VICTORIA UNIVERSITY OF WELLINGTON
Te Herenga Waka

School of Mathematics and Statistics
Te Kura Mātai Tatauranga

PO Box 600
Wellington 6140
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Email: sms-office@vuw.ac.nz

Graph Algorithms with Hostile Partners

Matthew Askes

Supervisor: Rod Downey

October 2020

Submitted in partial fulfilment of the requirements for
Bachelor of Science with Honours in Mathematics.

Abstract

A graph algorithm with a hostile partner is a game played between two players, Alice and Bob. In each game Alice and Bob take it in turns to construct some object. Alice wins if the object has a specific property, and Bob wins if it doesn't. In this report we will explore a variety of games along with variations that allow Alice and Bob to play more than once per turn. We will also examine various strategies for both Alice and Bob. And, for each stategy we will see under what conditions Alice or Bob will always win. The four main games we will consider are; the dominating game, the independent dominating game, the colouring game, and the marking game.

Contents

List of Figures

Chapter 1

Introduction

1.1 Graphs

In 1736 the city of Kaliningrad, Russia was known as Königsberg and was part of the now defunct Kingdom of Prussia. Königsberg lay on either side of the Pregel river. The only way to cross the river was by a series of seven bridges. The bridges connected two large islands and are arranged as in figure 1.1.

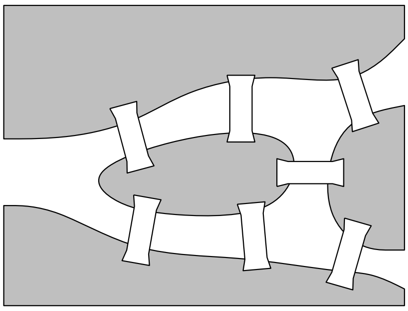

Figure 1.1: The bridges of Königsberg
(Image courtesy of Dillion Mayhew)

The *Bridges of Königsberg Problem* involves finding a way to travel around the city in a way that crosses each bridge exactly once. This problem attracted the attention of Leonhard Euler. Euler noticed that the path through the land masses didn't matter. So to simplify the problem we replace each land mass with a single point (called a vertex) and draw a line (called an edge) between two landmasses if they are connected by a bridge. We now have the representation in figure 1.2, we call this representation a graph. Now the problem is to find a way to travel along each edge exactly once. Euler noticed that apart from the first and last vertex every time we enter a vertex we must also leave it. Thus the number of times we enter a vertex is the same as the number of times we leave. So, all bar two vertices must be connected be an even number of bridges. But in our graph,1 each vertex has an odd number of edges. This means that any path will always get stuck somewhere. Euler concludes that the bridges of Königsberg problem has no solution.

Euler noted that many problems have similar abstractions. Such abstractions are known as graphs and form the basis for graph theory. More generally, a graph $G = (V, E)$ consists of a set of vertices, $V(G)$, and a set of edges, $E(G)$. Each edge connects two vertices and is represented as a pair of vertices. For example in figure 1.3 we have three edges (a, b),

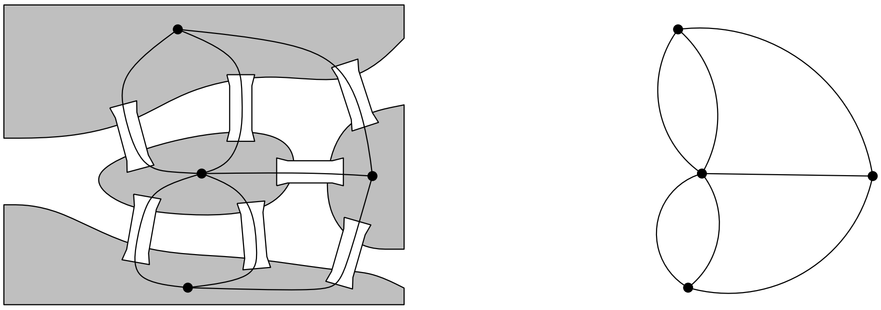

Figure 1.2: The bridges of Königsberg simplified
(Image courtesy of Dillion Mayhew)

(a, c), and (b, c). There are two main types of graphs we consider, directed graphs where the direction of the edge matters ($(a, b) \neq (b, a)$) and undirected graphs where the direction doesn't matter ($(a, b) = (b, a)$).

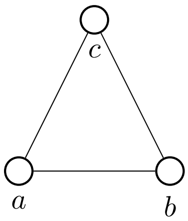

Figure 1.3: An undirected graph with three vertices and three edges

One use of graphs is to represent relationships between objects. They do this in a way that is easy to visualise and analyse. An example from recent events is COVID-19, and more generally infectious diseases. Suppose we are in charge of monitoring an outbreak in a small country. We have a list of all the people who have contracted COVID-19. To represent the problem we consider a graph. We consider people as vertices in the graph and there to be an edge between two people if the disease could spread between them. Then by colouring infected vertices we can easily visualise things like, who is the most infectious (who has the most infected neighbours), and where is community transmission occurring (a group of infections that are disconnected from the rest of the infections).

1.2 Graph Algorithms

A graph algorithm is a set of instructions that define some procedure on a graph. An algorithm can be as simple as finding a vertex of even degree. Or more complicated, as when colouring all the vertices using only a finite number of colours.

Suppose we are charged with laying fibre optic cable in a neighbourhood. Our goal is to connect all the houses as cheaply as possible. We can use a graph to model the relationship between houses. We consider the houses as vertices and there to be edge between two vertices if it is possible to lay cable between the corresponding houses. It costs different amounts to lay cable between different houses. This is because some houses are further apart, have water between them, have harder soil, etc. We associate each edge with a number representing the cost of laying cable between the two corresponding houses. This forms what is known as a *weighted graph*. We now have all the information needed to lay cable. The first step is to lay cable between the corresponding houses at the edge with the least cost. Next we lay cable between the corresponding houses at the edge with the least cost such that laying cable at this edge will not introduce a cycle (a closed path) of cable. By repeating this last step, eventually we will have laid enough cable to connect all the houses. Further, the cable we laid will have the least possible cost. These steps are an example of a graph algorithm. This particular algorithm is called *Kruskal's algorithm*.

