vertex class of size δ to v and the other vertex class to the neighbours of v. Hence, $\hom(K_{\delta,n-\delta},H) \geq k^{n-\delta}$. We can improve this bound by mapping the vertex class of size δ to any element of $S(\delta,H)$ and the other vertex class to the k common neighbours of this element. It follows that $\hom(K_{\delta,n-\delta},H) \geq s(\delta,H)k^{n-\delta}$.

We will proceed to determine the structure of G. The assumption that G has most H-colourings tells us that $\hom(G,H) \geq \hom(K_{\delta,n-\delta},H)$. We will show that, for n sufficiently large, we must have $G = K_{\delta,n-\delta}$.

Claim 2.9. *G has a bounded number of disjoint cycles.*

Proof. Suppose that G has d disjoint cycles. We colour G in the following way. Pick any vertex of G and map it to any vertex of H. Take a shortest path from the starting vertex to a vertex on one of the disjoint cycles. There are at most k ways to map each vertex on this path to vertices of H. We then consider the other vertices on the cycle (as the end vertex of the path has already been mapped to a vertex of H). Lemma 2.7 gives at most $(k^2-1)k^{t-3}$ ways to map these vertices to H, where t is the number of vertices in the cycle. We then repeat this process of finding a shortest path from the already mapped vertices to one of the disjoint cycles and mapping the vertices in the path and cycle to H. Once all of the vertices in disjoint cycles have been considered, any remaining vertices can be mapped greedily with at most k choices for each by Proposition 2.8. Therefore

$$\hom(G, H) \leq |V(H)|(k^2 - 1)^d k^{n-2d-1} < |V(H)|k^{n-1}e^{-\frac{d}{k^2}}.$$

This is strictly smaller than $k^{n-\delta}$ whenever $d > k^2 \log |V(H)| + k^2(\delta - 1)\log k$. As $\hom(G, H)$ is maximal, it follows that G has a bounded number of disjoint cycles. This bound only depends on H and δ. Hence we have proved the claim. $\diamondsuit$

Applying Theorem 2.6 to G, we find that there exists a constant $\alpha = \alpha(\delta, H)$ such that G can be made acyclic by removing at most α vertices. We can therefore partition the vertices of G into a set A of size at most α and a set F such that $G[F]$ is a forest.

We will show that we can make F into an independent set by moving at most a constant number of vertices from F to A. This constant depends only on δ and H and not on the number of vertices in G.

We say that a component of a graph is *non-trivial* if it contains at least one edge.

Claim 2.10. *The forest F has a bounded number of non-trivial components.*

Proof. Suppose F has f non-trivial components, $G_1, \ldots G_f$. Each G_i is a tree and so contains a maximal path P_i. As every vertex in G has degree at least $\delta \geq 3$, each end-vertex of P_i must have a neighbour in A. We colour G in the following way. First map A into H. There are at most $|V(H)|^{|A|}$ ways to do this. We then consider each G_i in turn. By Lemma 2.2, there are at most $(k^2 - 1)k^{|P_i|-2}$ ways to colour P_i and at most k ways to colour each of the other vertices of G_i. Finally, we consider the remaining vertices of G, each of which has at most k possible choices by Proposition

2.8. Hence

$$\hom(G,H) \leq |V(H)|^{|A|}(k^2-1)^f k^{n-|A|-2f} < |V(H)|^{\alpha} k^{n-\alpha} e^{-\frac{f}{k^2}}.$$

This is strictly less than $k^{n-\delta}$ whenever $f > k^2\alpha \log|V(H)| + k^2(\delta - \alpha)\log k$. The maximality of $\hom(G,H)$ means that there exists a constant depending only on δ and H that bounds the number of non-trivial components of F and hence proves the claim. $\diamondsuit$

Let T be any non-trivial component of F. Define T' to be the subtree obtained from T by deleting all of the leaves. We will show that the size of T' is bounded by a constant that only depends on δ and H. This is done in two steps: first we show that the maximal length of a path in T is bounded and then we show that T' can only have a bounded number of leaves. Together, these two claims bound the size of T'.

Claim 2.11. *The length of the longest path in T is bounded.*

Proof. Suppose the longest path P in T is $u_1v_1u_2v_2\dots$ and has length b. We may write $b = 2b' + r$ where $r \in \{0,1\}$. The minimum degree of G is at least $\delta \geq 3$ and T is acyclic. Therefore, each vertex of P has a neighbour which is not on P. Further, every leaf of T must have a neighbour in A.

We colour the vertices of G as follows. First, colour A. Next, we colour the vertices of P using the following algorithm. Initially, $i = 1$. The algorithm colours vertices u_i and v_i at step i (and possibly some other vertices of T that do not lie on P).

At the i^{th} step, consider vertices u_i and v_i on P. If $i = 1$, u_i is an end-vertex of P and so has a neighbour in A; if $i \neq 1$, u_i has v_{i-1} as a neighbour. Hence, we know u_i is adjacent to a vertex which has already been coloured. Consider the vertex v_i. If v_i has a neighbour in A, we have a path of length 4 starting and ending at vertices which have already been coloured. Lemma 2.7 tells us there are at most $k^2 - 1$ choices for

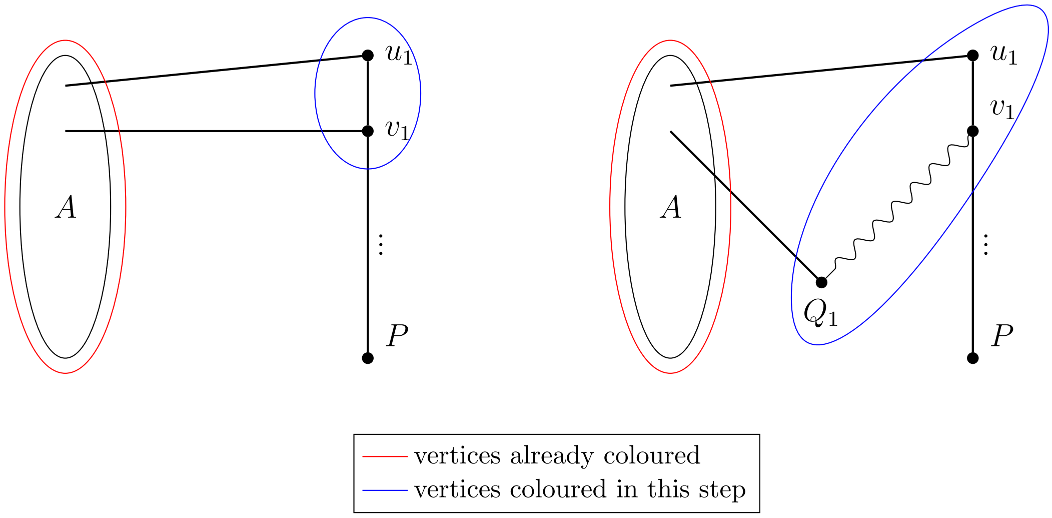


Figure 2.1: On the left, v_1 has a neighbour in A; on the right, v_1 does not.

u_i and v_i (see Figure 2.1). If v_i does not have a neighbour in A, it must have another neighbour in T which does not lie on P. Take a maximal path Q_i in T, which starts at v_i and avoids P. The end-vertex of Q_i that is not v_i must be a leaf in T and hence has a neighbour in A (see Figure 2.1). We therefore have a path of length $|Q_i| + 3$ which starts and ends with vertices that have already been coloured and has $u_i \cup Q_i$ as the internal vertices. Lemma 2.2 gives at most $(k^2 - 1)k^{|Q_i|-1}$ ways to colour the path $u_i \cup Q_i$. We then proceed to the $(i + 1)^{\text{th}}$ step of the algorithm.

After b' steps, we have coloured $2b'$ vertices of P (and possibly some other vertices of T). We finish by colouring all of the remaining vertices of G, each of which has at most k choices by Proposition 2.8. Therefore

$$\hom(G, H) \leq |V(H)|^{|A|}(k^2 - 1)^{b'} k^{n-|A|-2b'} < |V(H)|^{\alpha} k^{n-\alpha} e^{-\frac{b'}{k^2}}.$$

This is strictly less than $k^{n-\delta}$ whenever $b' > k^2\alpha \log |V(H)| + k^2(\delta - \alpha) \log k$. Because $\hom(G, H)$ is maximal, there exists a constant depending only on δ and H which bounds the length of a maximal path in any non-trivial component of F as required. $\diamondsuit$

Claim 2.12. *T' has a bounded number of leaves.*

Proof. Suppose T' has ℓ leaves. Each leaf of T' has at least two neighbours which are not in T' because the minimum degree of G is at least $\delta \geq 3$. At least one of these neighbours is a leaf of T. Similarly, every leaf of T has a neighbour in A.

We colour G by first colouring the vertices of A. For each leaf v of T', there are two possibilities. If v has two neighbours u and w which are leaves of T, there is a path of length 5 with end vertices in A and internal vertices u, v and w. By Lemma 2.7 there are at most $(k^2-1)k$ ways to colour the path uvw. If v only has one neighbour u which is a leaf of T, then v must also have a neighbour in A because it has at least δ neighbours and only one of these can be in T' (see Figure 2.2). Apply Lemma 2.7 to the path with end vertices in A and internal vertices u and v. There are at most k^2-1 choices for the colours of u and v.

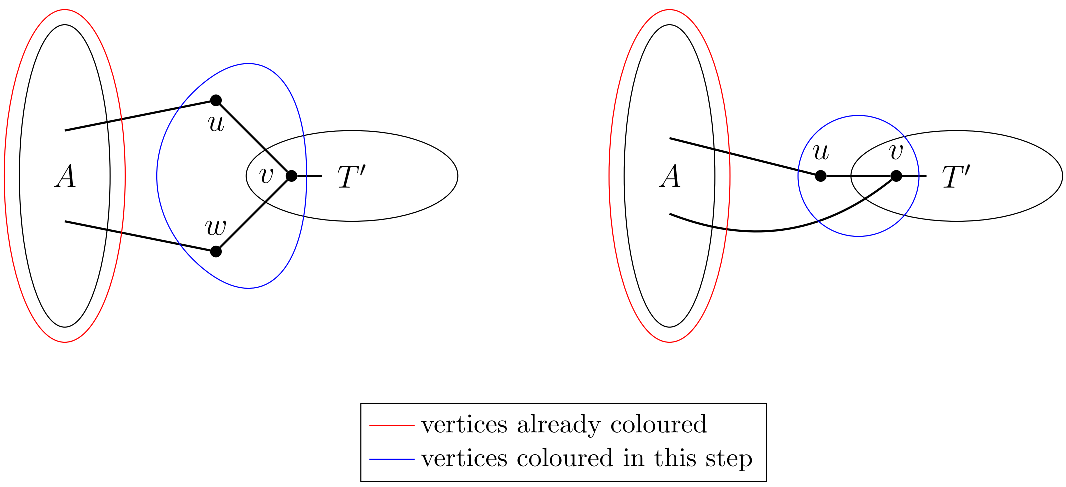


Figure 2.2: On the left, v has two leaves as neighbours; on the right, v has one.

Once each leaf of T' has been assigned to a vertex of H, there are at most k choices for each of the remaining vertices of G by Proposition 2.8. Therefore

$$\hom(G,H) \leq |V(H)|^{|A|}(k^2-1)^{\ell}k^{n-|A|-2\ell} < |V(H)|^{\alpha}k^{n-\alpha}e^{-\frac{l}{k^2}}.$$

This is strictly less than $k^{n-\delta}$ whenever $\ell > k^2\alpha \log|V(H)| + k^2(\delta-\alpha)\log k$. The maximality of $\hom(G,H)$ means that the maximum number of leaves T' can have is bounded above by a constant depending only on δ and H as required. $\Diamond$

Claims 2.11 and 2.12 show that, for each non-trivial component T of F, the subtree T' consisting of T without its leaves has maximal size bounded by a constant $t(\delta, H)$. Claim 2.10 shows that there are at most $f(\delta, H)$ non-trivial components of F for some constant $f(\delta, H)$.

We can make F into an independent set by moving some (possibly all) of the vertices of each T' from F to A. If any non-trivial component has $T' = \emptyset$, then T is a single edge and in this case we just move one of the end vertices from F to A. Hence, by moving at most $f(\delta, H)t(\delta, H)$ vertices from F to A, we can turn the forest into an independent set.

We have now partitioned the vertices of G into two sets of vertices L and R where $|L| \leq \alpha(\delta, H) + f(\delta, H)t(\delta, H)$ and R is an independent set. The size of L is bounded above by a constant that only depends on δ and H; it does not depend on the size of G.

Each vertex in R has at least δ neighbours in L because of the minimum degree of the vertices in G. By the pigeonhole principle, there exists a set $Y \subseteq L$ of size δ such that Y is contained in the neighbourhood of at least $(n - |L|)/\binom{|L|}{\delta} \geq cn$ vertices of R for some constant $c = c(\delta, H)$. Hence, G contains the subgraph $K_{\delta,cn}$.

If G does not contain $K_{\delta,n-\delta}$ as a subgraph, then Y is not a dominating set for G. Therefore, the subgraph induced by $G \backslash Y$ has a non-trivial component. If $G \backslash Y$ contains a non-trivial tree, take a maximal path X in this tree. Otherwise, choose X to be a cycle together with a shortest path from the cycle to Y.

We may colour the vertices of G in such a way that Y is always coloured first. Recall the definition of $S(\delta, H)$ given at the beginning of Section 2.2.

If Y is coloured using a δ-tuple from $S(\delta, H)$, we then colour the vertices of X. There are at most $(k^2 - 1)k^{|X|-2}$ ways do this. Finally, we colour the remaining vertices, each of which has at most k choices by Proposition 2.8. This gives at most $s(\delta, H)(k^2-1)k^{n-\delta-2}$ such colourings.

Alternatively, if Y is not coloured using a δ-tuple from $S(\delta, H)$, then there are at most $k-1$ ways to map each of the other cn vertices of the $K_{\delta,cn}$ subgraph into H. There are then at most k choices for each of the remaining vertices of G by Proposition 2.8. There are at most $|V(H)|^\delta(k-1)^{cn}k^{n-\delta-cn}$ such colourings.

Combining the above gives

$$\begin{aligned}\hom(G,H) &\leq s(\delta,H)(k^2-1)k^{n-\delta-2} + |V(H)|^\delta(k-1)^{cn}k^{n-cn-\delta} \\ &= s(\delta,H)k^{n-\delta} - s(\delta,H)k^{n-\delta-2} + |V(H)|^\delta(k-1)^{cn}k^{n-cn-\delta} \\ &< s(\delta,H)k^{n-\delta}\end{aligned}$$

for sufficiently large values of n. As $\hom(K_{\delta,n-\delta}, H) \geq s(\delta,H)k^{n-\delta}$, it follows that $\hom(G,H) < \hom(K_{\delta,n-\delta}, H)$.

If G contains $K_{\delta,n-\delta}$ as a subgraph and $G \neq K_{\delta,n-\delta}$, then we know that G contains at least one extra edge between two vertices in the same partition class. Clearly, every mapping of G into H is also a mapping of $K_{\delta,n-\delta}$ into H. We will show below that the converse is not true.

If ij is an edge in H, then mapping the size δ partition class of $K_{\delta,n-\delta}$ to i and the other partition class to j is a proper mapping of $K_{\delta,n-\delta}$ into H. However, it is only a proper mapping of G to H if the partition class containing the extra edge is mapped to a looped vertex. Therefore, if H has a non-looped vertex, $\hom(G,H) < \hom(K_{\delta,n-\delta})$.

Suppose every vertex of H is looped. We assumed that H was connected and not the complete looped graph so there will be non-adjacent vertices j and k which have a common neighbour i. We may map the partition class with the extra edge to vertices j and k and the other partition class to i. If the extra edge has one endpoint in j and the other in k, we do not get a proper H-colouring of G but it is a valid

H-colouring of $K_{\delta,n-\delta}$. Hence $\hom(G,H) < \hom(K_{\delta,n-\delta})$.

Therefore, if $\hom(G,H)$ is maximal and n is sufficiently large, then we must have $G = K_{\delta,n-\delta}$. $\square$

2.3 Counterexample to Conjecture 2.4

We write $T_t(x)$ for the *t-partite Turán graph on x vertices* (i.e. the complete t-partite graph on x vertices with the vertex classes as equal as possible).

For every $\delta \geq 3$, we will construct a graph H such that, for infinitely many values of n, the number of H-colourings is uniquely maximised by a disjoint union of complete multipartite graphs. This shows that Conjecture 2.4 does not hold.

For simplicity, we first assume that $(t-1)|\delta$ for some $3 \leq t \leq \delta$.

Theorem 2.13. *Fix $\delta \geq 3$ and $3 \leq t \leq \delta$ such that $\delta = (t-1)\alpha$ for some $\alpha \in \mathbb{N}$. Then there exists a constant $k_0(\delta)$ such that the following holds for all values of $m \in \mathbb{N}$: if $k \geq k_0(\delta)$ and G is any graph on $n = mt\alpha$ vertices with minimum degree at least δ, then we have* $\hom(G, kK_t) \leq \hom(mT_t(t\alpha), kK_t)$ *with equality if and only if $G = mT_t(t\alpha)$.*

Proof. Fix $\delta \geq 3$ and $3 \leq t \leq \delta$ as above where $\delta = (t-1)\alpha$. Take k sufficiently large such that $(t!k)^{1/(t\alpha)} > tk^{1/(t\alpha+1)}$.

Clearly, $\hom(K_{t+1}, kK_t) = 0$ and so we only need to consider graphs which are K_{t+1}-free.

Any K_{t+1}-free graph with minimum degree at least δ has at least $t\alpha$ vertices. Turán's theorem tells us that $T_t(t\alpha)$ is the only such graph with exactly $t\alpha$ vertices. It is easy to see that $\hom(T_t(t\alpha), kK_t) = t!k$.

Let $m \in \mathbb{N}$ and G be any graph on $n = mt\alpha$ vertices with minimum degree at least δ. We may assume that G has a components $G_1, \ldots G_a$ with $|G_1| \geq \cdots \geq |G_a| \geq t\alpha$.

Then $\hom(G, kK_t) = \prod_{i=1}^{a} \hom(G_i, kK_t)$. If $|G_1| = t\alpha$, then $|G_i| = t\alpha$ for all i and hence $G = mT_t(t\alpha)$.

Suppose that $|G_1| > t\alpha$. We know that, if $|G_i| = t\alpha$, then $G_i = T_t(t\alpha)$ and $\hom(G_i, kKt) = t!k$. If $|G_i| > t\alpha$, then we may colour the vertices of G_i greedily to get $\hom(G_i, kK_t) \leq tk(t-1)^{|G_i|-1} < kt^{|G_i|}$. We chose k such that $(t!k)^{1/(t\alpha)} > tk^{1/(t\alpha+1)}$. Using this and the fact that $|G_i| \geq t\alpha + 1$, we have $\hom(G_i, kK_t) < (t!k)^{|G_i|/(t\alpha)}$. Combining these two observations, we get

$$\hom(G, kK_t) = \prod_{i=1}^{a} \hom(G_i, kK_t) < (t!k)^{n/(t\alpha)} = (t!k)^m = \hom(mT_t(t\alpha), kK_t).$$

Therefore, if G is any graph on $n = mt\alpha$ vertices with minimum degree at least δ, we have $\hom(G, kK_t) \leq \hom(mT_t(t\alpha), kK_t)$. We have equality if and only if $G = mT_t(t\alpha)$. $\square$

We may use the techniques above to show that, if $(t-1)|(\delta+1)$, then a similar result holds – there is a graph H such that the number of H-colourings is uniquely maximised by a union of complete t-partite graphs. Therefore, for every $\delta \geq 3$, by taking $t = 3$, we can produce a counterexample to Conjecture 2.4.

In all of the examples we have seen so far, the number of H-colourings has been maximised by the union of complete multipartite graphs. We will now give an example where this is not the case.

Take $\delta = 7$ and $t = 4$ and choose k as in Theorem 2.13. Let $H = kK_4$, $m \in \mathbb{N}$ and take G to be any graph on $n = 10m$ vertices with minimum degree at least 7. As before, we may assume that G is 4-colourable. If G has a component with at least 11 vertices, then we can show, in a similar way to Theorem 2.13, that $\hom(G, kK_4) < \hom(mT_4(10), kK_4)$. Any union of complete multipartite graphs except $mT_4(10)$ is either not 4-colourable or contains a component with at least 11 vertices. Therefore $mT_4(10)$ maximises the number of H-

colourings among unions of complete multipartite graphs. However, the number of H-colourings is not maximised by $mT_4(10)$. Let T' be the graph formed from $T_4(10)$ by removing a perfect matching between the two vertex classes of size 2. Then $\hom(mT', kK_4) = 2\hom(mT_4(10), kK_4)$.

2.4 Proof of Theorem 2.5

We will need the following simple observation.

Proposition 2.14. *Fix $d \in \mathbb{N}$. Let G be any graph with minimum degree at least $3d$. Then G has at least d disjoint cycles.*

Proof. If $d = 1$, the minimum degree of G is at least 3 and so G contains a cycle. If $d > 1$, take C to be a shortest cycle in G. Each vertex in G has at most 3 neighbours on C or else we would be able to find a shorter cycle. Removing the vertices in C reduces the minimum degree by at most 3. Therefore, by induction, we can find at least $d-1$ disjoint cycles in $G \backslash V(C)$. $\square$

Before proving Theorem 2.5, we will prove a couple of useful lemmas. Recall the definitions of $S(\delta, H)$ and $s(\delta, H)$ given at the start of Section 2.2.

Lemma 2.15. *Fix $\delta \geq 1$ and $k \geq 2$. Fix H to be any graph with maximum degree k. Then there exists a constant $\beta(\delta, H)$ such that, for $n \geq \beta(\delta, H)$, we have $\hom(K_{\delta,n-\delta}, H) \leq s(\delta, H)k^{n+1-\delta}$.*

Proof. The graph $K_{\delta,n-\delta}$ has two vertex classes. Denote the class of size δ by Z. When we are counting the number of H-colourings of $K_{\delta,n-\delta}$, we will colour vertices in Z first and then the remaining vertices may be coloured greedily. There are two possibilities: either Z is coloured so that all of the vertices used in H have the same k neighbours (i.e. we use a δ-tuple from $S(\delta, H)$) or the vertices in H used to colour Z have strictly fewer than k neighbours in common.

First, we consider the case where Z is coloured using a δ-tuple from $S(\delta, H)$. When we come to colour the vertices of $G \backslash Z$, there are exactly k choices for each one. Therefore, there are exactly $s(\delta, H)k^{n-\delta}$ such colourings.

Next, we consider the case where Z is coloured so that the vertices used do not have k common neighbours in H. This leaves at most $k-1$ ways to map the vertices of $G \backslash Z$ into H. Hence, there are at most $|V(H)|^{\delta}(k-1)^{n-\delta}$ such colourings.

Combining the above gives

$$\hom(K_{\delta,n-\delta}, H) \leq s(\delta, H)k^{n-\delta} + |V(H)|^{\delta}(k-1)^{n-\delta}.$$

Hence, for n sufficiently large, we have

$$\begin{aligned}\hom(K_{\delta,n-\delta}, H) &\leq s(\delta, H)k^{n-\delta} + k^{n-\delta} \\ &\leq s(\delta, H)k^{n+1-\delta}.\end{aligned}$$

This proves the required result. □

Lemma 2.16. *Fix H to be any graph with maximum degree $k \in \mathbb{N}$ that does not have the complete looped graph on k vertices or $K_{k,k}$ as a component. There exists a constant $\delta_0(H)$ such that, if $\delta \geq \delta_0(H)$ and G is a connected graph on n vertices with minimum degree δ, then* $\hom(G, H) < k^{n-1}$.

Proof. The minimum degree condition on G ensures that $n \geq \delta + 1$. The restrictions on H mean that $k \geq 2$.

Let H have h components $H_1, \ldots H_h$. As G is connected, any H-colouring of G maps G to a single component H_i and so $\hom(G, H) = \sum_{i=1}^{h} \hom(G, H_i)$. We therefore first count the number of H_i-colourings of G for each $i \in [h]$. There are three cases to consider.