1.3 Graph Games

We introduce the idea of a graph game by considering two related problems, the *Dinner Party Problem* and the *Dinner Party Game Problem*.

Suppose Alice is hosting a dinner party, and all the guests are mingling happily. However, the guests are hungry and need to be fed. But, they are lazy and will not move to collect food. When serving, food platters are placed around the room next to particular guests. A guest will take some food if there is a platter within arm's reach. Alice needs to place platters in such a way that every guest can reach a platter. The food is expensive. So, Alice wants to place the smallest number of platters possible. The dinner party problem is, what is the smallest number of platters that Alice needs to feed everybody?

After the success of the first party, Alice decides to host another party. To alleviate the pressure of hosting she decides to hire a caterer, Bob. As before, the platters are placed around the room. A guest will take some food if there is a platter within arms reach. Starting with Alice and on alternating turns Alice and Bob place a single platter. This continues until all the guests are within arms reach of a platter. As before, Alice tries to use the smallest number of platters possible. Bob on the other hand makes a profit for every platter and so will try to place as many platters as possible. Bob is being paid by Alice. So, every platter he places must feed at least one new person. If not, then Alice would fire Bob. If Bob was to always place a platter such that it would feed the least number of people, then the total number of platters placed would be greater than the first party. Hence, Alice requires some strategy to minimise the number of platters placed. The Dinner Party Game Problem is then, what is the smallest number of platters that Alice can guarantee will always feed everyone? We further explore this idea in the context of the domination game and the game domination number in Chapter 2.

It is easy to see how this concept could be applied to other situations. For example, in wartime a nation may be trying to destroy railroads, using the minimum number of bombs possible, but their allies are secretly colluding with the enemy. Such an ally would try to make it as costly as possible to destroy railroads. Other examples include a measure of robustness in network infrastructure, scheduling, and register allocation.

1.4 Report

The Diner Party Problem is a specific example of trying to find what is known as a minimal dominating set. To find such a set Alice would employ some algorithm. Such an algorithm may not be efficient but would solve the Dinner Party Problem. In the Dinner Party Game Problem, Bob is trying to assert his will over this algorithm. Hence, we have a hostile parter as part of our graph algorithm. When the hostile partner (Bob) is included the algorithm will no longer solve the problem. By including Bob we have turned the problem into a game.

In this report we explore four different graph algorithms with hostile partners. These are:

- Dominating Game (Section 2.2):
  Alice and Bob take turns building a dominating set.

- Independent Dominating Game (Section 2.4):
  Alice and Bob take turns building a maximal independent set.

- Colouring Game (Section 3.2):
  Alice and Bob take turns colouring a graph.

- Marking Game (Section 3.3):
  Alice and Bob take turns ordering the vertices of a graph.

In chapter 2 we explore the dominating and the independent dominating games. To begin, we formally define the dominating game. We then introduce upper and lower bounds for some classes of graphs. To bound these classes we provide some explicit strategies for Alice. We then extend the game to allow Alice and Bob to play more than once per turn. Finally, we consider some bounds in the independent dominating game.

In chapter 3 we explore the colouring and marking games. We begin by exploring the standard colouring game. This game is then extended to a version where Alice and Bob play more than once per turn. We then provide lower bounds for the colouring game. The marking game is then introduced as a way to bound the colouring game. To do this we use the activation strategy. We conclude chapter 3 with a brief look at the current best bound for the class of planar graphs.

1.5 Some Notation and Definitions

Before we move on we make a note of some notation and definitions that are common throughout all the games.

All the graphs we have discussed so far are undirected. A directed graph is a graph in which each edge is assigned a direction. In such a graph the edges are not symmetric. That is, the edge (a, b) is not the same as the edge (b, a). An undirected graph can be turned into a directing graph by assigning each edge a direction. For example, in figure 1.4 we directed the graph from figure 1.3.

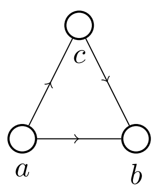

Figure 1.4: A directed graph

In a game, whenever Alice or Bob has their turn by choosing a vertex v we say they *play* the vertex v. For example, in the colouring game a play is when Alice chooses a vertex and assigns it a colour. A *round* is a play of both Alice and Bob.

At any stage in the game we can consider the sequence in which the vertices were played, along with the graph we are playing on. This pair forms a snapshot of the game detailing all the pertinent information. The size of number of played vertices tells you whose turn it is (Alice's if even and Bob's if odd). This snapshot is called a *game state*.

Definition 1.1 (Game State)**.** Let $G = (V, E)$ be a graph. A *game state* for G a sequence $\langle v_n \rangle$ of vertices in V such that each vertex appears at most once.

A strategy for some game is a prescribed next move for every possible game state. In other words a strategy tells a player exactly how to play the game. We define a strategy as a function from game states to game states. Note that the following definition only makes sense for games in which the players only pick vertices. For example in the colouring game the players assign colours to vertices. Hence, the strategy and game state would need to take this extra information into account. To allow us to a have a single definition, we omit this extra detail.

Definition 1.2 (Strategy)**.** For $G = (V, E)$ a graph and $\sigma = \langle v_n \rangle$ a game state for G let

$$\varphi_G \colon V(G)^{<\mathbb{N}} \to V(G)^{<\mathbb{N}}$$

where $V(G)^{<\mathbb{N}}$ denotes the set of all sequences of vertices in V. φ_G is a *strategy* if it is defined on any game state and $\varphi_G(\sigma) = \langle v_0, v_1, \ldots, v_{n-1}, u \rangle$, such that $u \in V$ is a legal move in the game.

A strategy for Alice is a strategy that is only defined on game states where it is Alice's turn. Equivalently, a strategy for Bob is a strategy that is only defined on game states where is Bob's turn.

For some fixed win condition a winning strategy for Alice is a strategy that guarantees that the game will end with Alice winning. Similarly, a winning strategy for Bob is a strategy that guarantees the game will end with Bob winning.

The formalisation of strategies provides good background to the various games. However, to simplify proofs and intuition we will not explicitly define strategies as functions. Rather, we will define the functions implicitly.

Chapter 2

Graph Domination

2.1 Introduction

In chapter 1 we introduced the Dinner Party Problem. To formalize the Dinner Party Problem in the language of graphs we introduce the concept of a dominating set. The Dinner Party Problem can be thought of as a graph game where Alice and Bob are building a dominating set in the graph of guests. A guest is in the dominating set if a platter has been placed next to them. A guest is dominated if they are within arms reach of a platter. The game continues until every guest is dominated (i.e. fed).

Definition 2.1 (Dominating Set)**.** Let G be a graph. A *dominating set* D of $G = (V, E)$ is a subset of V such that every vertex in V is either in D or is adjacent to at least one vertex in D.

Definition 2.2 (Domination Number)**.** Let G be a graph. The *domination number* of G, $\gamma(G)$, is the minimum size of a dominating set in G.

One way of visualising dominating sets is to consider cell tower placement. Suppose we wish to build a 5G network in a city. We consider a graph of buildings in a city. The vertices are buildings and two vertices u and v are connected if a 5G tower placed on building u one will provide coverage to building v. A set of buildings that would provide coverage to the whole city is a dominating set. The minimum number of towers needed is the domination number of this graph.

The size of a minimum dominating set is in some sense a measure of how closely connected a graph is. A graph with a low domination number is densely connected, and a graph with a high domination number is loosely connected. For example, a city that is densely packed requires fewer towers. Every new 5G tower reaches many buildings. Whereas, a sparsely populated city will require more towers. Each tower services only a few buildings.

As a further example, consider the wheel graph W_9 and the cycle graph C_8 in figures 2.1 and 2.2. The wheel has a dominating set of size 1, just the centre vertex. Whereas, the cycle graph has a dominating set of size 3. In figures 2.1 and 2.2 the dominating sets are denoted as black vertices. This can be interpreted as the wheel graph being more closely connected than the cycle graph. And, when you observe the graphs this distinction makes sense.

The *dominating game* was introduced by Brešar, Klavžar, and Rall 2010 [5]. In the dominating game Alice and Bob take turns adding vertices to a set until it forms a dominating set.

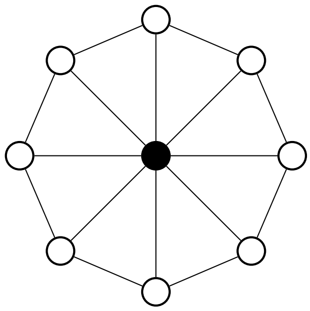

Figure 2.1: The wheel graph W_9

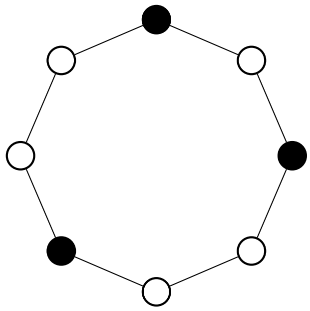

Figure 2.2: The cycle graph C_7

In this chapter we will show how the dominating game is bounded in terms of both the number of vertices and the domination number.

2.2 The Dominating game

For a graph $G = (V, E)$ and a set $X \subseteq V$ we denote $N[X]$ the set of neighbours of X including X. That is $N[X] = \{v \in V : \exists_{u \in X}(u, v) \in E\} \cup X$.

We define the dominating game as follows. Let G be a graph, t a target score, and D a dominating set that we initialise to $D = \emptyset$. On alternating turns, beginning with Alice, Alice and Bob add an unchosen vertex to D such that the number of dominated vertices increases. That is, the set $N[D]$ increases in size. The game stops when D forms a dominating set in G. The score of the game, s, is the size of the dominating set at the end of the game. That is $s = |D|$. Alice wins if $s \leq t$ and Bob wins otherwise.

Definition 2.3 (Game Domination Number)**.** Let G be a graph. The *game domination number* $\gamma_g(G)$ is the minimum target score such that Alice has a winning strategy.

Let $\mathcal{C}$ be a class of graphs. $\gamma_g(\mathcal{C})$ is the smallest k such that for every graph $H \in \mathcal{C}$, $\gamma_g(H) \leq k$. We say a class $\mathcal{C}$ is bounded above by k if $\gamma_g(\mathcal{C}) \leq k$. $\mathcal{C}$ is bounded below by k if there is a graph $H \in \mathcal{C}$ such that $\gamma_g(H) = k$ and we write $k \leq \gamma_g(\mathcal{C})$. Hence, if $k \leq \gamma_g(\mathcal{C}) \leq k$ then $\gamma_g(\mathcal{C}) = k$.

2.2.1 Lower Bounds for the Game Domination Number

A graph $G = (E, V)$ has no dominating sets smaller than $\gamma(G)$. This means that if the target score of the dominating game is less than $\gamma(G)$ then there is no strategy that will allow Alice to win. Therefore $\gamma(G)$ is a lower bound for the game domination number. That is

$$\gamma(G) \leq \gamma_g(G)$$

Theorem 2.4, is a well known result in graph theory.

Theorem 2.4 (Ore 1962 [15])**.** *For any connected graph G with no isolated vertices and n vertices,*

$$\gamma(G) \leq \frac{n}{2}$$

In this report we will show that theorem 2.4 also provides us with a lower bound for the game domination number. But, before we can show $n/2$ is a lower bound we introduce lemmas 2.5 and 2.6.