Case 1. Let H_i be a complete looped graph on ℓ vertices where $\ell < k$. Then $\hom(G, H_i) = \ell^n \leq (k-1)^n$. This is strictly less than k^{n-h-1} whenever we have $n > \frac{(h+1)\log k}{\log k - \log(k-1)}$.

Case 2. Let $H_i = K_{\ell,\ell}$ where $\ell < k$. Then $\hom(G, H_i) = 2\ell^n \leq 2(k-1)^n$. This is strictly less than k^{n-h-1} whenever $n > \frac{\log 2 + (h+1)\log k}{\log k - \log(k-1)}$.

Case 3. Let H_i be any connected graph which is not the complete looped graph on ℓ vertices or $K_{\ell,\ell}$ for some $\ell \leq k$. Suppose G has d vertex disjoint cycles $C_1, \ldots C_d$. We colour G in the following way:

1. Pick any vertex of G and map it to any vertex of H_i.
2. Find a shortest path P from the already coloured vertices of G to an uncoloured vertex on one of the cycles C_j. There are at most k ways to map each vertex on this path to vertices of H_i.
3. The end vertex of P has already been mapped to a vertex of H_i so we consider the other vertices on the cycle C_j. Lemma 2.7 gives at most $(k^2-1)k^{|C_j|-3}$ ways to map these vertices to H_i.
4. If, for some $j' \in [d]$, the cycle $C_{j'}$ has not yet been coloured, go back to step 2.
5. Colour any remaining uncoloured vertices in a greedy fashion. By Proposition 2.8, there are at most k choices for each vertex.

By colouring G in this way, we find that

$$\hom(G, H_i) \leq |V(H_i)|(k^2-1)^d k^{n-2d-1} < |V(H_i)| k^{n-1} e^{-\frac{d}{k^2}}.$$

This is strictly less than k^{n-h-1} whenever $d > k^2 \log|V(H_i)| + k^2 h \log k$.

Choose $\delta \geq \max\left\{3k^2 \log|V(H)| + 3k^2 h \log k, \frac{\log 2 + (h+1)\log k}{\log k - \log(k-1)}\right\}$ and note that $n \geq \delta + 1$. If H_i is in either Case 1 or Case 2, then n is large enough that $\hom(G, H_i) < k^{n-h-1}$. If H_i is in Case 3, then, by Proposition 2.14, we have that the number of disjoint cycles in G is at least $k^2 \log|V(H)| + k^2 h \log k$ and hence $\hom(G, H_i) < k^{n-h-1}$. Then

$$\hom(G, H) = \sum_{i=1}^{h} \hom(G, H_i) < hk^{n-h-1} < k^{n-1}.$$

Hence, if H does not contain the complete looped graph on k vertices or $K_{k,k}$ as a component, we have $\hom(G, H) < k^{n-1}$ for δ sufficiently large as required. □

We are now ready to prove the main result.

Proof of Theorem 2.5. Let H be any graph with maximum degree k that does not have the complete looped graph on k vertices or $K_{k,k}$ as a component. This allows us to apply Lemma 2.16 as required.

Choose $\delta \geq \delta_0(H)$ where $\delta_0(H)$ is the constant found in Lemma 2.16. Set $\lambda(\delta, H) = \max\{\kappa(\delta, H), \beta(\delta, H)\}$ where $\kappa(\delta, H)$ is the constant found in Theorem 2.1 and $\beta(\delta, H)$ is the constant found in Lemma 2.15. Now, choose $n > (\delta - 1)(\lambda(\delta, H) - 1)$.

Let G be a graph on n vertices with minimum degree δ that has the maximum number of H-colourings. Then $\hom(G, H) \geq \hom(K_{\delta, n-\delta}, H) \geq s(\delta, H)k^{n-\delta} \geq k^{n-\delta}$.

Let G have t components $G_1, \ldots G_t$. An H-colouring of G comprises of separate H-colourings of each component G_i and therefore $\hom(G, H) = \prod_{i=1}^{t} \hom(G_i, H)$. As G has the most H-colourings among all graphs on n vertices with minimum degree δ, we must also have that G_i has the most H-colourings among all graphs on $|G_i|$ vertices with minimum degree δ for each $i \in [t]$.

Claim 2.17. *G has a bounded number of components.*

Proof. By Lemma 2.16, we have that $\hom(G_i, H) < k^{|G_i|-1}$ for each $i \in [t]$ so

$$\hom(G, H) = \prod_{i=1}^{t} \hom(G_i, H) < \prod_{i=1}^{t} k^{|G_i|-1} = k^{n-t}.$$

If $t \geq \delta$, then we have $\hom(G, H) < k^{n-\delta} \leq \hom(K_{\delta,n-\delta}, H)$ and this contradicts our assumption that G has the maximum number of H-colourings. $\Diamond$

Hence we know that G has at most $\delta - 1$ components. By the pigeonhole principle, there is a component of G with at least $\lambda(\delta, H)$ vertices. Without loss of generality, we may assume this component is G_1. By Theorem 2.1, we have that $G_1 = K_{\delta,|G_1|-\delta}$ and, applying Lemma 2.15, we find that $\hom(G_1, H) \leq s(\delta, H)k^{|G_1|+1-\delta}$.

Claim 2.18. *G has exactly one component.*

Proof. Suppose $t > 1$. We know $\hom(G_1, H) \leq s(\delta, H)k^{|G_1|+1-\delta}$. By Lemma 2.16, we have $\hom(G_2, H) < k^{|G_2|-1}$. Hence

$$\begin{aligned}
\hom(G_1 \cup G_2, H) &< s(\delta, H)k^{|G_1|+1-\delta}k^{|G_2|-1} \\
&= s(\delta, H)k^{|G_1|+|G_2|-\delta} \\
&\leq \hom(K_{\delta,|G_1|+|G_2|-\delta}, H).
\end{aligned}$$

Replacing $G_1 \cup G_2$ by $K_{\delta,|G_1|+|G_2|-\delta}$ increases the number of H-colourings of G, which contradicts our assumption that G has the maximum number of H-colourings. $\Diamond$

We have seen that G has exactly one component G_1 and that this component is $K_{\delta,|G_1|-\delta}$. In other words, if G has the maximum number of H-colourings, then $G = K_{\delta,n-\delta}$ as required. $\square$

2.5 Conclusion

We have shown that, given any graph H and any $\delta \geq 3$, for sufficiently large n, the graph $G = K_{\delta,n-\delta}$ maximises $\hom(G,H)$ among all connected graphs on n vertices with minimum degree δ. If H has a component which is neither a complete looped graph nor a complete balanced bipartite graph, then $K_{\delta,n-\delta}$ is the unique such maximising graph.

We have also considered the more general question which was asked by Engbers [20]: what happens if we consider all graphs on n vertices with minimum degree δ, rather than just those which are connected? We will look at the case where H is fixed and $\delta \geq \delta_0(H)$. By making δ sufficiently large in relation to $|H|$, we are able to identify the maximising graph for certain graphs H.

In what follows, we take G to be any graph on n vertices with minimum degree δ. We assume that G has t components $G_1, \ldots G_t$.

If H is fixed with maximum degree k and δ is sufficiently large, then the graph which maximises the number of H-colourings depends on the structure of H. Some of the different possible graphs which maximise $\hom(G,H)$ are given below.

1. *H is h disjoint copies of the complete looped graph on k vertices.*

   It is easy to see that $\hom(G,H) = \prod_{i=1}^{t} |V(H)| k^{|G_i|-1} = h^t k^n$. When $h = 1$, $\hom(G,H) = k^n$ for any graph G on n vertices and so every graph G maximises the number of H-colourings. When $h > 1$, $\hom(G,H)$ is maximised when G has as many components as possible. The minimum number of vertices in a component of G is $\delta + 1$ which occurs when the component is $K_{\delta+1}$. Writing $n = a(\delta+1) + b$ where $b \in \{0, \ldots \delta\}$, we have that $\hom(G,H)$ is maximised by any graph with a components, e.g. $(a-1)K_{\delta+1} \cup K_{\delta+b+1}$.

2. *H is h disjoint copies of $K_{k,k}$.*

   It is easy to see that, if a graph is not bipartite, it is not possible to map it into

H. Therefore

$$\hom(G,H) = \begin{cases} \prod_{i=1}^{t} \hom(G_i,H) = (2h)^t k^n & \text{if } G \text{ is bipartite} \\ 0 & \text{if } G \text{ is not bipartite.} \end{cases}$$

Clearly, the number of H-colourings is maximised when G is bipartite and has as many components as possible. The smallest possible bipartite component of G is $K_{\delta,\delta}$ which has 2δ vertices. Writing $n = 2a\delta + b$ where $b \in \{0, \dots 2\delta - 1\}$, we have that $\hom(G,H)$ is maximised by any bipartite graph with a components, e.g. $(a-1)K_{\delta,\delta} \cup K_{\delta,\delta+b}$.

3. *No component of H is the complete looped graph on k vertices or $K_{k,k}$.*
   In Section 2.4, we showed that, for any $\delta \geq \delta_0(H)$, there exists a constant $n_0(\delta,H)$ such that, if $n \geq n_0(\delta,H)$, then $K_{\delta,n-\delta}$ uniquely maximises the number of H-colourings.

From the examples given above, it is clear to see that there is not a simple answer to the question of which graph G maximises $\hom(G,H)$ when H is fixed and δ is sufficiently large. We make the following conjecture.

Conjecture 2.19. *For any graph H and any $\delta \geq \delta_0(H)$, there exists a constant $n_0(\delta,H)$ such that the following holds: if G is a graph with minimum degree δ and at least $n_0(\delta,H)$ vertices, then*

$$\hom(G,H) \leq \max\left\{\hom(K_{\delta+1},H)^{\frac{|G|}{\delta+1}}, \hom(K_{\delta,\delta},H)^{\frac{|G|}{2\delta}}, \hom(K_{\delta,|G|-\delta},H)\right\}.$$

This conjecture implies that, for a fixed graph H and δ sufficiently large, the following holds: for sufficiently large n satisfying suitable divisibility conditions, the number of H-colourings is always maximised by one of $\frac{n}{\delta+1}K_{\delta+1}$, $\frac{n}{2\delta}K_{\delta,\delta}$ or $K_{\delta,n-\delta}$.

Chapter 3

Colouring the Complete Infinite Graph with 3 Colours

3.1 Introduction

For any $n \in \mathbb{N}$, let K_n be the complete graph on n vertices. Let $K_{\mathbb{N}}$ be the complete graph on the positive integers. In this chapter, we will write a *k-colouring* to mean a k-edge-colouring (i.e. a partition of the edges into k parts).

As noted in Chapter 1, in any 2-colouring of K_n or $K_{\mathbb{N}}$, there is a monochromatic spanning tree. Erdős, Gyárfás and Pyber and Tuza [22, 54] proved that, for any $n \in \mathbb{N}$ and any k-colouring of K_n, there exists a partition of the vertices into at most $k-1$ connected monochromatic subgraphs if $k \in \{3, 4, 5\}$. Somewhat surprisingly, nothing is known for $k \geq 6$.

In the infinite case, Hajnal, Komjáth, Soukup and Szalkai [34] showed that, for any k-colouring of $K_{\mathbb{N}}$, there is a partition of the vertices into at most k monochromatic connected subgraphs. It is believed that Conjecture 1.1 also holds for infinite graphs.

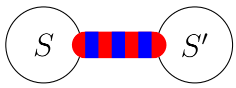
Conjecture 3.1. *Fix $k \in \mathbb{N}$. For any k-colouring of $K_{\mathbb{N}}$, there exists a partition of $\mathbb{N}$ into at most $k-1$ connected monochromatic subgraphs.*

As noted above, this is true for $k = 2$. Erdős, Gyárfás and Pyber [22] mention in their paper that Nagy and Szentmiklóssy proved it for $k = 3$, but, to our knowledge, a proof has never appeared in print. In Section 3.2, we will show that Conjecture 3.1 is indeed true for $k = 3$ by proving the following theorem.

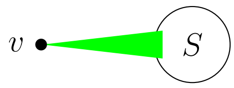
Theorem 3.2. *For every* 3*-colouring of the edges of* $K_{\mathbb{N}}$*, there exists a partition of* $\mathbb{N}$ *into at most 2 monochromatic connected subgraphs.*

Remark. In this chapter, there are a number of figures which depict the colours of edges between different sets of vertices. We will use the following conventions where S and S' are arbitrary sets:

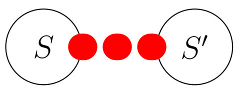
- A line of colour c between S and S' indicates that all edges between S and S' are colour c. If the line consists of two colours c and c', then all edges are either colour c or c'.



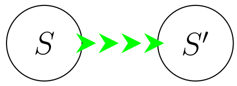
- A filled triangle of colour c from a vertex v to S indicates that all edges from v to S are colour c. As above, if the triangle is two colours, then edges may be of either colour.



- A dotted line of colour c between S and S' indicates that there is a path of colour c between any vertex in S and any vertex in S'.



- Arrows of colour c from S to S' indicates that every vertex in S has an edge of colour c to some vertex in S'.



Given a subgraph G, we will slightly abuse notation and use G to refer to both the subgraph and its set of vertices. We hope that this will not cause any confusion.

3.2 Proof of Theorem 3.2

Throughout this section, a *maximal* subgraph is defined to be a subgraph which is not properly contained in a monochromatic connected subgraph of any colour. A subgraph is *cofinite* if it covers all but a finite number of vertices.

In the first preprint of their paper [15], DeBiasio and McKenney proved Theorem 3.3 as a stepping stone to another result. However, they subsequently found a more direct proof of that result and so Theorem 3.3 was not included in the published version of their paper [16].

Theorem 3.3. *For every* 3*-colouring of the edges of* $K_{\mathbb{N}}$*, one of the following holds:*

1. *there exists a partition of* $\mathbb{N}$ *into at most two monochromatic connected subgraphs.*

2. *every maximal monochromatic connected subgraph is cofinite and there exists a cover of* $\mathbb{N}$ *by two monochromatic connected subgraphs* T_1 *and* T_2 *so that* $V(T_1) \cap V(T_2)$ *is finite.*

This theorem goes some way towards proving Theorem 3.2. In Lemma 3.4, we will adapt the proof of Theorem 3.3 to show that, if there is no partition, then the colouring must have a certain structure.

Lemma 3.4. *For every* 3*-colouring of the edges of* $K_{\mathbb{N}}$*, one of the following holds:*

1. *there exists a partition of* $\mathbb{N}$ *into at most two monochromatic connected subgraphs.*

2. *there are three maximal cofinite monochromatic subgraphs G, R and B with G green, R red and B blue and every vertex in $\mathbb{N}$ lies in at least two of G, R and B.*

Proof of Lemma 3.4. Let G be a maximal monochromatic connected subgraph of $K_{\mathbb{N}}$. Without loss of generality, G is green. Set $U = \mathbb{N} \setminus G$. If possible, we choose G so that U is infinite. If not, choose G so that U is finite of minimal size.

If $|U| \leq 2$, then it is easy to find a monochromatic subgraph covering U and hence we have found the required partition. Therefore, we may assume that $|U| \geq 3$. By the maximality of G, every edge from G to U is either red or blue. Consider the bipartite subgraph with vertex sets G and U and let R' be a connected monochromatic component in this subgraph with $R' \cap U$ as large as possible. Without loss of generality, R' is red. Because G is maximal, we know that $G \setminus R' \neq \emptyset$ (see Figure 3.1).

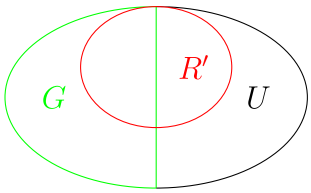


Figure 3.1: The subgraphs G and R'.

The following claim comes from the proof of Theorem 3.3. For completeness, we recap the argument here.

Claim 3.5. *There is a partition of $\mathbb{N}$ into two monochromatic connected subgraphs if $U \setminus R' \neq \emptyset$.*

Proof. Suppose that $U \setminus R' \neq \emptyset$. Then every edge from $U \setminus R'$ to $G \cap R'$ must be blue. Similarly, every edge from $U \cap R'$ to $G \setminus R'$ must be blue. This gives two blue connected subgraphs which are disjoint and cover $\mathbb{N}$ (see Figure 3.2). $\Diamond$

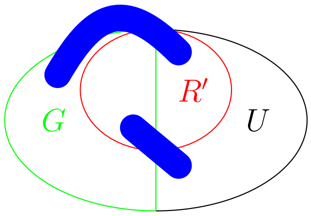


Figure 3.2: Partition of $\mathbb{N}$ if $U \setminus R' \neq \emptyset$.

Therefore we may assume that together G and R' cover $\mathbb{N}$. As noted in Claim 3.5, $G \setminus R'$ is complete to U in blue. We call this bipartite subgraph B'. We define

$$X = \{v \in G \cap R' : \forall u \in U \ uv \text{ is a red edge}\},$$
$$Y = (G \cap R') \backslash X.$$

We have the situation seen in Figure 3.3.

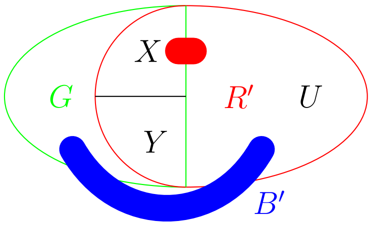


Figure 3.3: The subgraphs G, R' and B'.

If $X = \emptyset$, then, as every vertex in Y has a blue edge to U, we have a connected blue bipartite subgraph covering all of $\mathbb{N}$, which contradicts the maximality of G. Therefore, $X \neq \emptyset$. The following claim comes from the proof of Theorem 3.3. Again, we recap the argument here for completeness.

Claim 3.6. *There is a partition of* $\mathbb{N}$ *into two monochromatic connected subgraphs if* U *is infinite or* U *is finite with* $|U| > |Y|$.

Proof. Suppose that U is infinite or U is finite with $|U| > |Y|$. Then there exists a function $f : Y \to U$ which is injective but not surjective. We partition $Y = Y_b \cup Y_r$

where $Y_b = \{y \in Y : yf(y) \text{ is blue}\}$ and $Y_r = Y \setminus Y_b$. Without loss of generality, we may suppose that $Y_b \neq \emptyset$. Let $f(Y_b)$ be the image of Y_b under f so $f(Y_b) \neq \emptyset$. As f is not surjective, $U \setminus f(Y_b) \neq \emptyset$. It follows that $\mathbb{N}$ is partitioned by the blue subgraph $(G \setminus R') \cup Y_b \cup f(Y_b)$ and the red subgraph $Y_r \cup (U \setminus f(Y_b)) \cup X$ (see Figure 3.4). $\Diamond$

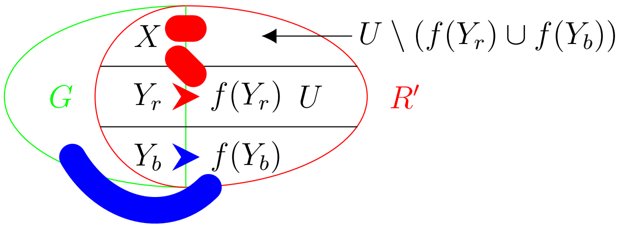


Figure 3.4: Partition of $\mathbb{N}$ if U is infinite or U is finite with $|U| > |Y|$.

Therefore, we may assume that U is finite and that $|Y| \geq |U| \geq 3$. Therefore, G is cofinite.

Let R be the largest connected red subgraph containing R'. As G is maximal, R cannot be contained within a connected green subgraph. If there was a blue connected subgraph containing R, then combining this blue subgraph with B' would give a blue connected subgraph spanning $\mathbb{N}$. Therefore, we may assume that R is a maximal monochromatic connected subgraph.

Similarly, let B be the largest connected blue subgraph containing B'. As both R and G are maximal, B cannot be contained with either a red or a green connected subgraph and hence we may assume that B is also maximal.

At the beginning of this proof, we took G to be an arbitrary maximal monochromatic connected subgraph of $K_{\mathbb{N}}$. By applying the same argument to R and then to B, we either find a partition of $\mathbb{N}$ into two monochromatic connected subgraphs or both R and B must be cofinite.

In the latter case, by construction, every vertex of $\mathbb{N}$ is in exactly two of G, R' and B'. Hence, we have that every vertex of $\mathbb{N}$ is in at least two of the subgraphs G, R and B. $\square$

We are now ready to prove Theorem 3.2.

Proof of Theorem 3.2. By Lemma 3.4, it suffices to consider the situation where there are 3 maximal cofinite monochromatic subgraphs G, R and B with G green, R red and B blue and every vertex in $\mathbb{N}$ lies in at least two of G, R and B. We define the complements of G, B and R to be $U = \mathbb{N} \setminus G$, $V = \mathbb{N} \setminus B$ and $W = \mathbb{N} \setminus R$. Let $Z = G \cap R \cap B$ be the set of vertices lying in G, R and B.

The sets U, V and W must all be finite because G, B and R are all cofinite. As noted in Lemma 3.4, we may assume that $|U|, |V|, |W| \geq 3$.

All of the vertices in Z have edges of all three colours because they are contained in G, R and B. As U is disjoint from G, vertices in U only send blue and red edges to Z. Similarly, vertices in V only send red and green edges to Z and those in W only send green and blue edges to Z.

Vertices in U only have red and blue edges to vertices not in U and vertices in V only have green and red edges to those not in V. Therefore, all edges between U and V must be red. Similarly, all edges between V and W are green and all edges between U and W are blue (see Figure 3.5).

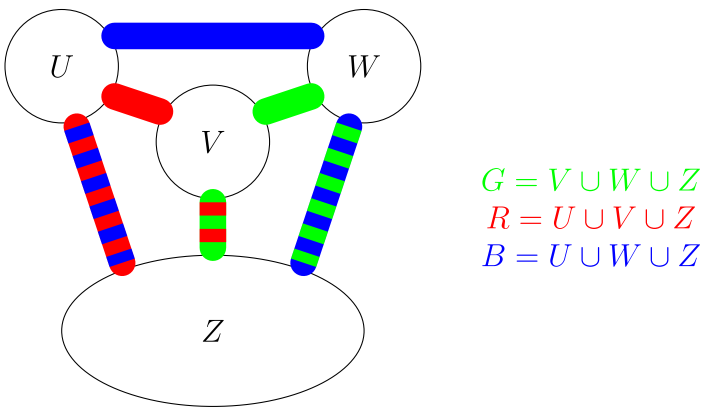


Figure 3.5: The subgraphs G, R and B.

We may also assume U, V and W contain edges of all three colours. If, for example, W only contained edges of two colours, we would be able to find a monochromatic connected subgraph which spans W and so we would have a partition of $\mathbb{N}$.

Fix $a \in V$ and let A be the set of vertices in Z which become disconnected from R if a is removed. Suppose that $R \setminus \{a\}$ is still connected so $A = \emptyset$. Then, the red subgraph $R \setminus \{a\}$ and the green subgraph $W \cup \{a\}$ are disjoint monochromatic connected subgraphs which cover all of $\mathbb{N}$ (see Figure 3.6).

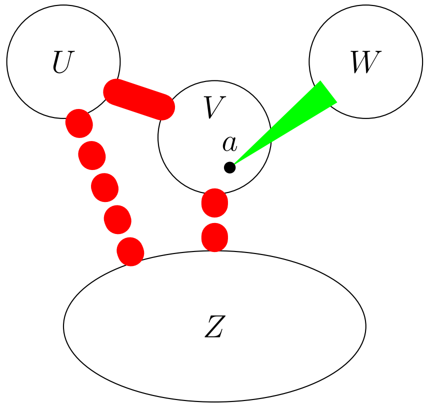


Figure 3.6: Partition of $\mathbb{N}$ if $A = \emptyset$.

Therefore, we may assume that R will become disconnected if any vertex from V is removed. We may partition A into two sets A_r and A_g where $A_r = \{v \in A : va \text{ is red}\}$ and $A_g = \{v \in A : va \text{ is green}\}$. If every vertex in A_r is joined to a vertex in $W \cup A_g$ by a green edge, then the green subgraph $W \cup \{a\} \cup A$ and the red subgraph $R \setminus (\{a\} \cup A)$ partition $\mathbb{N}$ (see Figure 3.7).