Lemma 2.5. *For every $n > 1$ there exists a connected graph G with n vertices such that $\gamma(G) = \lfloor n/2 \rfloor$. Hence the bound $\gamma(G) \leq \frac{n}{2}$ is tight*

Proof. Fix $n > 1$. If n is odd then it suffices to show $\gamma(G) \leq (n-1)/2$. So, without loss of generality suppose n is even. Consider the path graph with $n/2$ vertices with a single additional vertex attached to each vertex, denote this graph G. See figure 2.3.

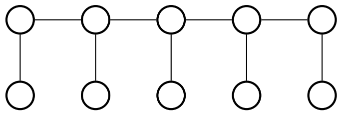

Figure 2.3: The extended path graph, G, with 10 vertices

A minimum dominating set in G is the set of vertices in the original path graph. Hence $\gamma(G) = n/2$. Therefore $n/2$ is a tight upper bound for $\gamma(\mathcal{C})$. □

Lemma 2.6 and theorem 2.7 are implicit in the literature. But, we state them here with proofs.

Lemma 2.6. *Let $\mathcal{C}$ be a class of graphs and $\gamma(\mathcal{C})$ tight upper bound for the domination number of $\mathcal{C}$. That is, for all $G \in \mathcal{C}$, $\gamma(G) \leq \gamma(\mathcal{C})$ and there exists some $G \in \mathcal{C}$ such that $\gamma(G) = \gamma(\mathcal{C})$. Then,*

$$\gamma(\mathcal{C}) \leq \gamma_g(\mathcal{C})$$

Proof. Let $\mathcal{C}$ be a class of graphs and $G \in \mathcal{C}$ a graph such that $\gamma(G) = \gamma(\mathcal{C})$. G has no dominating sets with less than $\gamma(\mathcal{C})$ vertices. Thus there is no winning strategy for Alice with a target score less than $\gamma(\mathcal{C})$. Therefore $\gamma(\mathcal{C}) \leq \gamma_g(\mathcal{C})$ □

Theorem 2.7. *Let G be a connected graph with n vertices, such that $n \geq 4$. Then, there is a winning strategy for Alice with*

$$\frac{n}{2} \leq \gamma_g(G)$$

Proof. Let G be a connected graph with n vertices. By lemma 2.5 theorem 2.4 is a tight upper bound. Therefore by theorems 2.4 and 2.6, $n/2 \leq \gamma_g(G)$. □

Theorem 2.7 does not say that for all connected graphs with domination number less than $n/2$, Alice will always lose. But, rather a target score greater than $n/2$ is needed to ensure Alice will always win on any arbitrary connected graph. As an example, consider the path graph P_n. That, is the graph with n vertices connected in a single line. Let D be the

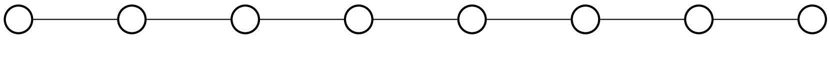

Figure 2.4: The path graph P_8

current partial dominating set in the dominating game. In P_n any vertex that Alice plays will increase the number of dominated vertices ($N[D]$) by at most 3. Bob can play a vertex in $N(D)$. Doing this will increase the size of $N[D]$ by at most 1. Hence, after each round $N(D)$ has increased by at most 4 and D has increased by 2. Therefore the game will end after $n/4$ turns with $|D| = 2(n/4) = n/2$. Hence a target score of $n/2$ is needed to ensure that Alice will win.

2.2.2 Upper bounds for Game Domination Number

Consider some graph $G = (V, E)$, the vertex set V is a dominating set. Therefore when D forms a dominating set $|D| \leq |V(G)|$. Thus a game with target score $|V(G)|$ guarantees Alice will win. Hence $|V(G)|$ is an upper bound for the game domination number. That is,

$$\gamma_g(G) \leq |V(G)|$$

For a better upper bound we introduce a new strategy for Alice. This strategy involves Alice imagining a perfect play and using this play as a strategy. Alice imagines a minimum dominating set and plays only these vertices. This simple strategy and the following bound were first observed in Brešar and Klavžar and Rall 2010 [5]. If Alice was playing herself this minimum dominating set would provide a perfect score ($\gamma_g(G) = \gamma(G)$). However, Alice is not playing herself. Bob's strategy forces a less than perfect score. But, by playing her imagined strategy she will always win on a score strictly less than twice her perfect score.

Theorem 2.8 (Brešar and Klavžar and Rall 2010 [5])**.** *For G a graph and $\gamma(G)$ the dominating number of G,*

$$\gamma_g(G) < 2\gamma(G)$$

In [5] the authors do not give a formal proof, they give a brief sketch. This is something we aim to remedy.

Proof. Let G be a graph and $X \subseteq V(G)$ a dominating set such that $|X| = \gamma(G)$. On Alice's first turn she plays any vertex in X. Now suppose Bob has just played a vertex and D is the current partial dominating set. Alice's strategy is to play any unchosen vertex, $v \in X \setminus D$. As X is a dominating set and Alice plays a vertex from X in each round, after no more than $\gamma(G)$ rounds the game must have ended.

If the game ends in the $\gamma(G)$-th round then it ended on Alice's turn. This is because after Alice's $\gamma(G)$-th turn the game is over. Thus Bob has had one less turn that Alice. In each round Alice and Bob each add one vertex to D. Hence the game ends with size of D exactly $2\gamma(G) - 1$.

If the game ends in strictly less than $\gamma(G)$ rounds then the size of D is strictly less than $2\gamma(G)$. $\square$

If the domination number of a class of graphs is known, then we know an upper bound for the game domination number. However, the domination number of most classes of graphs is not known. This is because finding dominating numbers is difficult. But, if an upper bound for the domination number is known we can still get an upper bound for the game domination number. Alice pretends that the domination number is its upper bound. Alice's strategy would be exactly the same as in theorem 2.8. This means that we can get an improvement on the upper bound of the domination number by finding better upper bounds for the game domination number. However, finding bounds for the dominating number is beyond the scope of this report.

2.2.3 The Domination number in Trees

Finding better bounds for the class of all graphs is a difficult problem. But for other, smaller, classes of graphs better bounds have be found.

Definition 2.9 (Forest)**.** A graph is a *forest* if it contains no cycles.

Definition 2.10 (Tree)**.** A graph is a *tree* if it is connected and contains no cycles.

For the class of trees Kinnersley, West, and Zamani 2013 [14] conjectured the following.

Conjecture 2.11 (3/5-Conjecture for trees [14])**.** *For a forest G with n vertices and no isolated vertices,*

$$\gamma_g(T) \leq \left\lceil \frac{3n}{5} \right\rceil$$

While conjecture 2.11 is still undecided it has been shown for certain types of trees.

A *caterpillar graph* is a tree in which all the vertices lie on a path or have distance at most one from a central path. Figure 2.5 provides some examples. Kinnersley, West, and Zamani

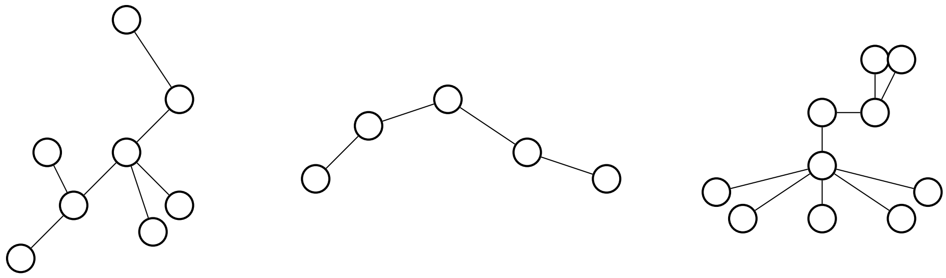

Figure 2.5: Some caterpillar graphs

2013 [14] showed conjecture 2.11 holds for n-vertex forests with no isolated vertices where each component is a caterpillar.

Theorem 2.12 (Kinnersley, West, and Zamani 2013 [14])**.** *For F a forest of caterpillars with no isolated vertices $\gamma_g(F) \leq 3n/5$*

Rather than provide a full proof we give the main ideas; namely we describe Alice's strategy for F, a forest of caterpillars with no isolated vertices.

Let D be the current partial dominating set on Alice's turn. A vertex is *totally dominated* if all it's neighbours are in the dominating set. The *residual graph* F' of F is F less any vertices that are totally dominated. That is $F' = F\backslash\{v \in V(F) : N(v) \subseteq N[D]\}$. The residual graph, F', is the subgraph of F that contains precisely the legal moves. Alice only considers the graph F' when deciding on her next play. Alice considers three cases, in order of preference.

1. F' has a vertex v incident to two leaves. Alice plays v.

2. F' has a component C isomorphic to one of the path graphs P_2, P_4, P_5. Alice plays a winning strategy on C.

3. F' has no vertices incident to 2 leaves and no component isomorphic to one of the path graphs P_2, P_4, P_5. If this is the case then no component of F' has less than 6 vertices. Fix a component C of F and let $u_1, \ldots, u_k$ the longest path in C. If there is some i such that $d(u_i) = 3$ then Alice plays u_i. If not, then Alice plays u_6.

Alice keeps playing according to these rules until there are no more vertices to play. This concludes the strategy.

Further work has been done to improve the class of trees that the conjecture 2.11 holds for. Bujtás, Csilla 2015 [6] showed that conjecture 2.11 holds for forests that have no isolated vertices and in which no two leaves have distance 4.

Theorem 2.13 (Bujtás, Csilla 2015 [6])**.** *For F a forest with no isolated vertices such that no two leaves have distance 4, $\gamma_g(F) \leq 3n/5$.*

Again, rather than provide a full proof we describe Alice's strategy. As part of this strategy Alice assigns a value to each vertex $v \in V(F)$ as follows,

- v has value 3 if it is not dominated.
- v has value 2 if it is dominated but has an undominated neighbour.
- v has value 0 if it is totally dominated.

The *gain* of a play is the difference between the sum of all the values in the tree before and after each play. In Alice's strategy for theorem 2.13 there are 4 stages. At each stage Alice's strategy changes. The game starts in stage 1. If on Alice's turn there is no suitable vertex at that stage then she moves to the next stage.

Stage 1: There is a vertex v with gain at least 7 such that $D \cup \{v\}$ totally dominates two new vertices. Alice plays v.

Stage 2: There is a vertex v with gain at least 7. Alice plays v.

Stage 3: There is a vertex v with gain at least 6. Subject to maximum gain, Alice plays a vertex with value 3 that is incident to a leaf with value 3.

Stage 4: There is a vertex v with gain at least 3. Alice plays v.

This continues until there are no more vertices with gain at least 3. At which point every vertex is dominated. This concludes the strategy.

2.3 The (a, b)-Dominating Game

In the dinner party problem, Alice and Bob each only place one platter in an alternating sequence. But, suppose instead Alice and Bob are each allowed to place 2 platters on their turn. The question here is, does this change the minimum number of platters Alice needs? What about when Alice places more platters than Bob in each turn?

This extended dinner party problem is the natural extension of the dominating game where we allow Alice and Bob to select more than one vertex per turn. We call this extension the (a, b)-dominating game. The rules and win conditions for the (a, b)-dominating game are the same as the dominating game. Except, Alice plays a vertices per turn and Bob plays b vertices.

As far as we are aware, there has been no published work on the (a, b)-dominating game.

Definition 2.14 ((a, b)-Game Domination Number)**.** Let G be a graph and $a, b \geq 1$. The (a, b)-game domination number $\gamma_g(G; a, b)$ is the smallest target score such that Alice has a winning strategy when playing the (a, b)-dominating game.