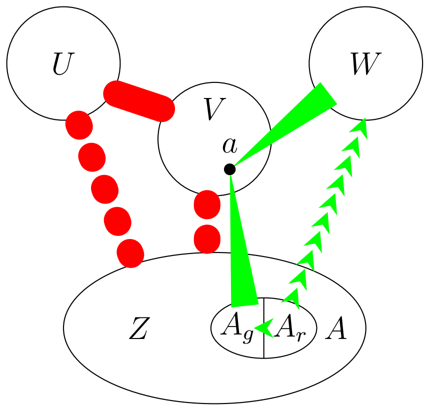


Figure 3.7: Partition of $\mathbb{N}$ if every vertex in A_r has a green edge to $W \cup A_g$.

Therefore, there must be some vertex in A_r which only sends blue and red edges to A_g and is complete to W in blue. Now fix any vertex $c \in U$ and consider the set

$C \subset Z$ of vertices which become disconnected from R by the removal of c. By an identical argument, we may assume that $C \neq \emptyset$ and that there exists a vertex in C which is adjacent to c by a red edge, only sends green edges to W and sends red and green edges to vertices in C which are adjacent to c by a blue edge.

We will consider the colours of edges between A and C. Note that, as vertices of A are disconnected from R by the removal of a, all edges between A and U must be blue. Similarly, all edges from C to V are green (see Figure 3.8).

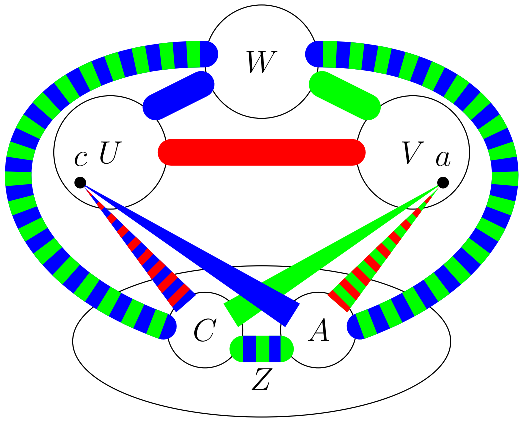


Figure 3.8: The edges adjacent to A and C.

Consider the edges between A and C. There cannot be any red edges between these two sets because A becomes disconnected from R when a is deleted. First suppose that every vertex in A is joined to a vertex in C by a green edge. Then $\mathbb{N}$ is partitioned by the green subgraph $W \cup \{a\} \cup A \cup C$ and the red subgraph $R \setminus (\{a\} \cup A \cup C)$ (see Figure 3.9).

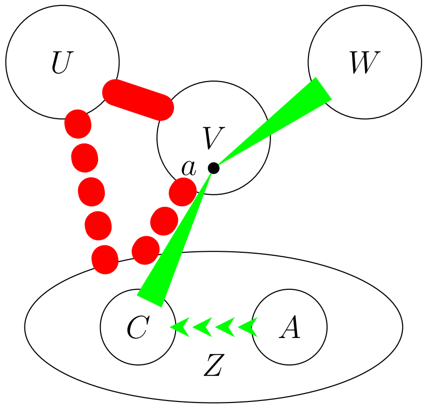


Figure 3.9: Partition if every vertex in A has a green edge to C.

Now suppose that there is a vertex $b \in A$ which only sends blue edges to C. Then $\mathbb{N}$ is partitioned by the blue subgraph $W \cup \{c\} \cup C \cup A$ and the red subgraph $R \setminus (\{c\} \cup A \cup C)$ (see Figure 3.10).

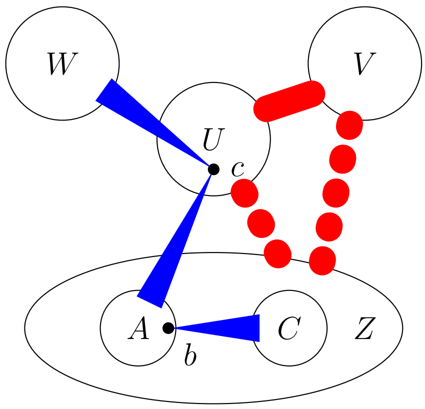


Figure 3.10: Partition if there exists $b \in A$ which only sends blue edges to C.

In every case, we have managed to find a partition of $\mathbb{N}$ into at most two monochromatic connected subgraphs. $\square$

Chapter 4

Monochromatic Components in Edge-Coloured Graphs with Large Minimum Degree

4.1 Introduction

We noted in Chapter 1 that any 2-colouring of the edges of K_n contains a monochromatic spanning subgraph and that Gyárfás [31] extended this result to $k \geq 3$ colours by proving the following theorem.

Theorem 4.1. *Fix $k \geq 2$. In every k-edge-colouring of the complete graph on n vertices, there exists a monochromatic component of order at least $\frac{n}{k-1}$.*

The bound in this theorem is sharp if $k-1$ is a prime power and n is divisible by $(k-1)^2$. Consider the affine plane of order $k-1$ and colour the edges in the i^{th} parallel class with colour i for each $i \in [k]$. Every monochromatic component contains exactly $\frac{n}{k-1}$ vertices.

For $k = 2$, it is easy to see that the conclusion of Theorem 4.1 does not hold for 2-colourings of the edges of a non-complete graph: if xy is not an edge, then colour edges red if they are incident with x and blue otherwise. However, Gyárfás and Sárközy [33] proved the following theorem.

Theorem 4.2. *Let G be a graph of order n with $\delta(G) \geq \frac{3}{4}n$. If the edges of G are 2-coloured, then there exists a monochromatic component of order at least $\delta(G)+1$.*

Gyárfás and Sárközy also showed that the bounds in this theorem are best possible in the following two senses.

Firstly, we cannot reduce $\delta(G)$ below $\frac{3}{4}n$. Suppose that n is divisible by 4 and partition the vertices into 4 sets V_1, V_2, V_3 and V_4, each of order $\frac{n}{4}$. Colour the following edges red: the edges within each V_i, those between V_1 and V_2 and those between V_3 and V_4. Colour the edges between V_1 and V_3 blue and the edges between V_2 and V_4 blue. All monochromatic components have order $\frac{n}{2}$ and $\delta(G) = \frac{3}{4}n - 1$.

Secondly, the largest monochromatic component that we can guarantee has order $\delta(G)+1$. Take the complete graph K_n and let $X, Y \subset V(K_n)$ be disjoint vertex sets with $|X| = |Y| = z(\delta)$ where $0 \leq z \leq \frac{n}{2}$. Form the graph G by removing all edges between X and Y. Colour the edges incident to X red, the edges incident to Y blue and all other edges arbitrarily. The largest monochromatic component in G in either colour has order $n - z = \delta(G) + 1$.

For $k \geq 3$, the situation is different. In this case, it is possible to remove some edges from a complete graph and still obtain a monochromatic component of order $\frac{n}{k-1}$ in every k-edge-colouring. Indeed, Gyárfás and Sárközy [32] showed that the complete graph can be replaced by any graph G with $\delta(G) \geq (1-\varepsilon_k)n$ for some constant $\varepsilon_k > 0$. They made the following conjecture.

Conjecture 4.3. *Fix $k \geq 3$. Let G be any graph with n vertices and minimum degree $\delta(G) \geq \left(1 - \frac{k-1}{k^2}\right)n$. If the edges of G are k-coloured, then there exists a monochromatic component of order at least $\frac{n}{k-1}$.*

Recently, there has been some progress towards Conjecture 4.3. The best current general result was proved by DeBiasio, Krueger and Sárközy [14].

Theorem 4.4. *Fix an integer $k \geq 3$ and let G be any graph of order n with minimum degree $\delta(G) \geq \left(1 - \frac{1}{3072(k-1)^5}\right) n$. If the edges of G are k-coloured, then there exists a monochromatic component of order at least $\frac{n}{k-1}$.*

For 3 colours, DeBiasio, Krueger and Sárközy [14] proved a stronger result.

Theorem 4.5. *Let G be a graph of order n with $\delta(G) \geq \frac{7}{8}n$. In every 3-colouring of the edges of G, there exists a monochromatic component of order at least $\frac{n}{2}$.*

This is not far from the constant $\frac{7}{9}$ predicted by Conjecture 4.3. However, in this chapter, we show that the conjecture is in fact false by proving the following theorem.

Theorem 4.6. *For every $n \in \mathbb{N}$, there exists a graph G of order n with minimum degree $\delta(G) \geq \left\lfloor \frac{5}{6}n \right\rfloor - 2$ and a 3-colouring of the edges of G such that every monochromatic component has order strictly less than $\frac{n}{2}$.*

We further show that $\frac{1}{6}$ is the largest possible value for ε_3 by proving the following theorem.

Theorem 4.7. *Let G be a graph of order n with $\delta(G) \geq \frac{5}{6}n$ where the edges of G have been 3-coloured. Then G has a monochromatic component with order at least $\frac{n}{2}$.*

The chapter is organised in the following way. In Section 4.2, we will prove Theorem 4.6. Then, in Sections 4.3 and 4.4, we will prove Theorem 4.7. First we will reduce the problem to two specific cases in Section 4.3. We will then explain how these two cases can be formulated as collections of linear programs and solved in Section 4.4. Together these results show that $\varepsilon_3 = \frac{1}{6}$.

4.2 Proof of Lower Bound

In this section, we give constructions to prove the lower bound given in Theorem 4.6. We will give a separate construction for each residue class modulo 6.

Proof of Theorem 4.6. First, take $n = 6q + 4$ for some $q \in \mathbb{N}$. We will consider other residue classes later.

We will construct a graph G of order n with $\delta(G) = 5q+2$. We will also show that there is a 3-colouring of the edges of G such that every monochromatic component has order strictly less than $\frac{n}{2} = 3q + 2$. The colours will be red, blue and green.

Partition the vertices into 8 sets $V_1, \dots, V_8$ with the following sizes:

- $|V_1| = |V_2| = 2$
- $|V_3| = q - 3$
- $|V_4| = q - 1$
- $|V_5| = |V_6| = |V_7| = |V_8| = q + 1$.

Observe that $|V_1| + \cdots + |V_8| = 6q + 4 = n$. There are no edges between:

- V_1 and V_2
- V_4 and $V_1 \cup V_2 \cup V_3$
- V_5 and V_8
- V_6 and V_7.

All other edges are present (including all edges inside vertex classes). This means that each vertex in V_3 has degree $5q + 4$ and every other vertex has degree $5q + 2$. Therefore $\delta(G) = 5q + 2 = \left\lfloor \frac{5}{6}n \right\rfloor - 1$ as required.

It remains to construct a 3-colouring of the edges in which every monochromatic component has order strictly less than $\frac{n}{2}$. We colour edges between vertex classes as shown in Figure 4.1; edges inside classes are coloured arbitrarily with the exception of the edge within V_1 which is red. Where a line in Figure 4.1 consists of two colours, edges between these two vertex sets may be of either colour.

In this colouring, the three red components have orders 2, $3q + 1$ and $3q + 1$, the three blue components have orders $q+3$, $2q$ and $3q+1$ and the three green components have orders $q + 3$, $2q$ and $3q + 1$. As $\frac{n}{2} = 3q + 2$, all monochromatic components have order strictly less than $\frac{n}{2}$ as required.

For the remaining residue classes modulo 6, we construct similar graphs with $\delta(G) = \left\lfloor \frac{5}{6}n \right\rfloor - c$ for some constant $c \in \{1, 2\}$ that depends on the residue class.

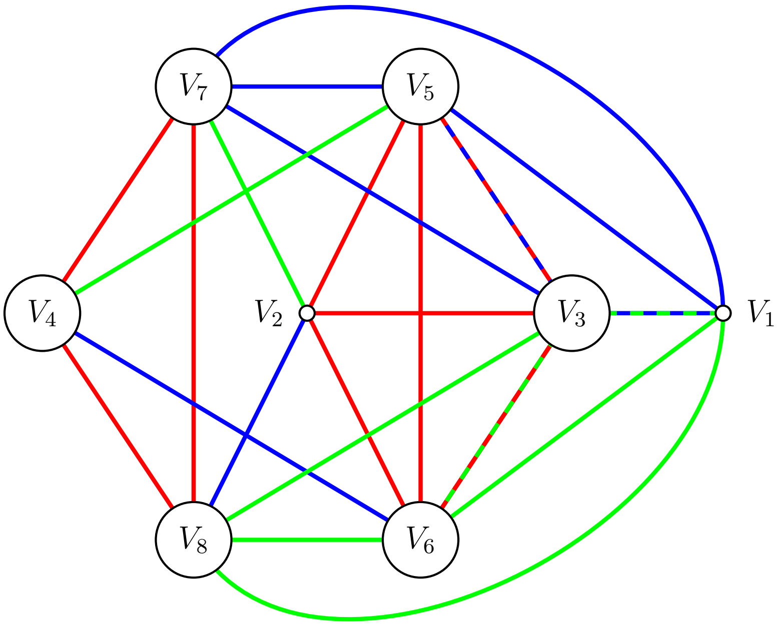


Figure 4.1: The graph G for $n = 6q + 4$.

Case: $n = 6q$. Partition the vertices into 6 sets with the following sizes:

- $|V_1| = |V_2| = |V_3| = q + 1$
- $|V_4| = |V_5| = |V_6| = q - 1$.

We colour edges between vertex classes as in Figure 4.2 and edges inside classes arbitrarily to get the graph G. The largest monochromatic component has order $3q - 1 = \frac{n}{2} - 1$ and $\delta(G) = 5q - 2 = \frac{5}{6}n - 2$.

Case: $n = 6q + 1$. Partition the vertices into 8 sets with the following sizes:

- $|V_1| = |V_2| = 1$
- $|V_3| = q - 3$
- $|V_4| = q - 2$
- $|V_5| = |V_6| = |V_7| = |V_8| = q + 1$.

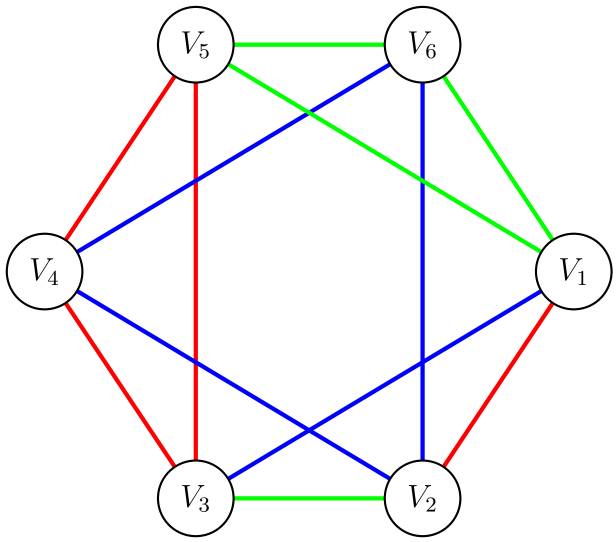


Figure 4.2: The graph G for $n = 6q$.

We colour edges between the vertex classes the same as in Figure 4.1 and edges inside classes arbitrarily to get the graph G. (Note that the number of vertices in each set is different from the case where $n \equiv 4 \mod 6$.) The largest monochromatic component has order $3q < \frac{n}{2}$ and $\delta(G) = 5q - 1 = \left\lfloor \frac{5}{6}n \right\rfloor - 1$.

Case: $n = 6q + 2$. Partition the vertices into 8 sets with the following sizes:

- $|V_1| = |V_2| = 2$
- $|V_3| = q - 4$
- $|V_4| = q - 2$
- $|V_5| = |V_6| = |V_7| = |V_8| = q + 1$.

We colour edges between the vertex classes the same as in Figure 4.1 and edges inside classes arbitrarily to get the graph G. We see that the largest monochromatic component has order $3q = \frac{n}{2} - 1$ and $\delta(G) = 5q = \left\lfloor \frac{5}{6}n \right\rfloor - 1$.

Case: $n = 6q + 3$. Partition the vertices into 6 sets with the following sizes:

- $|V_1| = |V_2| = |V_3| = q + 1$
- $|V_4| = |V_5| = |V_6| = q$.

We colour edges between the vertex classes as in Figure 4.2 and edges inside classes arbitrarily to get the graph G. We see that the largest monochromatic component has order $3q + 1 < \frac{n}{2}$ and $\delta(G) = 5q + 1 = \left\lfloor \frac{5}{6}n \right\rfloor - 1$.

Case: $n = 6q + 5$. Partition the vertices into 9 sets with the following sizes:

- $|V_1| = |V_2| = |V_3| = 1$
- $|V_4| = |V_5| = |V_6| = |V_7| = q$
- $|V_8| = |V_9| = q + 1$.

We colour edges between vertex classes as in Figure 4.3 and edges inside classes arbitrarily to get the graph G. The largest monochromatic component has order $3q + 2 < \frac{n}{2}$ and $\delta(G) = 5q + 2 = \left\lfloor \frac{5}{6}n \right\rfloor - 2$.

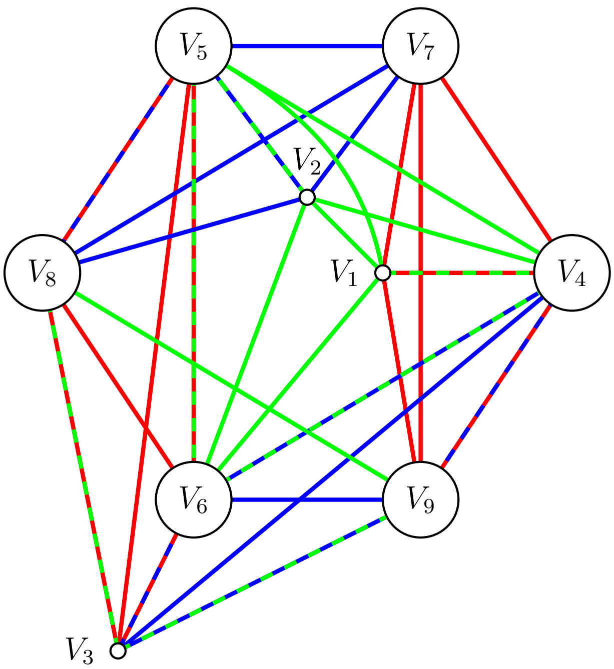


Figure 4.3: The graph G for $n \equiv 5 \mod 6$.

$\square$

It is worth noting that, as $n \to \infty$, the graph shown in Figure 4.1 (corresponding to the cases where $n \equiv 1, 2, 4 \mod 6$) is close to the graph in Figure 4.4, which is one of the optimal cases found in Section 4.4. For the other residue classes, we similarly find that they are close to other optimal cases found in Section 4.4 as $n \to \infty$.

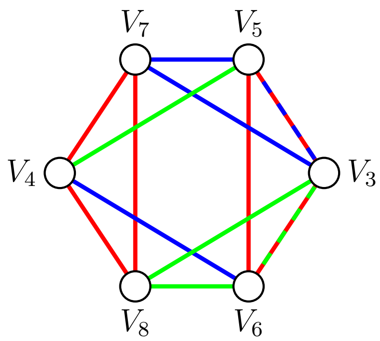


Figure 4.4: The "limit" graph of G for $n \equiv 1, 2, 4 \mod 6$.

The graphs given in the proof of Theorem 4.6 are by no means unique. For each residue class, we can find other graphs G' with $\delta(G') = \left\lfloor \frac{5}{6}n \right\rfloor - c$ for some small c that have no monochromatic component covering half of the vertices. Indeed, for each of the optimal graphs found in Section 4.4, it is possible to find some such G' which is close to it as $n \to \infty$.

4.3 Reducing the Upper Bound to Special Cases

To prove Theorem 4.7, we will reduce the problem to one which can be written as a series of linear programs. We solve these linear programs by computer to show that the theorem is true.

We begin by proving a series of lemmas. Throughout this section, we will assume that the three colours used to colour the edges are red, blue and green.

Lemma 4.8. *Let G be a graph of order n with $\delta(G) \geq \frac{5}{6}n$. Suppose that the edges of G are 3-coloured. If there exists a vertex v which is not incident to edges of all 3 colours, then there exists a monochromatic component of order at least $\frac{n}{2}$.*

Proof. Colour the edges of G red, blue and green. First consider the case where the vertex v is only incident to edges of one colour, say red. As $\delta(G) \geq \frac{5}{6}n$, the red component containing v covers at least $\frac{5}{6}n + 1 > \frac{n}{2}$ of the vertices of G.

Now consider the case where v is only incident to edges of two colours, say red and blue. Let $R \subseteq V(G)$ be the vertices of the red component containing v and $B \subseteq V(G)$ be the vertices of the blue component containing v. We may assume that $|R| < \frac{n}{2}$ and $|B| < \frac{n}{2}$.

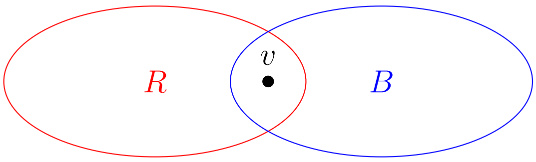


Figure 4.5: The vertices of the red and blue components containing v.

Let $x = |R \cap B|$, $r = |R \setminus B|$ and $b = |B \setminus R|$. Without loss of generality, we will assume that $r \geq b$. As $v \in R \cap B$ and $\delta(G) \geq \frac{5}{6}n$, we find that $x > 0$, $\frac{5}{12}n < r + x < \frac{n}{2}$ and $\frac{1}{3}n < b + x \leq r + x$.

Suppose that $x < \frac{1}{6}n$. As $r + x + b \geq \frac{5}{6}n$ and $r \geq b$, we must have $2r \geq \frac{5}{6}n - x$ giving $r > \frac{1}{3}n$. It follows from $b + x > \frac{1}{3}n$ that $b > \frac{1}{6}n$. Any edges that are incident to both $R \setminus B$ and $B \setminus R$ must be green. As $\delta(G) \geq \frac{5}{6}n$ and $r > \frac{1}{3}n$, every pair of vertices in $B \setminus R$ must have a green neighbour in common in $R \setminus B$. As $b > \frac{1}{6}n$, every vertex in $R \setminus B$ has a green neighbour in $B \setminus R$. Hence there is a green component covering all of $(R \cup B) \setminus (R \cap B)$. This green component has order at least $\frac{1}{6}n + \frac{1}{3}n = \frac{n}{2}$.

Now suppose that $x \geq \frac{1}{6}n$. As $r + x + b \geq \frac{5}{6}n$ and $r \geq b$, it follows that $r \geq \frac{5}{12}n - \frac{1}{2}x$ and hence $|R| = r + x \geq \frac{5}{12}n + \frac{1}{2}x \geq \frac{n}{2}$ giving a red component covering half of the vertices. $\square$

Given a graph G, the *t-blow-up* G' is the graph formed from G by replacing each vertex with a copy of K_t and each edge with a copy of $K_{t,t}$. If the edges of G have been coloured, then the edges of G' are coloured as follows:

- edges between two K_t are coloured according to the 3-edge-colouring on G
- edges within a K_t are coloured arbitrarily.

The graph G' behaves like a larger version of G.

Lemma 4.9. *Fix $t > 1$. Let G be a graph of order n. Suppose the edges of G are 3-coloured so that every vertex is incident to all three colours and there is no monochromatic component covering half of the vertices. Let G' be the t-blow-up of G. Then G' has no monochromatic component covering half of its vertices and further $\delta(G') = t\delta(G) + t - 1$.*

Proof. Let $v_1, \ldots, v_n$ be the vertices of G and let $V_1, \ldots, V_n$ be the corresponding copies of K_t in G'. Fix a colour, say red. Since every vertex of G is incident with a red edge, each set V_i is contained in some red component of G' and furthermore V_i and V_j lie in the same red component if and only if v_i and v_j do. The remaining assertions are immediate. □

Lemma 4.10. *Let G be a graph of order n with $\delta(G) \geq \frac{5}{6}n$. Suppose that the edges of G are 3-coloured and there are exactly 2 red components. Then there exists a monochromatic component of order at least $\frac{n}{2}$.*

Proof. By Lemma 4.8, we are done unless every vertex is incident to edges of all three colours. Therefore every vertex is in a red component. As there are only two red components, one of them must cover at least $\frac{n}{2}$ of the vertices. □

Lemma 4.11. *Let G be a graph of order n where the edges are 3-coloured and there is no monochromatic component of order at least $\frac{n}{2}$. Suppose that G has r red, b*

blue and g green components and there exist red components R_1 and R_2 such that $|R_1| + |R_2| < \frac{n}{2}$. Then there is a graph G' together with a 3-edge-colouring such that there are $(r-1)$ red, b blue and g green components, $\frac{\delta(G')}{|G'|} \geq \frac{\delta(G)}{|G|}$ and there is no monochromatic component in G' covering at least half of the vertices.

Proof. Note that, by Lemma 4.8, we may assume that every vertex is incident to all three colours.

Suppose first that there exists $v_1 \in R_1$ and $v_2 \in R_2$ such that $v_1v_2 \notin E(G)$. Let G' be a copy of G with the additional red edge v_1v_2. Then $\delta(G') \geq \delta(G)$ and all components in G' have the same number of vertices as they do in G with the exception of R_1 and R_2 which form a single component in G'. As $|R_1| + |R_2| < \frac{n}{2}$, G' contains no monochromatic component covering at least half of the vertices.

Now suppose that every vertex in R_1 is connected to every vertex in R_2. Fix $u \in R_1$ and $v \in R_2$ and, without loss of generality, assume that the edge uv is blue. Let G' be a 2-blow-up of G with same 3-edge-colouring. The vertex u in G corresponds to vertices u_1 and u_2 in G' and v corresponds to v_1 and v_2. Change the colour of the edge u_1v_1 from blue to red.

The red components corresponding to R_1 and R_2 in G' now form a single component of order $2(|R_1| + |R_2|) < n = \frac{1}{2}|G'|$. The vertices u_1 and v_1 still lie in the same blue component via the blue path $u_1v_2u_2v_1$ and so changing the colour of the edge u_1v_1 does not change the orders of the other components in G'. By Lemma 4.9, we have $\delta(G') = 2\delta(G) + 1$ and G' does not contain a monochromatic component of order at least $\frac{1}{2}|G'|$. As $|G'| = 2|G|$, it follows that $\frac{\delta(G')}{|G'|} > \frac{\delta(G)}{|G|}$. □

Lemma 4.11 means we may make the following assumption: in each colour, G either has 3 components or 4 components each of order exactly $\frac{n}{4}$.

Lemma 4.12. *Let G be a graph of order n where the edges of G are 3-coloured. Suppose that G has no monochromatic component of order at least $\frac{n}{2}$. If, in two* *colours, there are* 4 *components of order exactly* $\frac{n}{4}$ *in that colour, then* $\delta(G) < \frac{5}{6}n$.

Proof. Without loss of generality, suppose G has 4 red components and 4 blue components, each with order exactly $\frac{n}{4}$. By Lemma 4.8, every vertex lies in components of all three colours. Lemma 4.10 tells us that there must be at least 3 green components. The smallest green component has order at most $\frac{n}{3}$. Choose a vertex v in the smallest green component. Then we find that $d(v) \leq (\frac{n}{3} - 1) + 2(\frac{n}{4} - 1) < \frac{5}{6}n$. □

The above lemmas allow us to make the following assumptions about G:

- Every vertex is incident to an edge of every colour (Lemma 4.8).
- There are either 3 components in each colour or there are 3 components in two colours and 4 components of order $\frac{n}{4}$ in the third colour (Lemmas 4.10, 4.11 and 4.12).

4.4 Linear Programs for the Upper Bound

In Section 4.3, we reduced the proof of Theorem 4.7 to the following two cases:

1. Every vertex is incident to an edge of every colour. There are 3 components of each colour.
2. Every vertex is incident to an edge of every colour. There are 3 components in two of the colours (without loss of generality, red and blue) and 4 components of order exactly $\frac{n}{4}$ in the third colour (without loss of generality, green).

We now formulate these cases as collections of linear programs. More details about the code used to implement these linear programs may be found in Appendix A.

4.4.1 Three Components in Each Colour

We begin by considering the first case where there are 3 components in each of the three colours. Let the vertex sets of the red components be R_i, the blue components be B_j and the green components be G_k for $i, j, k \in [3]$. We know that every vertex v of G lies in the intersection $R_i \cap B_j \cap G_k$ for some $i, j, k \in [3]$ and so $d(v) \leq |R_i \cup B_j \cup G_k| - 1$ with equality if v is adjacent to every vertex in $R_i \cup B_j \cup G_k$. Proving Theorem 4.7 for the first case is equivalent to showing that Question 4.13 has an answer of $\alpha < \frac{5}{6}$.

Question 4.13. *What is the maximum value of α such that the following conditions can hold simultaneously?*

1. $|R_i|, |B_j|, |G_k| < \frac{n}{2} \qquad \forall i, j, k \in [3]$
2. $d(v) \geq \alpha n \qquad \forall v \in V(G)$

The addition of any missing edges between v and $R_i \cup B_j \cup G_k$ (appropriately coloured) will not change the number of vertices in each monochromatic component but may increase $\delta(G)$ and hence α. Therefore we may assume that $d(v) = |R_i \cup B_j \cup G_k| - 1$ for every vertex $v \in R_i \cap B_j \cap G_k$ (although it is important to note that $R_i \cap B_j \cap G_k$ may be empty).

We can avoid dependence on n by rescaling. Let $x_{ijk} = \frac{1}{n}|R_i \cap B_j \cap G_k|$ for each $i, j, k \in [3]$. For fixed $i \in [3]$, we find that $|R_i| = n \sum_{j=1}^{3} \sum_{k=1}^{3} x_{ijk}$ (and similarly for $|B_j|$ and $|G_k|$) and, for fixed $i, j, k \in [3]$, we have

$$\frac{1}{n}|R_i \cup B_j \cup G_k| = \sum_{\substack{i',\ j',\ k' \\ i'=i \text{ or } j'=j \text{ or } k'=k}} x_{i'j'k'}.$$

Using this notation and dividing through by n, the first condition in Question 4.13

becomes:

$$\sum_{j=1}^{3}\sum_{k=1}^{3} x_{ijk} < \frac{1}{2} \qquad \forall i \in [3]$$
$$\sum_{i=1}^{3}\sum_{k=1}^{3} x_{ijk} < \frac{1}{2} \qquad \forall j \in [3]$$
$$\sum_{i=1}^{3}\sum_{j=1}^{3} x_{ijk} < \frac{1}{2} \qquad \forall k \in [3]$$

In linear programs, the condition statements must consist of weak, rather than strict, inequalities in order to guarantee that the space of feasible solutions is closed and, if this space is non-empty, that an optimal solution exists. Relaxing the above conditions to allow equality will increase the space of feasible solutions. As the maximum value of α for our original problem will be at most the maximum value of α for the relaxed problem, showing that $\alpha < \frac{5}{6}$ for the relaxed problem is sufficient to prove Theorem 4.7.

The second condition holds whenever $R_i \cap B_j \cap G_k$ is non-empty. We obtain:

$$\begin{cases} \displaystyle\sum_{\substack{i',\ j',\ k' \\ i'=i \text{ or } j'=j \text{ or } k'=k}} x_{i'j'k'} \geq \alpha + \frac{1}{n} & \text{if } R_i \cap B_j \cap G_k \neq \emptyset \\ x_{ijk} = 0 & \text{otherwise.} \end{cases}$$

We would like to remove the dependence on n completely. We therefore relax the second condition by removing the $\frac{1}{n}$; we will obtain an upper bound of $\alpha \leq \frac{5}{6}$ which is sufficient to prove Theorem 4.7.

We encode whether $R_i \cap B_j \cap G_k$ is empty with an additional variable y_{ijk} by setting:

$$y_{ijk} = \begin{cases} 1 & \text{if } R_i \cap B_j \cap G_k \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

The variables y_{ijk} represent the pattern of intersections. For a fixed intersection pattern, an upper bound on α in Question 4.13 can be found by solving Linear Program 4.1.

$$
\begin{array}{lrl}
\textbf{maximise} & & \alpha \\
\textbf{subject to} & \displaystyle\sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3} x_{ijk} = 1 & \\
& \displaystyle\sum_{j=1}^{3}\sum_{k=1}^{3} x_{ijk} \leq \frac{1}{2} & \forall i \in [3] \\
& \displaystyle\sum_{i=1}^{3}\sum_{k=1}^{3} x_{ijk} \leq \frac{1}{2} & \forall j \in [3] \\
& \displaystyle\sum_{i=1}^{3}\sum_{j=1}^{3} x_{ijk} \leq \frac{1}{2} & \forall k \in [3] \\
& \displaystyle\sum_{\substack{i',\, j',\, k' \\ i'=i \text{ or } j'=j \text{ or } k'=k}} x_{i'j'k'} - \alpha \geq 0 & \forall i,j,k \in [3] \text{ such that } y_{ijk} = 1 \\
& x_{ijk} = 0 & \forall i,j,k \in [3] \text{ such that } y_{ijk} = 0 \\
& x_{ijk} \geq 0 & \forall i,j,k \in [3]
\end{array}
$$

Linear Program 4.1: 3 red, 3 blue and 3 green components.

We have already assumed that there are 3 components of each colour or else we would be done by Lemma 4.10. Therefore, we only need to consider intersection patterns that satisfy the following condition: for each $i \in [3]$, there exists $j, k \in [3]$ such that $y_{ijk} \neq 0$ (and similarly for j and k).

Using a computer, we ran the linear program for all valid intersection patterns (roughly 2^{27} linear programs were run) and found the maximum overall value of α and the optimal solutions corresponding to this value of α. The maximum value was $\alpha = \frac{5}{6}$.

There were five different optimal solutions and these are shown below in graph format in Figure 4.6. Each circle represents a set of vertices and circles of the same size contain the same number of vertices. The large circles contain twice as many vertices as the small circles. Where two circles are connected by striped lines, the edges between these vertex sets may be either of the two colours indicated.

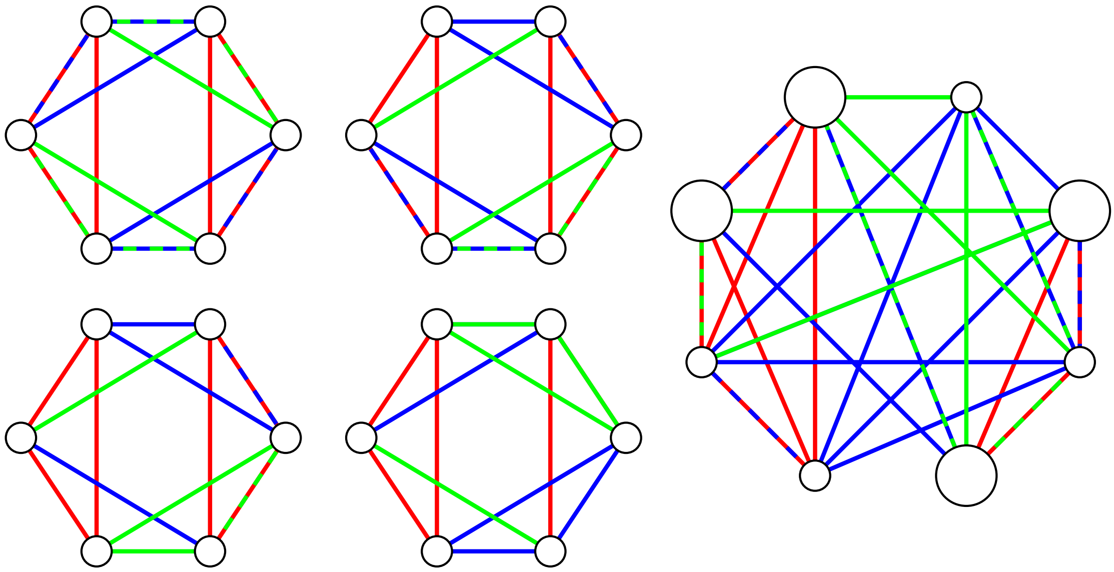


Figure 4.6: Optimal results of Linear Program 4.1.

4.4.2 Three Components in Two Colours; Four in the Third

Now we consider the case where there are 3 red, 3 blue and 4 green components and each green component has order exactly $\frac{n}{4}$. The set-up is very similar to the case above (see Linear Program 4.2). For a given intersection pattern y_{ijk}, the only difference in the linear program is the condition that each green component has size exactly $\frac{1}{4}$ (rather than just being at most $\frac{1}{2}$).

$$
\begin{aligned}
\textbf{maximise} \quad & \alpha \\
\textbf{subject to} \quad & \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{4} x_{ijk} = 1 \\
& \sum_{j=1}^{3}\sum_{k=1}^{4} x_{ijk} \le \frac{1}{2} \qquad \forall i \in [3] \\
& \sum_{i=1}^{3}\sum_{k=1}^{4} x_{ijk} \le \frac{1}{2} \qquad \forall j \in [3] \\
& \sum_{i=1}^{3}\sum_{j=1}^{3} x_{ijk} = \frac{1}{4} \qquad \forall k \in [4] \\
& \sum_{\substack{i',\, j',\, k' \\ i'=i \text{ or } j'=j \text{ or } k'=k}} x_{i'j'k'} - \alpha \ge 0 \quad \forall i, j \in [3], k \in [4] \text{ such that } y_{ijk} = 1 \\
& x_{ijk} = 0 \quad \forall i, j \in [3], k \in [4] \text{ such that } y_{ijk} = 0 \\
& x_{ijk} \ge 0 \quad \forall i, j \in [3], k \in [4]
\end{aligned}
$$

Linear Program 4.2: 3 red, 3 blue and 4 green components.

As above, we only considered intersection patterns where, for each $i \in [3]$, there exists $j \in [3], k \in [4]$ such that $y_{ijk} \neq 0$ (and similarly for j and k). Using a computer, we ran the linear program for all valid intersection patterns ($O(2^{30})$ linear programs were run) and found that the maximum overall value of α was $\frac{3}{4}$. As $\frac{3}{4}$ is strictly smaller than $\frac{5}{6}$, we may conclude that Theorem 4.7 holds in the case where there are four components in one colour and three components in the other two colours.

4.5 Conclusion

It is worth remarking that the constructions used in the proof of Theorem 4.6 are sharp in some cases. Suppose that there is a graph G on n vertices with no monochromatic component of order $\frac{n}{2}$ and $\delta(G) = \frac{5}{6}n - a$ for some a. Theorem 4.7 tells us that $a > 0$. If G' is a t-blow-up of G, then G' has no monochromatic component of order $\frac{tn}{2}$. By

Lemma 4.9, we find

$$\begin{aligned}\delta(G') &= t\delta(G) + (t-1)\\ &= \frac{5}{6}nt - at + (t-1).\end{aligned}$$

Theorem 4.7 tells us that $\delta(G') < \frac{5}{6}tn$. Combining these inequalities gives $a > 1 - \frac{1}{t}$ for every $t \in \mathbb{N}$ and so $a \geq 1$. Hence it follows that $\delta(G) \leq \left\lfloor \frac{5}{6}n - 1 \right\rfloor = \left\lfloor \frac{5}{6}n \right\rfloor - 1$. In the proof of Theorem 4.6, the graphs given for $n \equiv 1, 2, 3, 4 \mod 6$ each had minimum degree $\left\lfloor \frac{5}{6}n \right\rfloor - 1$.

For any $n \in \mathbb{N}$ and $k \geq 3$, let $f_k(n)$ be the maximum value such that there exists a graph G on n vertices with $\delta(G) = f_k(n)$ and a k-edge-colouring of G where every monochromatic component has order strictly less than $\frac{n}{k-1}$. In the proof of Theorem 4.6, we found that

$$\begin{aligned}
f_3(n) &= \left\lfloor \frac{5}{6}n \right\rfloor - 1 && \text{if } n \equiv 1, 2, 3, 4 \mod 6\\
f_3(n) &\in \left\{ \left\lfloor \frac{5}{6}n \right\rfloor - 2, \left\lfloor \frac{5}{6}n \right\rfloor - 1 \right\} && \text{if } n \equiv 0, 5 \mod 6.
\end{aligned}$$

We believe that $f_3(n) = \left\lfloor \frac{5}{6}n \right\rfloor - 1$ for all residue classes but we were unable to find examples when $n \equiv 0, 5 \mod 6$.

Theorems 4.6 and 4.7 prove that Conjecture 4.3 is false for $k = 3$ and that the correct constant is $\frac{5}{6}$. It is natural to ask what the correct bound is for other values of k. Unfortunately, although our methods extend in principle to 4 or more colours, the computational time needed to run all of the required linear programs makes it infeasible to do so. For example, when $k = 4$, a naive implementation of our approach would entail solving around 2^{240} linear programs. It would be nice to resolve Conjecture 4.3 for more values of k.

Chapter 5

Monochromatic Paths in the Complete Infinite Symmetric Digraph

5.1 Introduction

Let $K_{\mathbb{N}}$ be the complete graph on the positive integers and $\vec{K}_{\mathbb{N}}$ be the complete symmetric digraph on the positive integers. Recall from Chapter 1 that the *upper density* of a set $A \subseteq \mathbb{N}$ is

$$\bar{d}(A) = \limsup_{n \to \infty} \frac{|A \cap [n]|}{n}$$

where $[n] = \{1, \ldots, n\}$ and that, for a graph or digraph G with vertex set $V(G) \subseteq \mathbb{N}$, we define the *upper density* of G to be that of $V(G)$. Throughout this chapter, by a k-colouring, we mean a k-edge-colouring. In a 2-colouring, we will assume that the colours are red and blue.

Raynaud [49] showed that, in any 2-colouring of $\vec{K}_n$, there is a directed Hamiltonian cycle C with the following property: there are two vertices a and b such that the directed path from a to b along C is red and the directed path from b to a along

C is blue.

In this chapter, we will be interested in the infinite directed case. In particular, we will be considering edge-colourings of $\vec{K}_{\mathbb{N}}$ and will prove a variety of results relating to the upper density of paths in $\vec{K}_{\mathbb{N}}$.

Let $P = v_1v_2...$ be a path in $\vec{K}_{\mathbb{N}}$. We say that P is a *directed path* if every edge in P is oriented in the same direction. By the *length* of a path, we mean the number of edges in the path. Note that, for us, an infinite path has only one infinite end. DeBiasio and McKenney [16] recently proved the following result.

Theorem 5.1. *For every $\varepsilon > 0$, there exists a 2-colouring of $\vec{K}_{\mathbb{N}}$ such that every monochromatic directed path has upper density less than ε.*

DeBiasio and McKenney also asked the following natural question: does there exist a 2-colouring of $\vec{K}_{\mathbb{N}}$ in which every monochromatic directed path has upper density 0? In Section 5.2, we will give a positive answer to this question.

Theorem 5.2. *There exists a 2-colouring of $\vec{K}_{\mathbb{N}}$ such that every monochromatic directed path has upper density 0.*

In light of this result, it is natural to ask under what conditions we can guarantee the existence of a monochromatic path of positive density. It is easy to see (from Ramsey's Theorem) that every 2-colouring of $\vec{K}_{\mathbb{N}}$ contains a monochromatic directed path of infinite length. In Section 5.3, we will consider what happens if we restrict the length of directed paths in one of the colours. In particular, we will prove the following density result.

Theorem 5.3. *Take any 2-colouring of $\vec{K}_{\mathbb{N}}$ that does not contain a red directed path of length r. Then there exists a blue directed path with upper density at least $\frac{1}{r}$.*

This result is the best possible as shown by the following construction. Divide the vertices of $\vec{K}_{\mathbb{N}}$ into sets $A_1, \ldots, A_r$ based on their residue modulo r so, for any $n \in \mathbb{N}$, we have $n \in A_i$ if and only if $n \equiv i \mod r$. Consider distinct vertices $m, n \in \mathbb{N}$. If $m, n \in A_i$, then colour both (m, n) and (n, m) blue. If $m \in A_i$, $n \in A_j$ and $i < j$, then colour (m, n) blue and (n, m) red (see Figure 5.1). We call this colouring C_r. It should be noted that each of the sets A_i has density $\frac{1}{r}$.

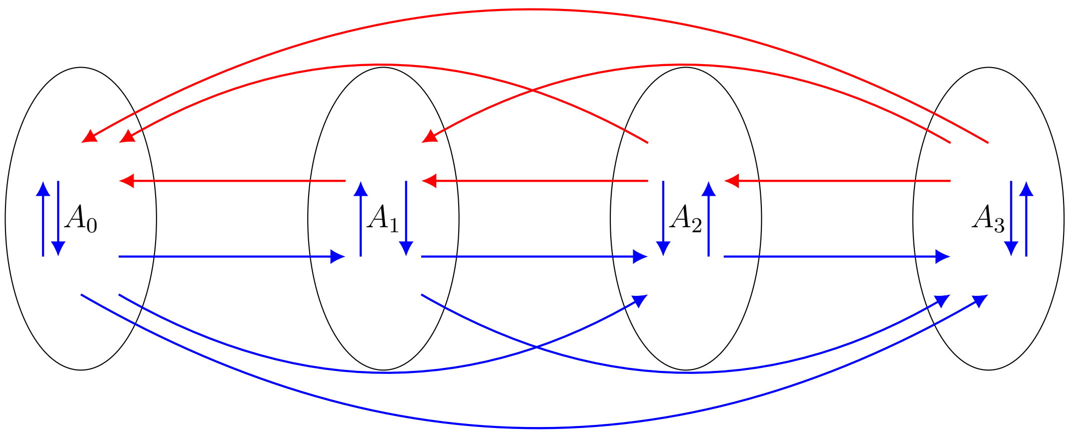


Figure 5.1: Extremal colouring C_4.

It is straightforward to see that the longest red directed path has length $r-1$. Any blue directed path has infinite intersection with at most one of the sets A_i and therefore has upper density at most $\frac{1}{r}$.

In Section 5.4, we will consider the stability of this result. We will prove the following theorem.

Theorem 5.4. *Take any 2-colouring of $\vec{K}_{\mathbb{N}}$ in which there are no red directed paths of length r and every blue directed path has upper density at most $\frac{1}{r}$. Then, there exists a finite set of vertices U such that the 2-colouring induced on $\mathbb{N} \backslash U$ is isomorphic to the 2-colouring C_r described above.*

Finally, in Section 5.5, we will extend the result of Theorem 5.3 to include edge-colourings of $\vec{K}_{\mathbb{N}}$ with $k + 1$ colours for any $k \geq 2$.

Theorem 5.5. *Fix $k \geq 2$ and consider any $(k+1)$-colouring of $\vec{K}_{\mathbb{N}}$ using colours $c_1, \ldots, c_{k+1}$. Suppose that, for every $t \in [k]$, there is no c_t-coloured directed path of length r_t. Then there exists a directed path of colour c_{k+1} with upper density at least $\prod_{t=1}^{k} \frac{1}{r_t}$.*

5.2 All Monochromatic Paths have Density 0

Proof of Theorem 5.2. We colour the edges of $\vec{K}_{\mathbb{N}}$ in the following way. Let $m, n \in \mathbb{N}$ be distinct positive integers. Set $t = \min\{s \in \mathbb{N} : m \not\equiv n \mod 2^s\}$. Exchanging m and n if necessary, we may assume that $m \equiv x \mod 2^t$ where $x \in \{0, \ldots, 2^{t-1} - 1\}$ and $n \equiv 2^{t-1} + x \mod 2^t$. We colour (m, n) red and (n, m) blue.

Let $P = v_1 v_2 \ldots$ be any monochromatic directed path. If P is a finite path, then $\bar{d}(P) = 0$. Therefore, we may assume that P is an infinite path. If the edges in P are oriented from v_i to v_{i+1}, then P is a *forward* path; otherwise P is a *backward* path. Without loss of generality, P is red.