Theorem 2.15 (Askes)**.** *Let G be a graph with n vertices and $a, b \geq 1$. Then,*

$$\gamma_g(G; a, b) \leq \frac{a+b}{a}\gamma(G) - 1$$

The following proof is an extension of the proof of theorem 2.8 to the (a, b)-dominating game. In fact, when $a = b = 1$ the proof is the same as the proof of theorem 2.8.

Proof. Let G be a graph and $X \subseteq V(G)$ a dominating set of G such that $|X| = \gamma(G)$. On Alice's first move she plays any a vertices in X. Now, suppose Bob has just played b vertices and D is the current partial dominating set. Alice's strategy is to play any a unchosen vertices, $v_1, \dots, v_a \in X \setminus D$. After no more than $\gamma(G)/a$ rounds the game must have ended as X is a dominating set and at least a elements of X are chosen each round. In each round Alice adds a vertices to D and Bob adds b vertices to D, hence the size of D is at most $(a+b)$ times the number of rounds.

If the number of rounds is exactly $\gamma(G)/a$ then the game ends on Alice's turn. This is because Alice starts each round and the game is over on her $(\gamma(G)/a)$-th turn. In such a case Bob has had one less turn than Alice. Hence $|D| \leq (a+b)\frac{\gamma(G)}{a} - b$.

If the number of rounds is strictly less than $\gamma(G)/a$ then the size of D is strictly less than $(a+b)\frac{\gamma(G)}{a}$. In either case the game ends with $|D| \leq (a+b)\frac{\gamma(G)}{a} - 1$. □

2.4 The Independent Dominating Game

To motivate the independent dominating game we introduce the cover band *David and the Derivatives*. David and the Derivatives are the headline act at a large concert. On a normal night there would be no problems filling the seats in the concert hall. However, they are under COVID restrictions. This means that all the guests must be seated 2 meters apart. Alice works for the Ministry of Health. Her job is to seat as few people as possible while still filling the hall. Bob is David and the Derivatives' manager. His job to maximise the number of people seated in the hall. There are only door sales for this concert. So the more people Bob can seat the more profit he and David and the Derivatives make. The guests arrive one by one. Alternating, Alice and Bob direct people to seats. This continues until no more people can be seated without breaking social distancing. Denoting the number of people seated as n. How small can Alice guarantee n will be? And, how large can Bob guarantee n will be? These questions have non-trivial solutions. And are a specific instance of the Independent Dominating Game.

Definition 2.16 (Independent Set)**.** For a graph $G = (V, E)$, an *independent Set* is a subset $I \subseteq V$ such that no two vertices in I are adjacent. That is, for all $u, v \in I$ there exists no edge $(u, v) \in E$.

Definition 2.17 (Independent Dominating Set)**.** For a graph $G = (V, E)$, an *independent dominating set* is a subset of V that is both an independent and a dominating set.

The independent domination game is played on a graph $G = (V, E)$ as follows. Starting with Alice, the players take turns playing a vertex v such that v is not incident with any vertex that is already dominated. That is $v \in V \setminus N[D]$, where D is the current dominating set. Alice and Bob continue playing vertices until D forms a dominating set. The score of the game is the size of the dominating set. Alice's goal is to minimise the score, and Bob's goal is to maximise it.

Since an independent dominating set is a maximal independent set and vice versa the independent dominating game is equivalent to the Independent set game. This game is called the *competition-independence game*. In this game Alice and Bob take turns constructing a maximal independent set. As before Alice tries to minimise the size of the independent set. Bob tries to maximise the independent set. The competition-independence number $I_A(G)$ (respectively $I_B(G)$) is the optimal size of a maximal independent set when Alice starts (or Bob respectively).

The competition-independence game was first introduced in Phillips and Slater 2001 [16]. In [16] the authors demonstrate some bounds for path graphs. In particular, they prove theorem 2.18.

Theorem 2.18 (Phillips and Slater 2001 [16])**.** *For P_n the path graph with n vertices*

$$I_A(P_n) = \lfloor (3n+4)/4 \rfloor$$
$$I_B(P_n) = \lfloor (3n+6)/4 \rfloor$$

Further work on the bounds of competition-independence game was done by Goddard and Henning 2008 [8]. In [8] the authors provide a winning strategy for Alice on trees of with maximum degree less than 3.

Theorem 2.19 (Goddard and Henning 2008 [8])**.** *For any tree T with greater than or equal to 2 vertices and with maximum degree 3*

$$I_A(T) \leq 4n/7$$

We demonstrate the strategy for Alice introduced in [8]'s proof of theorem 2.19. Fix some tree T. Suppose it is Alice's turn. Let I be the current independent set. Define J_I to be the set of isolated vertices in $T \backslash N[I]$. The *energy* of I is defined as

$$\varphi(I) = |I| + |J_I| + 4/7|(V(T) - N[I] - J_I)|$$

Alice's strategy is to play the vertex that minimises the energy of I. That is, the $v \in V(T)$ that minimises $\varphi(I \cup \{v\})$

Compared to the domination and total domination game there has been comparatively little work on the competition-independence game. Recent work by Worawannotai, Ruksasakchai 2020 [19] compares the competition-independence game to the dominating game. Most of the results in [19] are a variation of the type: For a positive integer n, there is some graph G such that

$$\gamma_g(G) - I_A(G) = n$$

There is an interesting fact about the domination game. If Bob has the first move the domination number is less than or equal to $\gamma_g(G) + 1$. To see this consider a graph G. Bob plays v as his first move. Alice pretends she has the first move in the subgraph $H = G \backslash N[v]$. Alice will then win on G with score $\gamma_g(H) + 1 \leq \gamma_g(G) + 1$. This fact is not true for the competition-independence game. As an example, consider the star graph S_7 in figure 2.6.

If Alice starts she will play v and the game ends with $|I| = 1$. If Bob starts then he will play a vertex u_i. In which case Alice cannot play v, so she must play a vertex u_j such that $i \neq j$. The game will end with $|I| = 7$. Thus $I_A(S_7) = 1$ and $I_B(S_7) = 7$.