Induction Hypothesis. *For any $k \in \mathbb{N}$, there exists $i \in \{0, \ldots, 2^k - 1\}$ and $j \in \mathbb{N}$ such that $v_{j'} \in \{n \in \mathbb{N} : n \equiv i \mod 2^k\}$ for every $j' \geq j$. Hence, $\bar{d}(P) \leq 2^{-k}$.*

Base Case. For $i \in \{0, 1\}$, let $A_i = \{n \in \mathbb{N} : n \equiv i \mod 2\}$. The sets A_0 and A_1 partition the vertices of $\vec{K}_{\mathbb{N}}$. Suppose first that P is a forward path. If P contains a vertex $v_j \in A_1$, then we must have $v_j' \in A_1$ for every $j' \geq j$ because P is a red forward path. If P does not contain a vertex from A_1, then P must be completely contained within A_0. When P is a backward path, we reach a similar conclusion: either P is entirely contained within A_1 or P contains a vertex $v_j \in A_0$ and hence $v_{j'} \in A_0$ for every $j' \geq j$. Regardless of the direction of P, for some $i \in \{0, 1\}$, all but a finite number of vertices of P lie in A_i and so $\bar{d}(P) \leq \bar{d}(A_i) = \frac{1}{2}$.

Inductive Step. Fix $k \geq 2$ and partition the vertices of $\vec{K}_{\mathbb{N}}$ into 2^k sets $A_0, \ldots, A_{2^k - 1}$ based on their residue modulo 2^k (see Figure 5.2). By the inductive hypothesis, there exists $i \in \{0, \ldots, 2^{k-1}-1\}$ and $j \in \mathbb{N}$ such that $v_{j'} \in A_i \cup A_{2^{k-1}+i}$ for every $j' \geq j$. By using the same argument as in the base case, we find that, for some $i' \in \{i, 2^{k-1}+i\}$, either $v_{j'} \in A_{i'}$ for every $j' \geq j$ or there exists $t > j$ such that $v_{t'} \in A_{i'}$ for every $t' \geq t$. Hence, there exists $i' \in \{0, \ldots, 2^k - 1\}$ such that all but a finite number of vertices in P lie in $A_{i'}$ and so $\bar{d}(P) \leq \bar{d}(A_{i'}) = 2^{-k}$.

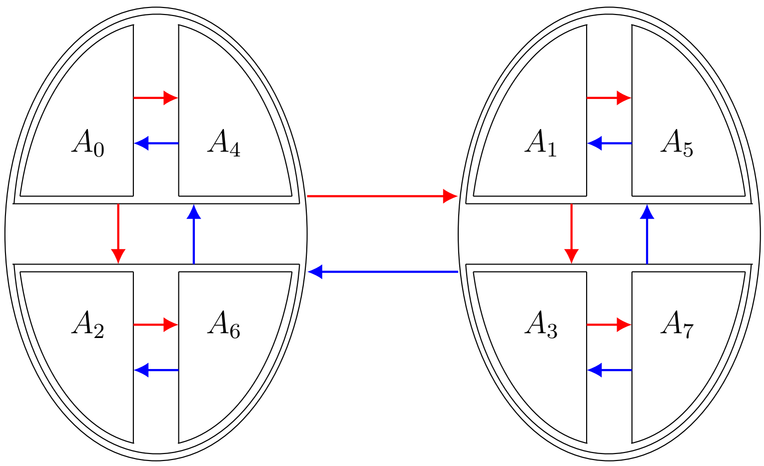


Figure 5.2: Diagram showing the edges between sets for $k = 3$.

The induction hypothesis holds for every $k \in \mathbb{N}$. Therefore, if P is any monochromatic directed path, we have that the upper density of P is at most 2^{-k} for every $k \in \mathbb{N}$. Hence, P has upper density 0. □

5.3 Restricting Red Path Length

Proof of Theorem 5.3. Fix a 2-colouring of $\vec{K}_{\mathbb{N}}$ and assume that there is no red directed path of length r. Partition the vertices into sets $A_0, \ldots, A_{r-1}$ where $n \in A_i$ if and only if the longest directed red path from n has length i.

Claim 5.6. *Let $v \in A_i$. Then the red indegree of v in A_i is at most i.*

Proof. Take a directed red path starting at v of maximal length $vv_{i-1} \ldots v_0$. Let $U = \{u \in A_i : (u, v) \text{ is red}\}$. For every $u \in U$, there must exist some $t \in \{0, \ldots, i-1\}$

such that $u = v_t$ or else $uvv_{i-1}\dots v_0$ would be a directed red path of length $i+1$ from u, which contradicts $u \in A_i$. Hence $|U| \leq i$. $\diamondsuit$

At least one of the sets A_i must have upper density at least $\frac{1}{r}$. Choose such an A_i and let $B = \{v \in A_i : v \text{ has finite blue outdegree in } A_i\}$.

Claim 5.7. *$|B| \leq i$.*

Proof. Suppose not and consider a subset $\{b_1, \dots, b_{i+1}\} \subseteq B$ of size $i+1$. We may list the vertices of A_i in order as $v_1, v_2, \dots$. For every $1 \leq j \leq i+1$, we define $t_j = \min\{k \in \mathbb{N} : \forall k' \geq k \text{ the edge } (b_j, v_{k'}) \text{ is red}\}$. This is well-defined because b_j has finite blue outdegree. Taking $t = \max\{t_1, \dots, t_{i+1}\}$, we find that the vertex v_t has red indegree at least $i+1$ in A_i, which contradicts the result of Claim 5.6. $\diamondsuit$

As A_i is infinite and B is finite, $A_i \backslash B$ is also infinite and $\bar{d}(A_i) = \bar{d}(A_i \backslash B)$. We may therefore assume, without loss of generality, that $B = \emptyset$. Every vertex in A_i has infinite blue outdegree in A_i. Listing the vertices of A_i in order $v_1, v_2, \dots$, we create a blue directed path $P = \{p_j\}$ as follows:

- Set $p_1 = v_1$.
- Given p_j, let $t = \min\{k : v_k \notin P\}$. If the edge (p_j, v_t) is blue, set $p_{j+1} = v_t$. If not, there exists $u \notin P$ such that the edges (p_j, u) and (u, v_t) are both blue. Such a vertex u always exists because p_j has an infinite number of blue out-neighbours and v_t has at most i red in-neighbours. Set $p_{j+1} = u$ and $p_{j+2} = v_t$.

The blue directed path P contains every vertex of A_i and hence has upper density $\bar{d}(A_i)$. $\square$

5.4 Stability Result

Proof of Theorem 5.4. Fix a 2-colouring of $\vec{K}_{\mathbb{N}}$ such that there is no red directed path of length r and every blue directed path has density at most $\frac{1}{r}$. Partition the vertices into sets $A_0, \dots, A_{r-1}$ where $v \in A_i$ if and only if the longest directed red path from v has length i.

By using the same argument as in the proof of Theorem 5.3, we also know that, if A_i is an infinite set, then there exists a blue directed path within A_i of density $\bar{d}(A_i)$. As there are r sets A_i, we must have $\bar{d}(A_i) = \frac{1}{r}$ for every $i \in \{0, \dots, r-1\}$.

We will prove by induction that, by removing a finite number of vertices, all edges from A_{i-1} to A_i are blue and all edges from A_i to A_{i-1} are red for every $i \in [r-1]$ (see Figure 5.3).

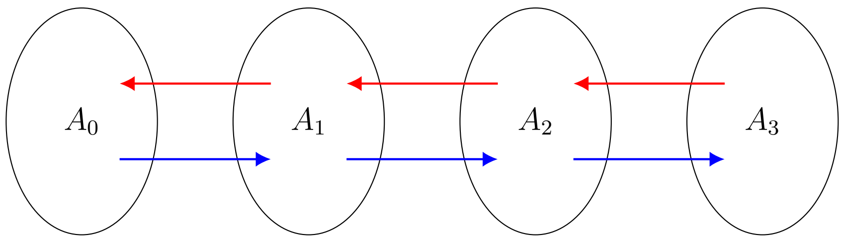


Figure 5.3: Result of the induction for $r = 4$.

Induction Hypothesis. *For fixed $k \geq 1$, by removing a finite number of vertices, $A_0 \cup \cdots \cup A_k$ has the following structure: for every $i \in [k]$, all edges from A_{i-1} to A_i are blue and all edges from A_i to A_{i-1} are red.*

Base Case. Let $k = 1$. By definition of A_0, all edges from A_0 to A_1 are blue. If there are no blue edges from A_1 to A_0, we are done. Otherwise, we form a matching M of blue edges from A_1 to A_0 by adding blue edges one by one until we cannot add any more.

Suppose M is a finite matching. Let B be the set of vertices which are incident to an edge in M. By definition, every blue edge from A_1 to A_0 has at least one endpoint in B. Therefore, by removing the vertices in B, we will ensure that every edge from A_1 to A_0 is red as required.

Suppose instead that M is an infinite matching. We noted earlier that A_0 contains a blue directed path $P = \{p_j\}$ and A_1 contains a blue directed path $Q = \{q_\ell\}$, both of which have density $\frac{1}{r}$. We create a blue directed path S as follows:

- Pick an edge $(q_\ell, p_j) \in M$ where $j > 1$. Set $S = p_1 \dots p_{j-1} q_1 \dots q_\ell p_j$. The edge (p_{j-1}, q_1) is blue because all edges from A_0 to A_1 are blue.

- Suppose the final two vertices of S are $q_{\ell_1} p_{j_1}$ where $(q_{\ell_1}, p_{j_1}) \in M$. Pick an edge $(q_{\ell_2}, p_{j_2}) \in M$ such that $\ell_1 < \ell_2$ and $j_1 < j_2$. Such an edge exists because M is an infinite matching and S only contains a finite number of vertices. We note that the edge $(p_{j_2-1}, q_{\ell_1+1})$ is blue because it is from A_0 to A_1. Add the subpath $p_{j_1+1} \dots p_{j_2-1} q_{\ell_1+1} \dots q_{\ell_2} p_{j_2}$ to the end of S.

By construction, the blue directed path S uses every vertex in both P and Q and so $\bar{d}(S) = \bar{d}(P) + \bar{d}(Q) = \frac{2}{r}$.

We assumed that every blue path had density at most $\frac{1}{r}$ so this is a contradiction. Therefore, any blue matching from A_1 to A_0 is finite and so we are done.

Inductive Step. Now let $k > 1$. By our induction hypothesis, the claim is true for $k-1$ so, by removing a finite number of vertices, we may assume that all edges from A_{i-1} to A_i are blue and all edges from A_i to A_{i-1} are red for every $i \in [k-1]$.

Consider $A_{k-1} \cup A_k$ and suppose there is a red edge from $x \in A_{k-1}$ to $y \in A_k$. All edges from y to $A_{k-1} \backslash \{x\}$ are blue or else we would have a red directed path of length $k+1$ from x which contradicts $x \in A_{k-1}$.

In the original graph, there is a red directed path P of length k from y because $y \in A_k$. Let $P = p_0 p_1 \dots p_k$ with $p_0 = y$. If, for every $t \in [k]$, we have that $p_t \neq x$, then $x y p_1 \dots p_k$ is a red directed path of length $k+1$ from x, which contradicts $x \in A_{k-1}$. Therefore, $x = p_t$ for some $t \in [k]$. No edges are removed when we apply the induction hypothesis and so there are at least k vertex-disjoint paths of length $k-1$ from x. Therefore, there exists a red directed path Q of length $k-1$ from x which is vertex-disjoint from $y p_1 \dots p_{t-1}$. The path $y p_1 \dots p_{t-1} \cup Q$ is a red directed path from y of length $t + k - 1$. If $t \neq 1$, this path has length strictly larger than

k, which contradicts $y \in A_k$. Therefore we must have that $t = 1$ and the edge (y, x) must be red.

Consider any vertex $u \in A_{k-1}$ with $u \neq x$. We can find a red directed path of length $k - 1$ from x that does not contain u or y. To prevent a red directed path of length $k+1$ from u, the edge (u, y) must be blue. Therefore, all edges from $A_{k-1} \backslash \{x\}$ to y must be blue (see Figure 5.4).

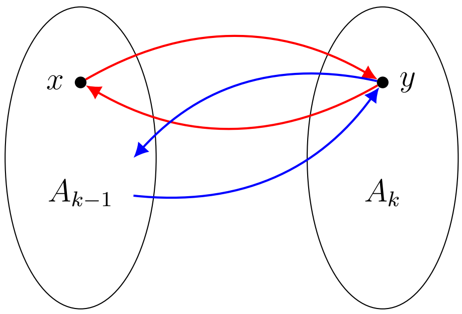


Figure 5.4: The colours of edges incident to y.

Hence, if there is a red edge xy from A_{k-1} to A_k, the corresponding backwards edge is also red and all edges between $A_{k-1} \backslash \{x\}$ and y in both directions are blue. Suppose there are an infinite number of red vertex-disjoint edges from A_{k-1} to A_k. As both A_{k-1} and A_k contain a blue directed path of density $\frac{1}{r}$, we can use these and the pairs of vertices corresponding to the red edges to create a blue directed path of density $\frac{2}{r}$. This contradicts our assumption that all blue directed paths have density at most $\frac{1}{r}$. Therefore, there are only a finite number of red vertex-disjoint edges from A_{k-1} to A_k. By removing a finite number of vertices, we can ensure that all edges from A_{k-1} to A_k are blue.

In order to show that we can remove a finite number of vertices to get all edges from A_k to A_{k-1} to be red, we use exactly the same argument as in the base case.

It follows directly from the inductive argument above that we can remove a finite number of vertices in order to obtain the following structure: for every $i \in [r - 1]$, all edges from A_{i-1} to A_i are blue and all edges from A_i to A_{i-1} are red.

We now consider the edges between A_i and A_j where $i < j$ and $i \neq j-1$. All edges from A_i to A_j are blue. If this were not the case, we could find a red directed path of length $j+1$ from a vertex in A_i, which is a contradiction. Next, we will show that we can remove a finite number of vertices to get all edges from A_j to A_i being red. If there are no blue edges from A_j to A_i, we are done. Otherwise, we may greedily construct a maximal matching M of blue edges from A_j to A_i. Using the same argument as the induction base case, we get that M must be a finite matching; if this was not the case, then we would be able to find a blue directed path of density $\frac{2}{r}$, which contradicts our original assumption. By deleting the vertices which are incident to an edge in M, we will remove all of the blue edges from A_j to A_i. We only deleted a finite number of vertices and now all of the edges from A_j to A_i are red as required.

There are only a finite number of pairs (A_i, A_j) to consider. Therefore, by removing a finite number of vertices, we can ensure that, for every $i < j$, all edges from A_i to A_j are blue and all edges from A_j to A_i are red.

Finally, we look at the edges between vertices in the same set. Suppose, for some $i \in \{0, \dots, r-1\}$ there exist $x, y \in A_i$ such that the edge from x to y is red. Then we can find a red directed path of length $i+1$ from x, which contradicts $x \in A_i$. Hence, all edges between vertices in the same set A_i are blue. $\square$

5.5 Restricting Path Length for Many Colours

Proof of Theorem 5.5. Let $k \geq 2$. Fix a $(k+1)$-colouring of $\vec{K}_{\mathbb{N}}$ where the colours are $c_1, \dots, c_{k+1}$. Assume that, for every $t \in [k]$, there is no c_t-coloured directed path of length r_t.

Induction Hypothesis. *For fixed $i \in [k]$, there exists a set $A \subseteq \mathbb{N}$ such that, for each $t \in [i]$, the c_t-indegree of vertices in A is at most r_t. Furthermore, $\bar{d}(A) \geq \prod_{t=1}^{i} \frac{1}{r_t}$.*

Base Case. Partition the vertices into sets $A_0, \ldots, A_{r_1-1}$ where $n \in A_j$ if and only if the longest directed path of colour c_1 from n has length j. Fix j such that $\bar{d}(A_j) \geq \frac{1}{r_1}$. By Claim 5.6 in the proof of Theorem 5.3, we find that the c_1-indegree of vertices in A_j is at most $j \leq r_1$.

Inductive Step. Let $i > 1$. By our induction hypothesis, there exists a set $A \subseteq \mathbb{N}$ with $\bar{d}(A) \geq \prod_{t=1}^{i-1} \frac{1}{r_t}$ such that, for every $t \in [i-1]$, the c_t-indegree of vertices in A is at most r_t. Partition A into r_i sets $A_0, \ldots, A_{r_i-1}$ where $n \in A_j$ if and only if the longest directed path of colour c_i from n in A has length j. Fix j such that $\bar{d}(A_j) \geq \frac{1}{r_i}\bar{d}(A)$. Arguing again as in Claim 5.6 in the proof of Theorem 5.3, we find that the c_i-indegree of vertices in A_j is at most $j \leq r_i$.

Let A be the set obtained by applying the induction hypothesis to $i = k$. Let $B = \{v \in A : v \text{ has finite } c_{k+1}\text{-outdegree in } A\}$. Using a similar argument to Claim 5.7 in the proof of Theorem 5.3, we find that $|B| \leq \sum_{t=1}^{k} r_t$.

As in the proof of Theorem 5.3, we may assume, without loss of generality, that $B = \emptyset$. We list the vertices of A in order $v_1, v_2, \ldots$ and create a directed path $P = \{p_j\}$ of colour c_{k+1} as follows:

- Set $p_1 = v_1$.
- Given p_j, let $s = \min\{s' : v_{s'} \notin P\}$. If the edge (p_j, v_s) is colour c_{k+1}, set $p_{j+1} = v_s$. If not, there exists $u \notin P$ such that the edges (p_j, u) and (u, v_s) are both colour c_{k+1} because p_j has an infinite number of out-neighbours with colour c_{k+1} and v_s has at most $\sum_{t=1}^{k} r_t$ in-neighbours which are not colour c_{k+1}. Set $p_{j+1} = u$ and $p_{j+2} = v_s$.

The directed path P of colour c_{k+1} contains every vertex of A and hence has upper density $\bar{d}(A)$. $\square$

This result is the best possible as shown by the following construction. Partition the vertices of $\vec{K}_{\mathbb{N}}$ into $\prod_{t=1}^{k} r_t$ sets based on their residue modulo r_t for each $t \in [k]$. The vertex $n \in \mathbb{N}$ is in the set $A_{i_1,i_2,\ldots,i_k}$ if and only if, for each $t \in [k]$, $n \equiv i_t \mod r_t$.

Consider vertices $m, n \in \mathbb{N}$. If $m, n \in A_{i_1,\ldots i_k}$, then colour both (m,n) and (n,m) colour c_{k+1}. Otherwise, suppose that $m \in A_{i_1,\ldots,i_k}$ and $n \in A_{j_1,\ldots,j_k}$. We define $t = \min\{t' \in [k] : i_{t'} \neq j_{t'}\}$. If $i_t < j_t$, then colour (m,n) colour c_{k+1} and (n,m) colour c_t (see Figure 5.5).

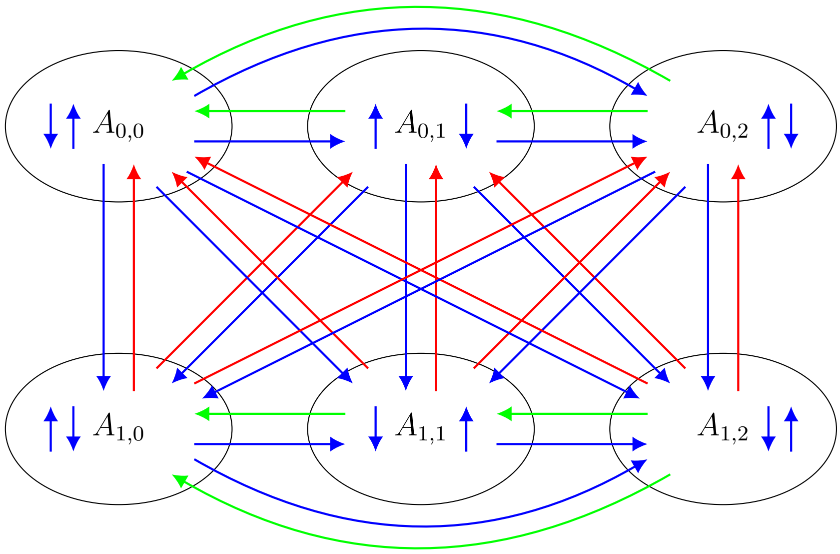


Figure 5.5: Extremal colouring when $r_1 = 2$ and $r_2 = 3$.

It is straightforward to see that, for every $i \in [k]$, the longest directed path of colour c_t has length $r_t - 1$. Any directed path of colour c_{k+1} has infinite intersection with at most one of the sets $A_{i_1,\ldots,i_k}$ and hence has upper density at most $\prod_{t=1}^{k} \frac{1}{r_t}$.

5.6 Conclusion

Consider a 2-colouring of $\vec{K}_{\mathbb{N}}$ that does not contain a red directed path of length r and partition the vertices into r sets, $A_0, \ldots, A_{r-1}$, where $v \in A_i$ if and only if the longest directed red path from v has length i. It follows from the proof of Theorem 5.3 that, if A_i is infinite, there is a blue directed path covering all but a finite number of vertices in A_i. Hence, there exist at most r disjoint blue paths covering all but a finite number of vertices of $\vec{K}_{\mathbb{N}}$.

In fact, if $r < 4$, it is possible to prove that we may cover all of the vertices of $\vec{K}_{\mathbb{N}}$ with at most r vertex-disjoint blue paths. We therefore make the following conjecture.

Conjecture 5.8. *Let $r \geq 1$. Take any 2-colouring of $\vec{K}_{\mathbb{N}}$ that does not contain a red directed path of length r. Then the vertices of $\vec{K}_{\mathbb{N}}$ can be covered by at most r vertex-disjoint blue directed paths.*

This conjecture has recently been proved to be correct [9]. More details can be found in Section 5.7.

In this chapter, we have restricted our attention to directed paths. Let $P = v_1v_2...$ be an orientation of a path in $\vec{K}_{\mathbb{N}}$. We say that v_i is a *switch* if either the indegree or the outdegree of v_i in P is 0. The endpoints of P are always switches. If P is a directed path, then the endpoints are the only switches in P. DeBiasio and McKenney [16] proved that Theorem 5.1 holds for any orientation of a path with finitely many switches. It should be noted that Theorem 5.2 holds for this broader class of paths.

5.7 Addendum

Shortly after the work in this chapter was completed, Bürger and Pitz [10] proved that Conjecture 5.8 is true and that it also extends to $(k+1)$-colourings of $\vec{K}_{\mathbb{N}}$ in the natural way.

Let $\vec{K}$ be the complete symmetric digraph on a finite or countably infinite number of vertices. The author, Bürger, DeBiasio and Pitz [9] proved the following theorem which nicely generalizes Theorem 5.3 and the results of Bürger and Pitz.

Theorem 5.9. *Let $c\colon E(\vec{K}) \to [k+1]$ be an edge-colouring of $\vec{K}$ with no directed path of length r_i in colour i for any $i \in [k]$. Then there is a partition of the vertices of $\vec{K}$ into $\prod_{i\in[k]} r_i$ complete symmetric digraphs monochromatic in colour $k+1$.*

The proof of this theorem relies on the following result of Gallai [24], Hasse [36], Roy [50] and Vitaver [56].

Theorem 5.10. *Let D be a finite or (not-necessarily-countably) infinite digraph and let G be the underlying graph of D. If the longest path in D has length r where $0 \leq r < \infty$, then $\chi(G) \leq r+1$.*

Proof of Theorem 5.9. Let $k \geq 0$ and colour the edges of $\vec{K}$ with $k+1$ colours so that there is no directed path of length r_i in colour i for any $i \in [k]$. The result will follow by induction.

Induction Hypothesis. *There is a partition of $\vec{K}$ into $\prod_{i \in [k]} r_i$ complete symmetric digraphs monochromatic in colour $k+1$.*

Base Case. If $k = 0$, then all of the edges are the same colour (i.e. colour $k+1 = 1$). The result follows because $\prod_{i=1}^{k} r_i = 1$.

Inductive Step. Let $k \geq 1$. Suppose that the result holds for all $(\ell+1)$-colourings satisfying the required path length condition with $\ell < k$.

Consider an $(k+1)$-colouring in which every path of colour $1 \leq i \leq k$ has length at most $r_i - 1$. Let D be the digraph induced by the edges of colour k. By applying Theorem 5.10 to D, we obtain a partition $\{U_0, \ldots, U_{r_k-1}\}$ of the vertices of D such that each U_j is an independent set. As $V(D) = V(\vec{K})$, it follows that $\{U_0, \ldots, U_{r_k-1}\}$ partitions the vertices of $\vec{K}$ so that each U_j contains no edges of colour k. Hence, each U_j is a k-coloured complete symmetric digraph such that, for all $i \in [k-1]$, every path of colour i has length at most $r_i - 1$. By applying induction to each U_j, we get a partition of U_j into $\prod_{i=1}^{k-1} r_i$ complete symmetric digraphs of colour $k+1$. Combining these, we obtain a partition of $\vec{K}$ into a total of $r_k \prod_{i=1}^{k-1} r_i = \prod_{i=1}^{k} r_i$ complete symmetric digraphs of colour $k+1$.