Definition 2.20 (Independence number)**.** The *independence number* of a graph G, denoted $\alpha(G)$, is the size of a maximum independent set in G.

A graph has no independent sets larger than its independence number. Consequently, when playing the competition-independence game the size of the independent set formed is no larger than the independence number. This means the competition-independence number is bounded above by the independence number. Hence, we get theorem 2.21.

Theorem 2.21. *For a graph G,*

$$I_A(G) \leq \alpha(G)$$

.

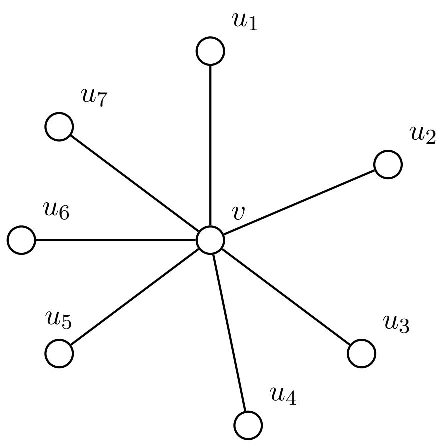

Figure 2.6: The star graph S_7

Chapter 3

Colouring

3.1 Introduction

Long ago all world maps were hand drawn. Alice has a business that specializes in drawing world maps. Each map is beautifully hand coloured. Each country getting its own colour. To ensure the maps are visually appealing and to help distinguish countries no two bordering countries can be the same colour. In those days ink was expensive. So Alice wishes to use the least number of colours possible. What is the smallest number of colours Alice needs to colour a map in such a fashion?

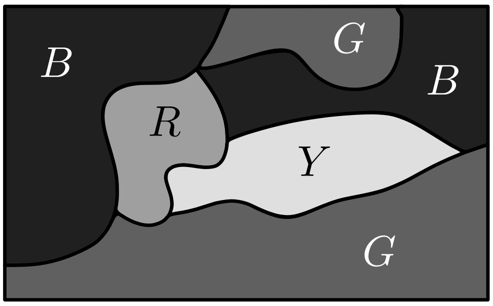

Figure 3.1: A coloured map

(Image courtesy of Dillion Mayhew)

We can translate the problem of map colouring to a problem about graphs. We do this by placing a vertex in each country and join two vertices with an edge if the corresponding countries share a border.

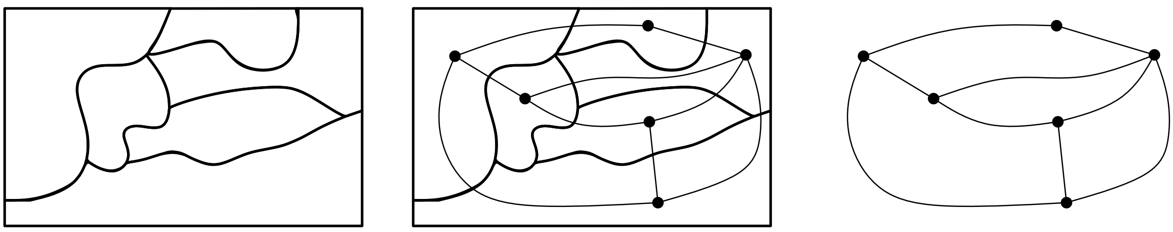

Figure 3.2: Translating maps to graphs

(Image courtesy of Dillion Mayhew)

By constructing a graph in this way we have a one to one correspondence between colourings of the graph and the corresponding map. So, if Alice assigns colours to the vertices such that no two neighbouring vertices have the same colour, she has a way to colour the map. Any graph that can be drawn on the plane (called a planar graph) can be coloured in this manner using only four colours. This is known as the *four colour theorem* [18]. The four colour theorem states that any planar graph can be coloured using only four colours. Thus, Alice only needs 4 colours to colour her maps.

We formally define a colouring as follows.

Definition 3.1 (Proper k-Vertex Colouring)**.** Let $C = \{1, \ldots, k\}$ be a set of colours, a k-vertex colouring of a graph $G = (V, E)$ is a mapping $c \colon V \to C$. A *proper k-vertex colouring* of $G = (V, E)$ is a mapping $c \colon V \to C$ such that for two vertices $u, v \in V$ if $(u, v) \in E$ then $c(u) \neq c(v)$.

When referring to graph colourings we will henceforth be referring to proper k-vertex colourings for some k.

Definition 3.2 (Chromatic Number)**.** The *chromatic number*, $\chi(G)$, of a graph G is the smallest k such that G has a proper k-vertex colouring.

3.2 The Colouring Game

Alice's map colouring business is a huge success. To help colour her maps Alice decides to hire Bob. Unbeknownst to Alice, Bob is part of a secret ink cabal. The cabal's sole goal is to drive up ink sales. Bob will try to use as many colours as possible when colouring maps. To colour a map Alice and Bob take turns assigning each country a colour. They continue until all the countries are coloured. A problem arises almost immediately. By strategically choosing colours Bob can make it so that some maps cannot be coloured with four colours. In fact for some maps Bob can force 8 colours [12]. So, what is the total number of colours Alice needs to ensure that any map can always be coloured? This is an open question. The current best bound is 17 [23]. In section 3.3.5, we will show how this bound was found. We can also ask the same question about classes of graphs that are not planar.

But first, we define the *colouring game*. Let G be a graph, and C a set of colours. Beginning with Alice, Alice and Bob take alternating turns. On their turn they choose an uncoloured vertex, v, and assign v a colour from C such that no two adjacent vertices in G have the same colour. This continues until one of two win conditions are met. First, Alice wins if all the vertices are coloured. Second, Bob wins if there is a vertex that cannot be coloured with the available colours.