□

Chapter 6

Size Reconstructibility of Graphs

6.1 Introduction

Throughout this chapter, all graphs are finite and undirected with no loops. Recall from Chapter 1 that, for any graph G and any vertex $v \in V(G)$, the *card* $G - v$ is the subgraph of G obtained by removing the vertex v and all edges incident to v. The multiset of all unlabelled cards of G is called the *deck*, $\mathcal{D}(G)$, and has size n.

A parameter of a graph is *reconstructible* if it can be calculated from its deck. In this chapter, we will focus on reconstructing the number of edges in a graph from an incomplete deck of cards. In a recent paper, Brown and Fenner [8] showed that, for $n \geq 29$, only $n - 2$ cards are required to determine the size of a graph. Woodall [57] found that, for any $p \geq 3$ and n sufficiently large, if two graphs on n vertices have $n - p$ common cards, then the number of edges in these two graphs differs by at most $p - 2$.

In Section 6.2, we will improve on both results by showing that the size of a graph is reconstructible with as many as $\frac{1}{20}\sqrt{n}$ missing cards. In particular, we will prove the following theorem.

Theorem 6.1. *For n sufficiently large and $k \leq \frac{1}{20}\sqrt{n}$, the number of edges m of a* *graph G on n vertices is reconstructible from any $n-k$ cards.*

We will also consider the following game played against an adversary. The adversary chooses a graph G of order n and gives us a collection of n cards, each showing a graph on $n-1$ vertices. We are told that there are $n-k$ *true cards*, which come from the deck $\mathcal{D}(G)$. The other k cards are *false cards*, which can depict any graph of order $n-1$. We win if we are able to reconstruct the size of G; otherwise the adversary wins. When are we guaranteed to win regardless of the graph G and the cards given by the adversary? This turns out to be a corollary of Theorem 6.1.

Corollary 6.2. *Let n be sufficiently large and $k \leq \frac{1}{40}\sqrt{n}$. The number of edges m of a graph G on n vertices is reconstructible from any collection $\mathcal{C}$ of cards where $n-k$ are true and k are false.*

6.2 Size Reconstruction from $n-k$ Cards

Throughout this section, G will be a graph of order n and size $m = e(G)$, where m is unknown. The vertex set of G will be $V(G) = \{v_1, \ldots, v_n\}$ and we will write G_i for the card $G - v_i$. We will assume that we are given the cards $G_1, \ldots, G_{n-k}$. For any graph G', let the number of vertices of degree t be $d_t(G') = |\{v \in V(G') : d_{G'}(v) = t\}|$. For convenience, we will write $d(v) = d_G(v)$ and $d_t = d_t(G)$. Note that d_t is unknown for every t.

6.2.1 Proof Overview

Using the cards we have been given, we first obtain a upper bound $\widetilde{m}$ on the number of edges m. We then show that this upper bound is close to m; writing $\alpha = \widetilde{m} - m$, we show that $0 \leq \alpha < 2k$. We use this to estimate d_t, the number of vertices of degree t in G. If we knew the number of edges m, then we could calculate the degree of each vertex v_i from its card G_i as $d(v_i) = m - e(G_i)$. Instead, we use $\widetilde{m}$ to estimate the degree of the vertex v_i missing from each of the cards we have been given. The estimated degree of v_i is $\widetilde{d}(v_i) = \widetilde{m} - e(G_i)$ and we then count the number of vertices

$$\widetilde{d}_t = \left|\{i \in \{1, \ldots, n-k\} : \widetilde{d}(v_i) = t\}\right|$$

with estimated degree t from the cards we have. Since $m \leq \widetilde{m}$, our estimate $\widetilde{d}(v_i)$ may be larger than the actual degree of vertex v_i. This means that the actual sequence (d_t) has been shifted to the right by α. Moreover, some of the cards are missing so, after applying the shift, we have $\sum_{t=0}^{n-1} |d_t - \widetilde{d}_{t+\alpha}| = k$.

Our goal is to discover the shift α since, together with $\widetilde{m}$, this allows us to calculate $m = \widetilde{m} - \alpha$. In order to do this, we construct d_t exactly for many values of t and match these known values to the estimated sequence $\widetilde{d}_t$. We are then able to find α.

6.2.2 Preliminary Results

Definition. Set $\widetilde{m} = \left\lfloor \frac{1}{n-2-k} \sum_{i=1}^{n-k} e(G_i) \right\rfloor$.

The quantity $\widetilde{m}$ is an estimate for the number of edges in G and can be calculated directly from the cards $G_1, \ldots, G_{n-k}$.

Lemma 6.3. *$\widetilde{m} = m + \alpha$ where $0 \leq \alpha \leq \left\lfloor \frac{k(n-1)}{n-2-k} \right\rfloor$.*

Note that if $k = o(n)$ then the error $\alpha \leq (1 + o(1))k$.

Proof of Lemma 6.3. Suppose that we have the entire deck of G. Every edge of G is on exactly $n-2$ cards and so $\sum_{i=1}^{n} e(G_i) = (n-2)m$. Furthermore, for each $v_i \in V(G)$, we have $d(v_i) = m - e(G_i)$. It follows that $\sum_{i=1}^{n-k} e(G_i) = (n-2-k)m + \sum_{j=1}^{k} d(v_{n-k+j})$. Thus

$$\widetilde{m} = \left\lfloor \frac{1}{n-2-k} \sum_{i=1}^{n-k} e(G_i) \right\rfloor = m + \left\lfloor \frac{1}{n-2-k} \sum_{j=1}^{k} d(v_{n-k+j}) \right\rfloor .$$

We know $0 \leq d(v_j) \leq n-1$ for every $v_j \in V(G)$. Since $\alpha = \left\lfloor \frac{1}{n-2-k} \sum_{j=1}^{k} d(v_{n-k+j}) \right\rfloor$, the result follows. $\square$

Definition. For $0 \le t \le n-1$, define s_t to be the total number of vertices of degree t seen on the cards $G_1, \ldots, G_{n-k}$.

Note that s_t can be calculated from the cards we have been given.

Lemma 6.4. *Let $\varepsilon_t = (n-1-t)d_t + (t+1)d_{t+1} - s_t$. Then $0 \le \varepsilon_t \le k(d_t + d_{t+1})$.*

Proof. Consider the entire deck of G and let $v \in V(G)$ be a vertex of degree t. Then v appears as a vertex of degree $t-1$ on exactly t cards and as a vertex of degree t on $n-t-1$ cards (and does not appear on its own card). Hence

$$s_t + \sum_{j=1}^{k} d_t(G_{n-k+j}) = \sum_{i=1}^{n} d_t(G_i) = (n-1-t)d_t + (t+1)d_{t+1}.$$

and $\varepsilon_t = \sum_{j=1}^{k} d_t(G_{n-k+j})$. For any $i \in \{1, \ldots, n\}$, a vertex of degree t in G_i actually has degree t or $t+1$ in G and so $0 \le d_t(G_i) \le d_t + d_{t+1}$. The result now follows. □

As noted by Brown and Fenner [8] among others, any result for a graph G implies a corresponding result for its complement $\overline{G}$.

Observation 6.5. *If $\mathcal{D}(G) = \{G_1, \ldots, G_n\}$, then $\mathcal{D}(\overline{G}) = \{\overline{G}_1, \ldots, \overline{G}_n\}$. Moreover, we have that $d_t(\overline{G}) = d_{n-1-t}(G)$ for any $t \in \{0, \ldots, n-1\}$.*

Definition. For $t < \frac{n}{2}$, set $d_t^*(G) = \max\{d_t(G_i) : 1 \le i \le n-k\}$. Now, for every $t \in \{0, \ldots, n-1\}$, define

$$d_t^* = \begin{cases} d_t^*(G) & \text{if } t < \frac{n}{2} \\ d_{n-1-t}^*(\overline{G}) & \text{if } t \ge \frac{n}{2}. \end{cases}$$

Note that, if $t \ge \frac{n}{2}$, then $n-1-t < \frac{n}{2}$ and so d_t^* is well-defined. Furthermore, d_t^* can be calculated directly from the cards we have been given.

Lemma 6.6. *Let $k \le \frac{n}{3}$. Then, for every $t \in \{0, \ldots, n-1\}$, the value d_t^* satisfies $\frac{1}{4}d_t - 1 \le d_t^* \le d_{t-1} + d_t + d_{t+1}$.*

Proof. First we consider the case where $t < \frac{n}{2}$. For these values of t, recall that $d_t^* = \max\{d_t(G_i) : 1 \le i \le n-k\}$.

Fix $t < \frac{n}{2}$ and choose $j \in \{1, \ldots, n-k\}$ such that $d_t(G_j) = d_t^*$. Every vertex of degree t on G_j corresponds to a vertex in G with degree t or $t+1$. Hence, $d_t^* \le d_t + d_{t+1}$.

Let N be the number of times a vertex of degree t in G is seen as a vertex of degree $t-1$ on the cards $G_1, \ldots, G_{n-k}$. We will find upper and lower bounds for N. For the upper bound, note that a vertex of degree t appears as a vertex of degree $t-1$ on the card $G_i = G - v_i$ if and only if v_i is one of its neighbours. Therefore, $N \le td_t$.

Now consider the card G_i for some $i \in \{1, \ldots, n-k\}$. The card G_i has $d_t(G_i)$ vertices of degree t and each of these vertices has degree t or $t+1$ in G. The vertex v_i is not present on the card G_i but may also have degree t. Therefore, there are at least $d_t - d_t(G_i) - 1$ vertices that have degree $t-1$ in G_i but degree t in G. It follows that $N \ge \sum_{i=1}^{n-k}(d_t - d_t(G_i) - 1)$. We combine these bounds on N to get

$$td_t \ge N \ge \sum_{i=1}^{n-k} (d_t - d_t(G_i) - 1) \ge (n-k)(d_t - d_t^* - 1).$$

Rearranging and using the assumptions that $t < \frac{n}{2}$ and $n - k \ge \frac{2n}{3}$, we find that $\frac{2}{3}d_t^* \ge \frac{1}{6}d_t - \frac{2}{3}$. It follows that $d_t^* \ge \frac{1}{4}d_t - 1$.

We next consider the case where $t \ge \frac{n}{2}$ (and so $n-1-t < \frac{n}{2}$). For these values of t, recall that $d_t^* = d_{n-1-t}^*(\overline{G})$. From the argument above, we have

$$\frac{1}{4}d_{n-1-t}(\overline{G}) - 1 \le d_{n-1-t}^*(\overline{G}) \le d_{n-1-t}(\overline{G}) + d_{n-t}(\overline{G}).$$

By Observation 6.5, we see that

$$\frac{1}{4}d_t(G) - 1 \le d_{n-1-t}^*(\overline{G}) = d_t^* \le d_t(G) + d_{t-1}(G).$$

As d_{t-1} and d_{t+1} are both non-negative for every value of t, the result follows. $\square$

In the proof of Theorem 6.1, we will compare the unknown sequence (d_t) to a sequence $(\widetilde{d_t})$ that can be calculated from the cards we have been given. In order to do this, we will need to know some values of d_t exactly.

For the proof of Theorem 6.1, we will only need the following lemma in the case when $\beta = \frac{1}{2}$ and in the interval $[\frac{n}{3}, \frac{2n}{3}]$. However, the result may be of independent interest and so we state it in a more general form.

Lemma 6.7. *Suppose $0 \leq \beta < 1$ and write $\gamma = \frac{3}{4} + \frac{1}{4}\beta < 1$. Let n be sufficiently large and $k = O(n^{\beta})$. Then, for any graph G of order n and any deck of $n-k$ cards, the value of d_t can be calculated exactly for all but $O(n^{\gamma})$ values of t.*

Proof. We will assume that n is large enough to ensure $k \leq \frac{n}{3}$. Let $I = \{0, \ldots, n-1\}$ and $A = \{t \in I : d_t^* + 1 \geq \frac{1}{4}K\}$ where $K = n^{1-\gamma}$. By Lemma 6.6, we have $\sum_{t\in A}(d_t^* + 1) \leq \sum_{t \in A}(d_{t-1} + d_t + d_{t+1} + 1) \leq 4n$. Hence, $|A| \leq \frac{16n}{K} = 16n^{\gamma}$. Note that, if $t \notin A$, then $d_t \leq K$ by Lemma 6.6 and so, for n sufficiently large, d_t is small in comparison to n. Let $I' = I \setminus (A \cup (A-1)) = \{t \in I : t \notin A \text{ and } t+1 \notin A\}$. We will show that we can calculate d_t exactly for most values of $t \in I'$.

If $t \in I'$, then $t, t+1 \notin A$ and hence $d_t, d_{t+1} \leq K$. By Lemma 6.4, we have $s_t = (n-1-t)d_t + (t+1)d_{t+1} - \varepsilon_t$ where $0 \leq \varepsilon_t \leq k(d_t + d_{t+1}) \leq 2kK$. For convenience, we will write $t+1 = qn$ where $q = \frac{t+1}{n} \in [0,1] \cap \mathbb{Q}$. Then $\frac{s_t}{n} = (1-q)d_t + qd_{t+1} - \frac{\varepsilon_t}{n}$. Note that we are able to find the value $\frac{s_t}{n}$ from the cards $G_1, \ldots, G_{n-k}$.

Consider the set $X = \{(1-q)a + qb : a, b \in \{0, \ldots \lfloor K \rfloor - 1\}\}$. Choose a, b such that $(1-q)a + qb$ is the closest element of this set to $\frac{s_t}{n}$. We would like to conclude that $d_t = a$ and $d_{t+1} = b$. This is true if every pair of elements of X takes values that are at least $\left|\frac{2\varepsilon_t}{n}\right|$ apart.

Suppose that, for some $\delta < \left|\frac{2\varepsilon_t}{n}\right|$, we are able to find integers $a > a'$ and $b < b'$ satisfying $a(1-q) + bq = a'(1-q) + b'q + \delta$. Rearranging, we get

$$a - a' = (b' - b + a - a')q + \delta.$$

In particular, $(b' - b + a - a')q + \delta$ is an integer. As $|b' - b + a - a'| \leq 2\lfloor K \rfloor - 2$, it suffices to ensure that q is at distance at least δ from every element of

$$R = \left\{ \frac{x}{y} : y \in \{1, \ldots, 2\lfloor K \rfloor - 2\}, x \in \{0, \ldots, y\} \right\}.$$

The set R has size strictly less that $2K^2$. For $i \in \{0, \ldots, n-1\}$ and $\frac{x}{y} \in R$, suppose that $\left|\frac{x}{y} - \frac{i}{n}\right| < \left|\frac{2\varepsilon_t}{n}\right|$. Then $\left|\frac{xn}{y} - i\right| < 2|\varepsilon_t|$. It is straightforward to see that, for each choice of $\frac{x}{y} \in R$, there are at most $4|\varepsilon_t|$ choices for i. Define

$$S = \left\{ t : \exists r \in R \text{ such that } \left|\frac{t+1}{n} - r\right| < \left|\frac{2\varepsilon_t}{n}\right| \right\}.$$

We see that $|S| \leq 4|\varepsilon_t||R| < 16kK^3$. If we let $J = I' \setminus S$, then it follows that $|J| \geq n - \frac{32n}{K} - 16kK^3 > n - 48n^\gamma$ for n sufficiently large. For every $t \in J$, we can calculate d_t exactly. ☐

6.2.3 Main Result

We are now ready to prove that, for n sufficiently large and $k \leq \frac{1}{20}\sqrt{n}$, the size of a graph of order n is reconstructible from $n-k$ cards.

Proof of Theorem 6.1. As noted earlier, we can obtain an estimate $\widetilde{m}$ for the number of edges in G from the cards $G_1, \ldots, G_{n-k}$. By Lemma 6.3, we have $\widetilde{m} = m + \alpha$ where $0 \leq \alpha \leq \left\lfloor \frac{k(n-1)}{n-2-k} \right\rfloor$ and so it suffices to find α. For n sufficiently large, we have $n - 1 < 2(n-2-k)$ and hence $\alpha < 2k$.

Throughout the remainder of this proof, we will say that d_t is *big* if $d_t > \sqrt{n}$ and *little* if $d_t \leq \frac{3}{4}\sqrt{n}$.

Claim 6.8. *Suppose that, for some $t \leq \frac{2n}{3} - 1$, the value of d_{t+1} is known exactly and is not big. Then either d_t can be calculated exactly or d_t can be identified as being big.*

Proof. From Lemma 6.4, we know that $s_t = (n-1-t)d_t + (t+1)d_{t+1} - \varepsilon_t$ where

$0 \leq \varepsilon_t \leq k(d_t + d_{t+1})$. Define $d'_t = \frac{1}{n-1-t}(s_t - (t+1)d_{t+1})$. Then d'_t can be calculated exactly and, furthermore, $d_t = d'_t + \frac{\varepsilon_t}{n-1-t}$ so $d_t \geq d'_t$.

Suppose that $d_t \leq 2\sqrt{n}$. By assumption, $d_{t+1} \leq \sqrt{n}$ and so it follows that $d_t - d'_t = \frac{\varepsilon_t}{n-t-1} \leq \frac{3}{n}k(d_t + d_{t+1}) \leq \frac{9}{20} < \frac{1}{2}$. Hence, if $d_t \leq 2\sqrt{n}$, the closest integer to d'_t is precisely d_t. In particular, this holds if d_t is not big.

Now suppose that $d_t > 2\sqrt{n}$. It suffices to show that the closest integer to d'_t must be big. We know $d'_t = d_t - \frac{\varepsilon_t}{n-1-t}$, $t+1 \leq \frac{2n}{3}$ and $d_{t+1} \leq \sqrt{n}$. Together, these give us that $\frac{\varepsilon_t}{n-1-t} \leq \frac{3}{n}k(d_t + d_{t+1}) \leq \frac{3d_t}{20\sqrt{n}} + \frac{3}{20} \leq \frac{1}{2}d_t$ for n sufficiently large and hence $d'_t \geq \frac{1}{2}d_t > \sqrt{n}$. We therefore correctly identify that d_t is big in this case. $\Diamond$

Claim 6.9. *Suppose that, for some $t \geq \frac{n}{3}+1$, the value of d_{t-1} is known exactly and is not big. Then either d_t can be calculated exactly or d_t can be identified as being big.*

Proof. If $t \geq \frac{n}{3}+1$, then $n-t-1 \leq \frac{2n}{3}-1$. Observation 6.5 tells us $d_{n-t}(\overline{G}) = d_{t-1}(G)$. By applying Claim 6.8 to $\overline{G}$, we see that either $d_t(G) = d_{n-t-1}(\overline{G})$ can be calculated exactly or it can be identified as being big. $\Diamond$

Claim 6.10. *The interval $\left[\frac{n}{3}, \frac{2n}{3}\right]$ contains $2k$ consecutive values of t such that every d_t can be calculated exactly and they are all little.*

Proof. Let $I = \left[\frac{n}{3}, \frac{2n}{3}\right] \cap \mathbb{N}$. Lemma 6.7 with $\beta = \frac{1}{2}$ gives a set $J \subseteq I$ and a constant c such that $|I \setminus J| \leq cn^{\frac{7}{8}}$ and we can calculate d_t exactly if $t \in J$. Hence there are at most $cn^{\frac{7}{8}}$ values of $t \in I$ for which we cannot calculate d_t exactly.

Partition I into $\left\lfloor \frac{n}{6k} \right\rfloor$ intervals of length $2k$. At most $\left\lfloor \frac{cn^{7/8}}{2k} \right\rfloor$ of them are completely contained in $I \setminus J$. For n sufficiently large, $\left\lfloor \frac{n}{6k} \right\rfloor - \left\lfloor \frac{cn^{7/8}}{2k} \right\rfloor \geq \frac{n}{8k}$. Therefore, for these sufficiently large values of n, there are at least $\frac{n}{8k}$ intervals which have non-empty intersection with J. By Claims 6.8 and 6.9, we are able to calculate d_t exactly for all values of t in each of these intervals unless the interval happens to contain a value of t for which d_t is big.

We know that there are at most $\frac{4}{3}\sqrt{n}$ values of $t \in \{0,\ldots,n-1\}$ for which d_t is not little. Therefore, as $\frac{n}{8k} \geq \frac{5}{2}\sqrt{n} > \frac{4}{3}\sqrt{n}$, there exists an interval which has non-empty intersection with J and which only contains little values of d_t, each of which we can calculate exactly. $\Diamond$

By Claim 6.10, we can find an interval $\mathcal{I} = \{b, b+1, \ldots, b+2k-1\} \subset \left[\frac{n}{3}, \frac{2n}{3}\right]$ such that, for every $t \in \mathcal{I}$, we can calculate d_t exactly and it is little. We may then recursively apply Claim 6.8, starting with $t+1 = b$. We continue until either we reach d_0 or we hit a big vertex d_{t_ℓ} for some $t_\ell < b$. Similarly, we may recursively apply Claim 6.9, starting with $t-1 = b+2k-1$. Again, we will either calculate d_{n-1} or we will identify that d_{t_r} is big for some $t_r > b+2k-1$.

If we are able to calculate both d_0 and d_{n-1}, then we will know d_t for every $t \in \{0,\ldots,n-1\}$. This tells us the degree sequence of G and hence we can directly calculate m.

Therefore, we may assume that we have the following situation: there exists an interval $\mathcal{J} \supseteq \mathcal{I}$ with endpoints t_ℓ and t_r such that $t_\ell < t_r$. For every $t \in \mathcal{J} \setminus \{t_\ell, t_r\}$, the value d_t is known exactly and is not big. At least one of d_{t_ℓ} and d_{t_r} has been identified as being big. By Observation 6.5, we may assume that d_{t_ℓ} is big.

Recall from the proof overview that

$$\widetilde{d}_t = \left|\left\{i \in \{1,\ldots,n-k\} : \widetilde{d}(v_i) = t\right\}\right|$$

can be calculated from the cards and also that $\sum_{t=0}^{n-1} |d_t - \widetilde{d}_{t+\alpha}| = k$. (Note that we need to calculate $\widetilde{d}_t$ for $0 \leq t \leq n+2k$ and that, for $t+\alpha \geq n$, it is possible for $\widetilde{d}_{t+\alpha}$ to take a non-zero value.) This means that the overall shape of $\widetilde{d}_0,\ldots,\widetilde{d}_{n-1}$ will be the same as the overall of shape of $d_0,\ldots,d_{n-1}$ but shifted to the right by α. We will determine α.

Although we do not know the exact value of d_{t_ℓ}, it is sufficient to redefine each d_t

and $\widetilde{d}_t$ to be the minimum of their current value and $\sqrt{n}$. After doing this, we still have $\sum_{t=0}^{n-1} |d_t - \widetilde{d}_{t+\alpha}| \leq k$. It follows that $\sum_{t=t_\ell}^{t_r-1} |d_t - \widetilde{d}_{t+\alpha}| \leq k$.

Claim 6.11. *For $s \in \{0, \ldots, 2k-1\}$, $\sum_{t=t_\ell}^{t_r-1} |d_t - \widetilde{d}_{t+s}| \leq k$ if and only if $s = \alpha$.*

Proof. Fix $s \in \{0, \ldots, 2k-1\}$. We noted above that $\sum_{t=t_\ell}^{t_r-1} |d_t - \widetilde{d}_{t+\alpha}| \leq k$. Now suppose $s \neq \alpha$. We have

$$\begin{aligned}\sum_{t=t_\ell}^{t_r-1} |d_t - \widetilde{d}_{t+s}| &= \sum_{t=t_\ell}^{t_r-1} |d_t - d_{t+s-\alpha} + d_{t+s-\alpha} - \widetilde{d}_{t+s}| \\ &\geq \sum_{t=t_\ell}^{t_r-1} |d_t - d_{t+s-\alpha}| - \sum_{t=t_\ell}^{t_r-1} |d_{t+s-\alpha} - \widetilde{d}_{t+s}|. \qquad (\star)\end{aligned}$$

Since $\sum_{t=0}^{n-1} |d_t - \widetilde{d}_{t+\alpha}| \leq k$, it follows that

$$\sum_{t=t_\ell}^{t_r-1} |d_{t+s-\alpha} - \widetilde{d}_{t+s}| = \sum_{t=t_\ell+s-\alpha}^{t_r+s-\alpha-1} |d_t - \widetilde{d}_{t+\alpha}| \leq k.$$

Hence, $(\star)$ will be strictly greater than k whenever $\sum_{t=t_\ell}^{t_r-1} |d_t - d_{t+s-\alpha}| > 2k$.

Recall that the interval $\mathcal{I}$ consists of $2k$ consecutive values of t such that every d_t is little. As $s \leq 2k-1$ and $s \neq \alpha$, then there exists some $\eta \in \mathbb{Z}$ such that $t_\ell + \eta(s-\alpha) \in \mathcal{I}$. We know that $\eta(s-\alpha) > 0$. First assume $\eta > 0$. Since d_{t_ℓ} is big and $d_{t_\ell+\eta(s-\alpha)}$ is little, we find

$$\begin{aligned}\sum_{t=t_\ell}^{t_r-1} |d_t - d_{t+s-\alpha}| &\geq \sum_{i=0}^{\eta-1} |d_{t_\ell+i(s-\alpha)} - d_{t_\ell+(i+1)(s-\alpha)}| \\ &\geq |d_{t_\ell} - d_{t_\ell+\eta(s-\alpha)}| \\ &\geq \sqrt{n} - \frac{3}{4}\sqrt{n} = \frac{1}{4}\sqrt{n} \\ &> 2k.\end{aligned}$$

If $\eta < 0$, then $s - \alpha < 0$ and

$$\sum_{t=t_\ell}^{t_r-1} |d_t - d_{t+s-\alpha}| \geq \sum_{i=0}^{-\eta} |d_{t_\ell+(i+1)(\alpha-s)} - d_{t_\ell+i(\alpha-s)}| \geq |d_{t_\ell-\eta(\alpha-s)} - d_{t_\ell}|.$$

The result then follows in a similar fashion. $\Diamond$

By Claim 6.11, we see that α is the only value $s \in \{0, \ldots, 2k-1\}$ which satisfies $\sum_{t=t_\ell}^{t_r-1} |d_t - \widetilde{d}_{t+s}| \leq k$. Therefore, by calculating $\sum_{t=t_\ell}^{t_r-1} |d_t - \widetilde{d}_{t+s}|$ for each $s \in \{0, \ldots, 2k-1\}$, we can identify α. Once we know α, we can then calculate m, the number of edges in G. $\square$

It is natural to ask how large n needs to be for Theorem 6.1 to hold. Although in the proof there are several places where we require n to be sufficiently large for a statement to hold, it is a statement in Claim 6.10 which determines how large n needs to be. In particular, we require

$$\left\lfloor \frac{n}{6k} \right\rfloor - \left\lceil \frac{cn^{7/8}}{2k} \right\rceil \geq \frac{n}{8k}$$

to hold where c is the constant found in Lemma 6.7. It follows that n must be at least 10^{22}.

We now consider the situation where our deck contains $n-k$ true cards and k false cards and show that we can calculate the number of edges in our graph exactly if $k \leq \frac{1}{40}\sqrt{n}$.

Proof of Corollary 6.2. Suppose that G and H are two graphs on n vertices and each has at least $n-k$ cards in common with a deck of cards $\mathcal{C}$. Then G and H must have at least $n-2k$ cards in common. We may apply Theorem 6.1 to these $n-2k$ common cards. If n is sufficiently large and $2k \leq \frac{1}{20}\sqrt{n}$, then G and H must have the same number of edges. $\square$

6.3 Conclusion

We have shown that the size of a graph can be reconstructed if we are given a deck from which either at most $\frac{1}{20}\sqrt{n}$ cards are missing or at most $\frac{1}{40}\sqrt{n}$ cards are false. The constants can be improved a bit, although we do not know whether the result remains true with $\sqrt{n}$ missing cards. However, we suspect that stronger results could be proved by using more information about the degree sequences on the cards.

We also note that $c\sqrt{n}$ is still very far away from the best known lower bounds, which are linear. For example, for $n = 3p + 1$, Bowler, Brown and Fenner [6] have given the following two graphs which differ in the number of edges but have $\frac{2}{3}(n-1)$ cards in common: the graphs $G = 2K_{p+1} + K_{p-1}$ and $H = K_{p+1} + 2K_p$ both have at least $2p$ cards of the form $K_{p+1} + K_p + K_{p-1}$. We suspect that the lower bound is closer to the truth and propose the following question.

Question 6.12. *Does there exist some $\varepsilon > 0$ such that, for any graph G on n vertices, we can reconstruct the number of edges of G from any $(1 - \varepsilon)n$ cards?*

Another direction for future work is to reconstruct other graph parameters, such as the degree sequence or the number of triangles. The following question seems very natural.

Question 6.13. *Fix $k \in \mathbb{N}$ and a graph H and let n be sufficiently large. Can we reconstruct the number of copies of H in G given any $n - k$ cards from $\mathcal{D}(G)$ for any graph G on n vertices?*

If we are given the entire deck $\mathcal{D}(G)$ (i.e. $k = 0$), then this question is answered by Kelly's Lemma [37].

Lemma 6.14. *For any two graphs G and H with $|G| > |H|$, the number of subgraphs of G isomorphic to H is reconstructible.*

If the number of edges is known, then the degree of a vertex can be calculated from the number of edges on its card. Therefore, by our main result, if $k \leq \frac{1}{20}\sqrt{n}$, then all but k of the degrees are known. If k is larger, then Lemma 6.7 still allows us to construct most of the degree sequence. We expect that, for a large range of k, it is possible to determine the whole degree sequence exactly. As a first step, we make the following conjecture.

Conjecture 6.15. *Fix $k \in \mathbb{N}$ and let n be sufficiently large. For any graph G on n vertices, the degree sequence of G is reconstructible from any $n-k$ cards.*

Note that a positive answer to Question 6.13 would give a positive answer to Conjecture 6.15: for fixed k and n sufficiently large, we can find the number of edges of the graph by Theorem 6.1 and hence determine all but k elements of the degree sequence. Provided n is sufficiently large, we can reconstruct the number of copies of the star $K_{1,j}$ for $j = 1, \ldots, k+1$; this is given by $\sum_{v \in V(G)} \binom{d(v)}{j}$. By subtracting the terms corresponding to vertices of known degree, we obtain a sequence of polynomials in the unknown degrees. Adding constants, these form a basis for all polynomials of degree at most $k+1$. From these, it is straightforward to evaluate the remaining degrees.

Chapter 7

Approximating the Position of a Hidden Agent in a Graph

7.1 Introduction

A *pursuit and evasion game* is a game where one player tries to localise a second player moving along the edges of a graph by using information about the current position of the second player.

In this chapter, we will study a variation of the Cat and Mouse pursuit and evasion game that was introduced by Dayanikli and Rautenbach [13]. The cat is trying to find the mouse up to a given distance in finite time. A time step consists of one turn for each player with the mouse playing first. On its turn, the mouse must either stay where it is or move to a neighbouring vertex. The mouse knows the strategy used by the cat and can use this information when choosing which vertex to move to. The cat is not constrained by the edges of G and may test any vertex to see if the mouse is there. At the end of each time step, the cat is told whether the distance between itself and the mouse has increased compared with the previous time step. The cat wins if it is able to localise the mouse up to a given distance d within finite time (i.e. the cat can determine a vertex v such that it knows the mouse is at distance at most d from v). Otherwise, the mouse wins. A more formal description of the game, including a precise definition of what it means for the cat to localise the mouse, is given in Section 7.2.

Dayanikli and Rautenbach [13] proved that, if G is a tree with maximum degree Δ and radius h, then the cat can localise the mouse up to distance $4\Delta - 6$ within time $O(h\Delta)$. They also showed that the cat can localise the mouse up to distance 8 within time $O(\log n)$ on the $n \times n$ grid. These results led them to make the following conjecture.

Conjecture 7.1. *The cat can localise the mouse up to distance $O(\log n)$ on any connected graph of order n.*

In Section 7.3, we will show that Conjecture 7.1 is false. We will prove the following theorem.

Theorem 7.2. *For all n sufficiently large, there exists a tree T on n vertices such that, regardless of the strategy employed by the cat, the mouse is able to avoid being localised up to distance $\frac{1}{12}\sqrt{n}$ on T for an infinite period of time.*

In Section 7.4, we will prove that the cat can localise the mouse up to distance $O(\sqrt{n})$ on any simple, undirected and connected graph of order n within time $O(\sqrt{n})$. In particular, we will prove the following theorem.

Theorem 7.3. *The cat can localise the mouse up to distance $\sqrt{32n}$ by time $\sqrt{2n}$.*

Together, these two results show that the cat can localise the mouse up to distance $O(\sqrt{n})$ on any simple, undirected and connected graph of order n and that this bound is tight up to a constant factor.

7.2 Rules of the Game

There are two players, the *cat* and the *mouse*, and the game is played on a simple, undirected, connected graph G of order n. It proceeds in discrete time steps, which are labelled by the positive integers.

The mouse is only able to move along the edges of G. Let m_i be the position of the mouse at time i. For $i \geq 2$, m_i belongs to the closed neighbourhood of m_{i-1}, that is m_i is either m_{i-1} or a neighbour of m_{i-1}.

The cat may test any vertex of G without any restrictions. At time i, the cat chooses a vertex c_i. If $i \geq 2$, at the end of time step i, the cat is told the value of b_i where

$$b_i = \begin{cases} 1 & \text{if } d(c_i, m_i) \leq d(c_{i-1}, m_{i-1}) \\ 0 & \text{if } d(c_i, m_i) > d(c_{i-1}, m_{i-1}) \end{cases}$$

and $d(u, v)$ denotes the length of a shortest path connecting vertices u and v. The value of b_i tells the cat whether it is further away from the mouse at step i than it was at step $i - 1$. The cat has no information about the location of the mouse when it chooses c_1 and c_2. However, for $i \geq 3$, the cat can use $b_2, \ldots, b_{i-1}$ to help it decide which vertex to choose for c_i. For each i, the cat can calculate M_i, the set of possible positions for m_i. A vertex v belongs to M_i if and only if there are vertices $\widetilde{m}_1, \ldots, \widetilde{m}_i$ satisfying:

- $v = \widetilde{m}_i$
- $\widetilde{m}_j$ is in the closed neighbourhood of $\widetilde{m}_{j-1}$ for every $2 \leq j \leq i$
- for every $2 \leq j \leq i$, $d(c_j, \widetilde{m}_j) \leq d(c_{j-1}, \widetilde{m}_{j-1})$ if and only if $b_j = 1$.

Let $f : \bigcup_{i \in \mathbb{N}} \{0, 1\}^i \to V(G)$. We say that the cat *follows strategy* (c_1, c_2, f) if c_1 and c_2 are the first two vertices it tests and $c_i = f(b_2, \ldots, b_{i-1})$ for $i \geq 3$. The *radius* of a set W of vertices is defined as

$$\mathrm{rad}_G(W) = \min \big\{ \max\{d(v,w) : w \in W\} : v \in V(G) \big\}.$$

We say that *the cat can localise the mouse up to distance d within time t on G* if there is some strategy (c_1, c_2, f) such that, whenever the cat follows this strategy (regardless of the movements of the mouse), there exists $\tau \leq t$ with $\mathrm{rad}_G(M_\tau) \leq d$. Note that, at time τ, the cat can identify a vertex $v \in M_\tau$ such that it knows the mouse is at most distance d from v.

The cat only knows G and the values b_i (and hence can calculate the sets M_i). The mouse has more information. As well as G, the mouse also knows about the strategy followed by the cat. Therefore, the mouse knows how the cat will choose its vertices and can adapt its own strategy accordingly.

Note that the cat must follow a strategy $f : \bigcup_{i \in \mathbb{N}} \{0,1\}^i \to V(G)$ which has been decided before the game begins. We do not allow the cat to follow a random strategy (as we require the cat to be certain of localising the mouse within the given time and distance).

7.3 Lower Bound

Before we prove Theorem 7.2 for every $n \in \mathbb{N}$, we will first show that it holds for some special cases. We will prove the following proposition.

Proposition 7.4. *Let t be a multiple of 12 and take $n = t^2 + 1$. There exists a tree T on n vertices such that, regardless of the strategy employed by the cat, the mouse is able to avoid being localised up to distance $\frac{t}{12}$ on T for an infinite period of time.*

Throughout the rest of this section, fix $n = t^2 + 1$ where t is divisible by 12. We define T to be the tree on n vertices formed from the star $K_{1,t}$ by subdividing each edge exactly $t-1$ times. We will call the centre vertex u and refer to the subdivided edges as branches. We do not regard u to be on any branch. The structure of T means (informally) that, at each turn, the cat is only able to check whether the mouse lies on a single branch of the graph.

In the proof of Proposition 7.4, we will provide a strategy for the mouse where m_i is always on a different branch to c_{i-1} and c_i and hence the cat can never determine from its tests which branch the mouse is on. We will also consider a second set of possible positions (w_i) for the mouse such that the cat cannot tell the difference between a mouse at m_i and a mouse at w_i. For every i, we will ensure that $d(m_i, w_i) > \frac{t}{6}$ and hence $\text{rad}_G(M_i) > \frac{t}{12}$. This is sufficient to ensure that cat cannot localise the mouse up to distance $\frac{t}{12}$ in finite time.

Proof of Proposition 7.4. Suppose that the cat is trying to localise the mouse to up distance $\frac{t}{12}$ on T. We will provide a strategy for the mouse such that $\text{rad}_G(M_i) > \frac{t}{12}$ for every $i \in \mathbb{N}$. Hence, the cat is never able to localise the mouse on T.

To simplify the numbers, the first turn is at time 0. At this time, the cat only knows the graph T. The mouse knows what T looks like as well as the strategy employed by the cat. The mouse will sometimes give the cat additional information about its position or movements. We will see later that this does not impact the result.

The mouse employs the following strategy, making the moves given by (m_i). In what follows, we will also consider a second set of possible moves (w_i) for the mouse. At each turn, we will ensure that both $m_i, w_i \in M_i$ and $d(m_i, w_i) > \frac{t}{6}$. Hence $\text{rad}_G(M_i) > \frac{t}{12}$ as required.

At several stages in the mouse's strategy, it needs to choose a branch which the cat will not check for a given period of time τ where $\tau < t$. As T has t branches and the cat can only check one per turn, such a branch certainly exists. In order to find such a branch, it is necessary to look ahead at the cat's strategy. Note that, as long as the cat does not play on the branch within the given time period, it does not matter which branch the mouse chooses because its movements, and consequently the cat's movements, will be the same.

Here is the strategy which the mouse will follow:

1. The mouse chooses a branch which the cat will not check for $\frac{2t}{3}$ turns and chooses m_0 to be the vertex at distance $\frac{t}{4}$ from u on this branch. Similarly, let w_0 be a vertex at distance $\frac{t}{4}$ from u on another branch that the cat will not check for $\frac{2t}{3}$ turns. The mouse tells the cat that it is at distance $\frac{t}{4}$ from u so $M_0 = \{v \in V(T) : d(v, u) = \frac{t}{4}\}$. The mouse also tells the cat $\frac{t}{4}$ branches on which it is not located (none of which are the branches with m_0 or w_0).

2. Over the next $\frac{t}{6}$ turns, the cat will gradually lose information about the position of the mouse. For $1 \leq i \leq \frac{t}{6}$, define m_i and w_i as follows:

   - If $d(c_i, u) \leq d(c_{i-1}, u)$, then m_i is one vertex closer to u than m_{i-1} and $w_i = w_{i-1}$.
   - If $d(c_i, u) > d(c_{i-1}, u)$, then $m_i = m_{i-1}$ and w_i is one vertex further away from u than w_{i-1}.

   Note that, in the first case, we have $b_i = 1$ and, in the second case, we have $b_i = 0$. In both cases, the cat receives the same value of b_i for a mouse at m_{i-1} and then m_i and a mouse at w_{i-1} followed by w_i and so cannot tell the difference between these positions. At time $\frac{t}{6}$, we have $\frac{t}{3} \leq d(m_{t/6}, w_{t/6}) \leq \frac{2t}{3}$.

3. Let $d = d(m_{t/6}, u)$ so $\frac{t}{12} \leq d \leq \frac{t}{4}$. The mouse tells the cat that it will now run towards u for d time steps and proceeds to do so. For $\frac{t}{6} < i \leq \frac{t}{6} + d$, we have m_i and w_i are both one vertex closer to u than m_{i-1} and w_{i-1} respectively.

4. At time $\frac{t}{6}+d$, we have $m_{t/6+d} = u$ and $d(w_{t/6+d}, u) = \frac{t}{6}$. The mouse now chooses a branch on which the cat will not play in the next $\frac{11t}{12}$ turns (and different from

the branch containing w_i). This is the branch on which the (m_i) will now lie. The (w_i) will remain on their original branch.

5. The mouse now tells the cat it is running away from the centre for $\frac{t}{4}$ time steps and does so. For $\frac{t}{6} + d < i \leq \frac{5t}{12} + d$, both m_i and w_i are one vertex further away from u than m_{i-1} and w_{i-1} respectively.

6. Let $j = \frac{5t}{12} + d$. At time j, we have $d(m_j, u) = \frac{t}{4}$. Let B be a branch which the cat has not checked in the previous $\frac{t}{4}$ turns and will not check for the next $\frac{2t}{3}$ turns, and which is different to the branch the mouse is currently on (i.e. $m_j \notin B$). Redefine w_j to be the vertex at distance $\frac{t}{4}$ from u on B. The mouse then tells the cat it is at distance $\frac{t}{4}$ from u. The situation is now exactly the same as in step 1.

Stage 6 is identical to stage 1, both in the positions of the mouse and the information known by the cat. Thus, the mouse may repeat stages 2 through 6 to avoid being localised by the cat indefinitely.

Let us now justify the mouse's choice of branch in Step 1. Provided that the mouse chooses a branch that the cat will not play on in the next $\frac{2t}{3}$ rounds, the sequence (b_i) and the branches the cat plays on do not depend on which branch the mouse is on. Therefore, as the mouse knows the cat's strategy, the mouse can look into the future, simulate how the cat will play and hence find a suitable branch.

At stage 4, we have $m_{t/6+d} = u$. The cat has no idea which branch the mouse will run down and so all of the branches it previously eliminated now need checking again. However, the cat cannot tell the difference between a mouse at $m_{t/6+d}$ and one at $w_{t/6+d}$ and so it does not know that the mouse is at u. As $d(w_{t/6+d}, u) = \frac{t}{6}$ but the cat does not know which branch $w_{t/6+d}$ is on, we still have $\mathrm{rad}_G(M_{t/6+d}) > \frac{t}{12}$.

Between times $\frac{t}{6} + d$ and j, the cat has eliminated at most $\frac{t}{4}$ branches for a mouse that reached the centre at time $\frac{t}{6} + d$. At time j, the cat (depending on its play for the previous $\frac{t}{4}$ turns) may realise that the mouse was indeed at the centre at time $\frac{t}{4}+d$, but this is no longer helpful as the mouse is now at distance $\frac{t}{4}$ from the centre along an unknown branch.

Now consider the situation where the cat is not given any information except for T and the values of the b_i for $i \geq 2$. The mouse can employ exactly the same strategy as it used above but without telling the cat anything. Let M_i' be the possible positions of the mouse at time i and M_i be the sets given by the above argument where the cat has extra information. This additional information enables the cat to narrow down the mouse's possible position more accurately and so, for every i, we find that $M_i \subseteq M_i'$

Therefore, as the cat cannot localise the mouse up to distance $\frac{t}{4}$ when it is told some information about its position and movements, it definitely cannot localise the mouse when it does not have access to this extra knowledge. $\square$

Theorem 7.2 follows as a corollary of Proposition 7.4.

Proof of Theorem 7.2. Suppose $n = t^2+r$ where t is divisible by 12 and $r \in [24t+144]$. Let T be the tree described in Proposition 7.4 – the tree obtained by subdividing each edge of the star $K_{1,t}$ exactly $t-1$ times. Create the tree T' on n vertices by adding a branch B of $r-1$ vertices to T. The cat and mouse play on T'. The mouse only plays on the subgraph T and is never located on B. It uses the same strategy as in Proposition 7.4 to avoid being localised up to distance $\frac{t}{12}$ on T'. $\square$

7.4 Upper Bound

In this section, we will prove Theorem 7.3. In fact we prove two slightly more precise results.

Theorem 7.5. *Let G be a connected graph on n vertices and let $c > 0$. Then the cat can localise the mouse up to distance $\left(\frac{8}{c} + c\right)\sqrt{n}$ by time $\frac{4}{c}\sqrt{n}$.*

Theorem 7.6. *Let G be a connected graph on n vertices. The cat can localise the mouse up to distance $\frac{9}{2}\sqrt{n}$ by time $\frac{n}{2}$.*

Theorem 7.3 follows from Theorem 7.5 by setting $c = \sqrt{8}$. Theorems 7.5 and 7.6 will follow from two lemmas giving upper bounds on how well a cat can localise a mouse in a graph.

For any $k \in \mathbb{N}$ and any $w \in V(G)$, the set $B_k(w) = \{v \in V(G) : d(w, v) \leq k\}$ is known as the k-ball around w. The first lemma depends on how easily we can cover the vertices of the graph G with balls. Thus we get stronger bounds when G is a "fat" graph with lots of clustering.

Lemma 7.7. *Let $L, k \in \mathbb{N}$ and let $G = (V, E)$ be a graph whose vertices may be covered by L balls of radius k. Then the cat can localise the mouse up to distance $4L + k$ in G by time $2L$.*

Assuming this lemma, we can prove Theorem 7.5.

Proof of Theorem 7.5. Let $c > 0$ and let $G = (V, E)$ be a connected graph. Let $\mathcal{R}$ be a maximal subset of V with pairwise distance at least $c\sqrt{n}$. Since the $\frac{c\sqrt{n}-1}{2}$-balls around vertices in $\mathcal{R}$ are disjoint and G is connected, $|\mathcal{R}| \leq \frac{2}{c}\sqrt{n}$. Thus the vertices of G are covered by $\frac{2}{c}\sqrt{n}$ balls of radius $c\sqrt{n}$. By Lemma 7.7 the cat can localise the mouse up to distance $\left(\frac{8}{c} + c\right)\sqrt{n}$ by time $\frac{4}{c}\sqrt{n}$. □

The second lemma depends on a different property of the graph G. It gives stronger bounds when G is a "thin" graph. We define the *diameter* of any connected graph G to be $\mathrm{diam}(G) = \max\{d(u, v) : u, v \in V(G)\}$.

Lemma 7.8. *Let $K \in \mathbb{N}$ and let $G = (V, E)$ be a graph such that, for all $v \in V$, there exists an $\ell = \ell(v) < K$ such that $|\{w \in V : d(v, w) = \ell\}| < \frac{\ell}{4}$. Let D be the diameter of G. Then the cat can localise the mouse up to distance $\frac{3K}{2}$ in G by time $D/2$.*

Assuming this lemma, we can prove Theorem 7.6.

Proof of Theorem 7.6. Let $G = (V, E)$ be a graph with diameter D. For any vertex $v \in V$, there exists an $\ell < 3\sqrt{n}$ such that $|\{w \in V : d(v, w) = \ell\}| < \frac{\ell}{4}$ by the pigeonhole principle. The theorem follows by appealing to Lemma 7.8 with $K = 3\sqrt{n}$ and $D = n$. $\square$

Let $L, k \in \mathbb{N}$ and let $G = (V, E)$ be any graph. Suppose $\{u_1, \ldots, u_L\} \subseteq V$ is such that $\bigcup_{i \in [L]} B_k(u_i) = V$. Consider the strategy for the cat given by Algorithm 7.1.

Initialization *Set* $i = 1$, $w_i = 1$;

while $i < L$ **do**

  Set $c_{2i-1} = u_{w_i}$, $c_{2i} = u_{i+1}$;

  if $d(c_{2i}, m_{2i}) \leq d(c_{2i-1}, m_{2i-1})$ **then**

    Increase i and set $w_i = i$;

  else

    Increase i and set $w_i = w_{i-1}$;

  end if

end while

Algorithm 7.1: Locating the mouse in a "fat" graph.

Note that if the mouse doesn't move, then this algorithm determines which of the u_i is closest to the mouse. It takes them two at a time and chooses the closer one. When we run the same algorithm with a moving mouse, then the mouse may add noise but only at a bounded rate. Lemma 7.7 follows immediately from the following proposition.

Proposition 7.9. *Independent of* $(m_i)_{i \in \mathbb{N}}$, *we have* $d(m_{2L-1}, u_{w_L}) \leq 4L + k$.

Proof. Let $i \in [L-1]$. Then w_{i+1} is either w_i or $i+1$. In the former case, since the mouse moves along an edge once per time step,

$$\begin{aligned} d(u_{w_{i+1}}, m_{2i+1}) &= d(u_{w_i}, m_{2i+1}) \\ &\leq d(u_{w_i}, m_{2i-1}) + 2. \end{aligned}$$

Otherwise $d(u_{i+1}, m_{2i}) \leq d(u_{w_i}, m_{2i-1})$ and $w_{i+1} = i+1$ in which case

$$\begin{aligned} d(u_{w_{i+1}}, m_{2i+1}) &= d(u_{i+1}, m_{2i+1}) \\ &\leq d(u_{i+1}, m_{2i}) + 1 \\ &\leq d(u_{w_i}, m_{2i-1}) + 1. \end{aligned}$$

In either case, we have $d(u_{w_{i+1}}, m_{2i+1}) \leq d(u_{w_i}, m_{2i-1}) + 2$. By induction, it follows that $d(u_{w_L}, m_{2L-1}) \leq d(u_{w_i}, m_{2i-1}) + 2(L-i)$ for every $i \leq L$. For each $i \in \{2, \ldots, L\}$, we have $d(u_{w_i}, m_{2i-1}) \leq d(u_i, m_{2i-2}) + 1$. Therefore, for every $i \in \{2, \ldots, L\}$, we have

$$\begin{aligned} d(u_i, m_{2L-1}) &\geq d(u_i, m_{2i-2}) - 2(L-i) - 1 \\ &\geq d(u_{w_i}, m_{2i-1}) - 2(L-i) - 2 \\ &\geq d(u_{w_L}, m_{2L-1}) - 4(L-i) - 2 \\ &\geq d(u_{w_L}, m_{2L-1}) - 4L. \end{aligned}$$

Since $w_1 = 1$, the above bound also holds for $i = 1$. Rearranging this, we get that $d(u_{w_L}, m_{2i-1}) \leq 4L + d(u_i, m_{2L-1})$ for all $i \leq L$. As $(B_k(u_i))_{i \in [L]}$ forms a cover of V, we find that $d(u_i, m_{2L-1})$ must be at most k for some i and so $d(u_{w_L}, m_{2L-1}) \leq 4L + k$. $\square$

Let K be a natural number. Let $G = (V, E)$ be a graph such that, for all $v \in V$, there exists an $\ell = \ell(v) < K$ such that $|\{w \in V : d(v, w) = \ell\}| < \frac{\ell}{4}$. Suppose D is the diameter of G. For each vertex v, we fix $\ell(v)$ such that $|\{w \in V : d(v, w) = \ell\}| < \frac{\ell}{4}$.

Consider Algorithm 7.2 which dictates a strategy for the cat given the movements of the mouse (m_i). Note that T_j is not necessary for the strategy but we include it to simplify the analysis.

Initialization *Set* $i = 0$, $j = 1$, $T_1 = 0$ *and pick* $v_1 \in V$ *arbitrarily*;

while $i < D$ **do**

  Set $U = \{w \in V : d(v_j, w) = \ell(v_j)\}$;

  Set $u_j = v_j$;

  while $U \neq \emptyset$ **do**

    Increase i;

    Pick $w \in U$;

    Set $c_{2i-1} = u_j$, $c_{2i} = w$;

    if $d(c_{2i}, m_{2i}) \leq d(c_{2i-1}, m_{2i-1})$ **then**

      Set $u_j = w$;

    end if

    Set $U = U \backslash \{w\}$;

  end while

  Set $T_{j+1} = i$, $v_{j+1} = u_j$;

  Increase j

end while

Algorithm 7.2: Locating the mouse in a "thin" graph.

Lemma 7.8 follows immediately from the following proposition.

Proposition 7.10. *$d(v_n, m_{2T_n-1}) \leq \frac{3K}{2}$ for all $T_n \geq \frac{D}{2} - \frac{3K}{4}$.*

Proof. Suppose we run Algorithm 7.2.

Claim 7.11. *For every $n \geq 1$, $T_{n+1} - T_n \leq \frac{\ell(v_n)}{4}$ and*

$$d(v_{n+1}, m_{2T_{n+1}-1}) \leq \max\left\{d(v_n, m_{2T_n-1}) - \frac{\ell(v_n)}{2}, \frac{3K}{2}\right\}.$$

Proof. Let $n \geq 1$ and choose $\ell(v_n) = \ell < K$ so that $|\{w \in V : d(v_n, w) = \ell\}| < \frac{\ell}{4}$.

First, suppose that $d(v_n, m_{2T_n-1}) \geq K$. Note that there must be some w such that $d(v_n, w) = \ell$ and w is in a shortest path from v_n to m_{2T_n-1}. It follows that $d(w, m_{2T_n-1}) = d(v_n, m_{2T_n-1}) - \ell$. Let $i \geq T_n$ be such that $c_{2i} = w$. Then

$$\begin{aligned} d(c_{2i}, m_{2i}) &\leq d(w, m_{2T_n-1}) + 2i - (2T_n - 1) \\ &= d(v_n, m_{2T_n-1}) - \ell + 2i - (2T_n - 1). \end{aligned}$$

On the other hand, $d(v_{n+1}, m_{2T_{n+1}-1}) \leq d(c_{2i}, m_{2i}) + 2T_{n+1} - 1 - 2i$ and so

$$\begin{aligned} d(v_{n+1}, m_{2T_{n+1}-1}) &\leq d(v_n, m_{2T_n-1}) - \ell + 2i - (2T_n - 1) + 2T_{n+1} - 1 - 2i \\ &= d(v_n, m_{2T_n-1}) + 2(T_{n+1} - T_n) - \ell. \end{aligned}$$

As $T_{n+1} - T_n = |\{w \in V : d(v_n, w) = \ell\}| < \frac{\ell}{4}$, it follows that

$$d(v_{n+1}, m_{2T_{n+1}-1}) \leq d(v_n, m_{2T_n-1}) - \frac{\ell}{2}.$$

Now suppose $d(v_n, m_{2T_n-1}) < K$. As $T_{n+1} - T_n = |\{w \in V : d(v_n, w) = \ell\}| \leq \frac{K}{4}$, the mouse moves at most $\frac{K}{2}$ steps during this time period. Therefore

$$d(v_{n+1}, m_{2T_{n+1}-1}) \leq d(v_n, m_{2T_n-1}) + \frac{K}{2} \leq \frac{3K}{2}.$$

In both cases, we have $T_{n+1} - T_n \leq \frac{\ell}{4}$ and either $d(v_{n+1}, m_{2T_n+1}) \leq \frac{3K}{2}$ or $d(v_{n+1}, m_{2T_{n+1}-1}) \leq d(v_n, m_{2T_n-1}) - \frac{\ell}{2}$. ◊

By Claim 7.11 and a simple induction, we see that $T_n \leq \frac{1}{4}\sum_{i=1}^{n-1} \ell(v_i)$ and

$$d(v_n, m_{2T_n-1}) \leq \max\left\{d(v_1, m_1) - \frac{1}{2}\sum_{i=1}^{n-1} \ell(v_i), \frac{3K}{2}\right\}.$$

Since we have $d(v_1, m_1) \leq D$, it follows that, if $\frac{1}{2}\sum_{i=1}^{n-1} \ell(v_i) \geq D - \frac{3K}{2}$, then $d(v_n, m_{2T_n-1}) \leq \frac{3K}{2}$. Therefore, if $T_n \geq \frac{D}{2} - \frac{3K}{4}$, then $d(v_n, m_{2T_n-1}) \leq \frac{3K}{2}$. □

7.5 Conclusion

While we have established $\Theta(\sqrt{n})$ as a general upper bound for a cat localising a mouse on a graph, it would be nice to get better bounds depending on the structure of G. Trivially we have $\mathrm{diam}(G)$ as an upper bound. In addition to this, when the diameter of G is $n - R$, we may appeal to Lemma 7.8 with $K = O(R)$ to get that the cat can localise the mouse up to distance $O(R)$. Unfortunately, nothing more can be said when the diameter is between these two extreme cases. Indeed, consider lengthening two of the branches of the tree given in Section 7.3. Lemmas 7.7 and 7.8 are perhaps steps in the right direction here. Lemma 7.7 works well when there is lots of clustering whilst Lemma 7.8 works well when there is very little clustering.

Another interesting direction to consider is the case where multiple cats are working co-operatively to try and localise the mouse exactly. In particular, suppose that there is a collection[1] of k cats $c^1, \ldots, c^k$. At turn i, each cat c^j chooses a vertex c_i^j and is told the value of b_i^j.

There is a simple argument which shows that 7 cats is sufficient to localise the mouse exactly on any tree in finite time. Using two cats, we can detect when the mouse crosses an edge $e = uv$: if c^1 is always on u and c^2 is always on v, the only time when $b_i^1 \neq b_i^2$ is if the cat has just crossed e. Let c^3 start on u and c^4 start on v.

[1]It seems there is a large choice of collective nouns for a collection of cats, including clowder, clutter, destruction (wild cats only), dout, glare, glorying, nuisance, and pounce [53].

By continually switching c^3 and c^4 between u and v, we can also track whether the mouse has moved towards e, away from e or not at all. We say that c^1, c^2, c^3 and c^4 are guarding the edge e.

We can first use c^5 to determine which side of e the mouse is on. Without loss of generality, assume that the mouse is closest to u. We then use c^5, c^6 and c^7 (together with c^1) to guard each edge adjacent to u in turn. By doing this, we can detect which branch the mouse is on unless it changes branches during our tests. If we know which branch the mouse is on, then we can repeat the argument down this branch. Otherwise, we can use the record of the mouse's movements to determine when it changed branches (it was the point when the mouse was closest to u). We now know the mouse's exact distance from u and so we will know if the mouse ever reaches u again. We guard each edge adjacent to u again to determine which branch the mouse is on. We believe that this argument is not optimal and suggest the following questions.

Question 7.12. *What is the minimum number of cats that are needed to localise the mouse exactly on any tree?*

Question 7.13. *What happens on a general graph G? How accurately can k cats localise a mouse?*

Bibliography

[1] K. Asciak, M. Francalanza, J. Lauri, and W. Myrvold, A survey of some open questions in reconstruction numbers, *Ars Combinatoria* **97**, (2010), 443–456.

[2] W. Baird and A. Bonato, Meyniel's conjecture on the cop number: a survey, *Journal of Combinatorics* **3**, (2012), 225–238.

[3] B. Bollobás, Almost every graph has reconstruction number three, *Journal of Graph Theory* **14**, (1990), 1–4.

[4] J. Bondy, A graph reconstructor's manual, *Surveys in Combinatorics*, Cambridge University Press, (1991), 221–252.

[5] J. Bondy and R. Hemminger, Graph reconstruction - a survey, *Journal of Graph Theory* **1**, (1977), 227–268.

[6] A. Bowler, P. Brown, and T. Fenner, Families of pairs of graphs with a large number of common cards, *Journal of Graph Theory* **63**, (2010), 146–163.

[7] A. Bowler, P. Brown, T. Fenner, and W. Myrvold, Recognising connectedness from vertex-deleted subgraphs, *Journal of Graph Theory* **67**, (2011), 285–299.

[8] P. Brown and T. Fenner, The size of a graph is reconstructible from any $n-2$ cards, *Discrete Mathematics* **341**, (2018), 165–174.

[9] C. Bürger, L. DeBiasio, H. Guggiari, and M. Pitz, Partitioning edge-coloured complete symmetric digraphs into monochromatic complete subgraphs, *Discrete Mathematics* **341**, (2018), 3134–3140.

[10] C. Bürger and M. Pitz, Decomposing edge-coloured complete symmetric digraphs into monochromatic paths, (2017), http://arxiv.org/abs/1711.08711.

[11] J. Cutler, Coloring graphs with graphs: a survey, *Graph Theory Notes of New York* **63**, (2012), 7–16.

[12] J. Cutler and A. Radcliffe, The maximum number of complete subgraphs in a graph with given maximum degree, *Journal of Combinatorial Theory Series B* **104**, (2014), 60–71.

[13] D. Dayanikli and D. Rautenbach, Approximately locating an invisible agent in a graph with relative distance queries, *Discrete Mathematics* **341**, (2018), 2302–2307.

[14] L. DeBiasio, R. Krueger, and G. Sárközy, Large monochromatic components in multicolored bipartite graphs, (2018), https://arxiv.org/abs/1806.05271.

[15] L. DeBiasio and P. McKenney, Density of monochromatic infinite subgraphs, (2016), http://arxiv.org/abs/1611.05423v1.

[16] L. DeBiasio and P. McKenney, Density of monochromatic infinite subgraphs, *Combinatorica* **39**, (2019), 847–878.

[17] R. Diestel, *Graph Theory (fourth edition)*, Springer, (2010).

[18] M. Elekes, D. Soukup, L. Soukup, and Z. Szentmiklóssy, Decompositions of edge-colored infinite complete graphs into monochromatic paths, *Discrete Mathematics* **340**, (2017), 2053–2069.

[19] J. Engbers, Extremal H-colourings of graphs with fixed minimum degree, *Journal of Graph Theory* **79**, (2015), 103–124.

[20] J. Engbers, Maximizing H-colourings of connected graphs with fixed minimum degree, *Journal of Graph Theory* **85**, (2017), 780–787.

[21] P. Erdős and F. Galvin, Monochromatic infinite paths, *Discrete Mathematics* **113**, (1993), 59–70.

[22] P. Erdős, A. Gyárfás, and L. Pyber, Vertex coverings by monochromatic cycles and trees, *Journal of Combinatorial Theory Series B* **51**, (1991), 90–95.

[23] A. Frieze, M. Krivelevich, and P. Loh, Variations on cops and robbers, *Journal of Graph Theory* **69**, (2012), 383–402.

[24] T. Gallai, On directed paths and circuits, *Theory of Graphs*, Academic Press, New York, (1968), 115–118.

[25] D. Galvin, Maximising H-colourings of a regular graph, *Journal of Graph Theory* **73**, (2013), 66–84.

[26] C. Groenland, H. Guggiari, and A. Scott, Size reconstructibility of graphs, (2018), `https://arxiv.org/abs/1807.11733`.

[27] H. Guggiari, Monochromatic paths in the complete symmetric infinite digraph, (2017), `https://arxiv.org/abs/1710.10900`.

[28] H. Guggiari, A. Roberts, and A. Scott, Approximating the position of a hidden agent in a graph, (2018), `https://arxiv.org/abs/1805.04386`.

[29] H. Guggiari and A. Scott, Maximising H-colourings of graphs, *Journal of Graph Theory* **92**, (2019), 172–185.

[30] H. Guggiari and A. Scott, Monochromatic components in edge-coloured graphs with large minimum degree, (2019), `https://arxiv.org/abs/1909.09178`.

[31] A. Gyárfás, Partition covers and blocking sets in hypergraphs (in Hungarian), *Communications in Computer and Automation Institute of the Hungarian Academy of Sciences* **71**, (1977), PhD thesis.

[32] A. Gyárfás and G. Sárközy, Large monochromatic components in edge colored graphs with a minimum degree condition, *The Electronic Journal of Combinatorics* **24**, (2017), P3.54.

[33] A. Gyárfás and G. Sárközy, Star versus two stripes Ramsey numbers and a conjecture of Schelp, *Combinatorics, Probability and Computing* **21**, (2012), 179–186.

[34] A. Hajnal, P. Komjáth, L. Soukup, and I. Szalkai, Decompositions of edge colored infinite complete graphs, *Colloquia Mathematica Societatis János Bolyai* **52**, (1987), 277–280.

[35] J. Haslegrave, An evasion game on a graph, *Discrete Mathematics* **314**, (2014), 1–5.

[36] M. Hasse, Zur algebraischen Begründung der Graphentheorie. I, *Mathematische Nachrichten* **28**, (1965), 275–290.

[37] P. Kelly, A congruence theorem for trees, *Pacific Journal of Mathematics* **7**, (1957), 961–968.

[38] P. Kelly, On isometric transformations, University of Wisconsin, PhD thesis, (1942).

[39] J. Lauri and R. Scapellato, *Topics in graph automorphisms and reconstruction*, Cambridge University Press, (2016).

[40] P. Loh, O. Pikhurko, and B. Sudakov, Maximizing the number of q-colorings, *Proceedings of the London Mathematical Society* **101**, (2010), 655–696.

[41] L. Lu and X. Peng, On Meyniel's Conjecture of the cop number, *Journal of Graph Theory* **71**, (2012), 192–205.

[42] V. Müller, Probabilistic reconstruction from subgraphs, *Commentationes Mathematicae Universitatis Carolinae* **17**, (1976), 709–719.

[43] W. Myrvold, Ally and adversary reconstruction problems, University of Waterloo, PhD thesis, (1988).

[44] W. Myrvold, The degree sequence is reconstructible from $n-1$ cards, *Discrete Mathematics* **102**, (1992), 187–196.

[45] C. Nash-Williams, The reconstruction problem, *Selected Topics in Graph Theory*, Academic Press, New York, (1978), 205–236.

[46] R. Nowakowski and P. Winkler, Vertex-to-vertex pursuit in a graph, *Discrete Mathematics* **43**, (1983), 235–239.

[47] A. Quilliot, Jeux et pointes fixes sur les graphes, Université de Paris VI, PhD thesis, (1978).

[48] R. Rado, Monochromatic paths in graphs, *Annals of Discrete Mathematics* **3**, (1978), 191–194.

[49] H. Raynaud, Sur le circuit hamiltonien bi-coloré dans les graphes orientés (in French), *Periodica Mathematica Hungarica* **3**, (1973), 289–297.

[50] B. Roy, Nombre chromatique et plus longs chemins d'un graphe, *Revue Française d'Informatique et de Recherche Opérationnelle* **1**, (1967), 129–132.

[51] A. Scott and B. Sudakov, A bound for the cops and robbers problem, *SIAM Journal on Discrete Mathematics* **25**, (2011), 1438–1442.

[52] L. Sernau, Graph operations and upper bounds on graph homomorphism counts, *Journal of Graph Theory* **87**, (2018), 149–163.

[53] D. Tersigni, Animal group names, (2017), `http://www.thealmightyguru.com/Pointless/AnimalGroups.html`.

[54] Z. Tuza, Ryser's conjecture on transversals of r-partite hypergraphs, *Ars Combinatoria* **16**, (1983), 201–209.

[55] S. Ulam, *A collection of mathematical problems*, Interscience, New York, (1960).

[56] L. Vitaver, Determination of minimal colouring of vertices of a graph by means of boolean powers of the incidence matrix (in Russian), *Doklady Akademii Nauk SSSR* **147**, (1962), 728.

[57] D. Woodall, Towards size reconstruction from fewer cards, *Discrete Mathematics* **338**, (2015), 2515–2522.

Appendix A

Implementing the Code for the Linear Programs

The main obstacle in implementing Linear Programs 4.1 and 4.2 was the large number of possible intersection patterns (2^{27} and 2^{36} respectively) that needed to be checked. We therefore used the implicit symmetry of the problem to reduce the number of linear programs which needed to be run.

Recall that the intersection pattern is given by the variables (y_{ijk}) where

$$y_{ijk} = \begin{cases} 1 & \text{if } R_i \cap B_j \cap G_k \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

We may assume that there are at least 3 components of each colour (Lemma 4.10) and that each component intersects at least one component in each of the other colours (Lemma 4.8).

Given an intersection pattern (y_{ijk}), if there exists i such that $y_{ijk} = 0$ for all j and all k, then this corresponds to the red component R_i being empty (i.e. there are at most two red components). It is therefore not necessary to run the linear program for this intersection pattern. (Indeed, the linear program has a value of $\alpha = \frac{7}{8}$ which is optimal if we do not specify that there must be at least three components of each colour.)

We therefore excluded intersection patterns which corresponded to one of the components being empty. As this check only requires knowledge of the current intersection pattern, it is straightforward to do this when the intersection pattern has been generated.

We also used the symmetry of the problem. There are two sources of symmetry: between components of the same colour and between components of different colours. Let us consider both.

Firstly, suppose we are given two intersection patterns, (y_{ijk}) and (y'_{ijk}). Suppose that, for some $t \in [2]$ and every i, j and k, we have $y_{ijk} = y'_{(i+t)jk}$ where $i + t$ is calculated modulo 3. Any optimal solution for (y_{ijk}) will also be an optimal solution for (y'_{ijk}) but with the red components relabelled. Therefore we only need to run the linear program for one of these intersection patterns to obtain the optimal solution for both. We can extend this idea to any intersection patterns which are the same up to relabelling of the components.

Secondly, if we have two intersection patterns (y_{ijk}) and (y'_{ijk}) such that, for every i, j and k, we have $y_{ijk} = y'_{jik}$, then any optimal solution for (y_{ijk}) will also be an optimal solution for (y'_{ijk}) but with the red and blue components swapped. Again we would only need to run the linear program for one of these intersection patterns to obtain the optimal solution for both. In the case where all three colours have exactly three components, all three colours are interchangeable; in the case where there are three red, three blue and four green components, only the red and blue components may be switched.

Unlike checking whether an intersection pattern corresponds to one of the components being empty, finding intersection patterns which are the same up to symmetry requires knowledge of both the current intersection pattern and other possible intersection patterns. Memory constraints make it impractical to generate all "non-symmetric" intersection patterns before running the linear programs. Instead, we consider only a subset of symmetries that we can handle efficiently using a version of lexicographic ordering.

First, for simplicity, suppose that we only have two colours, red and blue, and that the intersection matrix is given by $Z = (z_{ij})$ where z_{ij} represents whether or not $R_i \cap B_j$ is empty. Swapping two rows in Z corresponds to swapping the labels of two red components and similarly for columns and blue components. We define the *lex value* of Z to be

$$\operatorname{lex}(Z) = \sum_{i=0}^{2}\sum_{j=0}^{2} z_{ij}100^{-i-j}.$$

Swapping pairs of rows and/or pairs of columns in Z can change its lex value and configurations with more 1 entries in the top left corner of the matrix will give higher lex values.

By only swapping pairs of rows or pairs of columns that strictly increase the value of $\operatorname{lex}(Z)$, we will eventually reach Z', a configuration of Z where the lex value is $\operatorname{maxlex}(Z)$, the unique maximum possible lex value of Z. Both Z and Z' are possible intersection patterns and, because we obtained Z' from Z through a series of row and column swaps, Z and Z' are "symmetric" intersection patterns.

We only run a linear program on Z if $\operatorname{lex}(Z) = \operatorname{maxlex}(Z)$. This significantly reduces the number of linear programs that need to be run whilst still ensuring that at least one linear program is run for each class of "symmetric" intersection patterns.

Now consider the situation we actually have where the intersection pattern is given by $Y = (y_{ijk})$. Whilst we could extend the definition of lex value to a three-dimensional matrix, it proved cumbersome to calculate the maximum lex. Instead, we calculated $\operatorname{lex}(Y^{(k)})$ where $Y^{(k)}$ is the 3×3 matrix obtained by restricting to a fixed value of k. We ran the linear program on Y if $\operatorname{lex}(Y^{(k)}) \geq \operatorname{lex}(Y^{(k+1)})$ for every k and $\operatorname{lex}(Y^{(1)}) = \operatorname{maxlex}(Y^{(1)})$. This method eliminated sufficiently many intersection patterns to make the computation tractable.