Definition 3.3 (Game Chromatic Number)**.** For a graph G the *game chromatic number*, $\chi_g(G)$, is the smallest number of colours such that Alice has a winning strategy for the colouring game on G.

Consider a graph G. If there is a winning strategy for Bob with n colours then $n + 1$ is a lower bound for the game chromatic number of G. That is $n + 1 \leq \chi_g(G)$. Conversely, if there is a strategy for Alice that guarantees a colouring with m colours then m is an upper bound for the game chromatic number. That is $\chi_g(G) \leq m$.

3.2.1 Lower Bounds for the (a, b)-Colouring Game

We consider an extension of the colouring game, the (a, b)-colouring game. In the (a, b)-colouring game the win conditions and rules are the same as the standard game. But, on each turn Alice colours a vertices and Bob colours b vertices.

Definition 3.4 ((a, b)-Game Chromatic Number)**.** Let G be a graph, and $a, b \geq 1$. Then, $\chi_g(G; a, b)$ is the smallest number of colours such that Alice has a winning strategy for the (a, b)-colouring game on G.

Note that $\chi_g(G) = \chi_g(G; 1, 1)$.

The first results we will look at are some lower bounds for the game chromatic number.

Bodlaender 1990 [3] showed that for $\mathcal{T}$, the class of trees, $4 \leq \chi_g(\mathcal{T})$. He did this by defining a tree and an associated strategy for which Bob will always win with 4 colours. We take the proof from [3] and extend it to a new proof of theorem 3.5.

Theorem 3.5. *Let $\mathcal{T}$ be the class of trees. If we have $b \geq 1$ then,*

$$b + 3 \leq \chi_g(\mathcal{T}; 1, b)$$

Proof (Askes). It suffices to show that there exists a tree in which Bob has a winning strategy with $b + 2$ colours.

Consider the tree T as defined in figure 3.3.

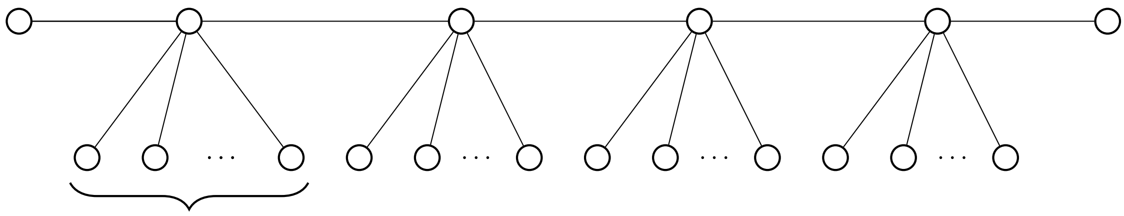

Figure 3.3: A tree, T

Let $\{c_1, c_2, \ldots, c_{b+1}, c_{b+2}\}$ be the set of available colours. On Alice's first move she plays any vertex, v, and colours it. Let the colour of v be c_1. Bobs first move is to colour any vertex with distance 3 to v. We now have a subgraph in T of the type shown in figure 3.4. Bob then colours $y_1, \ldots, y_{b-1}$ with $c_2, \ldots, c_b$ respectively.

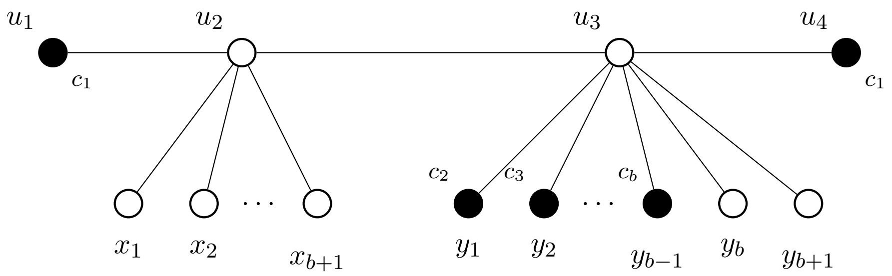

Figure 3.4: A subgraph of the tree T in figure 3.3

We consider three cases.

1. Alice colours u_2, x_1, x_2, $\ldots$, or x_{b+1}.

   Bob colours y_b with c_{b+1} and y_{b+1} with c_{b+2}. u_3 now has $b + 2$ different coloured neighbours and thus Bob wins.

2. Alice colours u_3.

   The colour of u_3 cannot be one of $c_1 \ldots c_b$. Therefore u_3 is either coloured c_{b+1} or c_{b+2}. Without loss of generality assume the colour of u_3 is c_{b+2}. Bob colours $x_1, \ldots, x_b$ with $c_2, \ldots, c_{b+1}$ respectively. Vertex u_2 now has $b+2$ uniquely coloured neighbours and thus Bob wins.

3. Alice colours y_b or y_{b+1}

   Bob colours u_2 with c_{b+1} and y_{b+1} (or y_b if Alice coloured y_{b+1}) with c_{b+2}. u_2 now has $b+2$ uniquely coloured neighbours and thus Bob wins.

Therefore we have a winning strategy on $G \in \mathcal{T}$ for Bob with $b+2$ colours. $\square$

Graphs of Bounded Pathwidth

The class of all trees is a comparatively restricted class of graph. To find a lower bound for a more general class of graphs we temporarily set aside trees and consider graphs of bounded pathwidth. The pathwidth of a graph can be considered a measure of how “path like” it is. For example, the graphs of path width 1 are caterpillars and unions of caterpillars, such graphs are almost paths (see figure 3.5). Figure 3.6 is a graph of pathwidth 3 and is noticeably less path like.

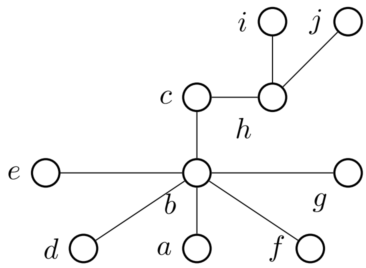

Figure 3.5: A graph of pathwidth 1

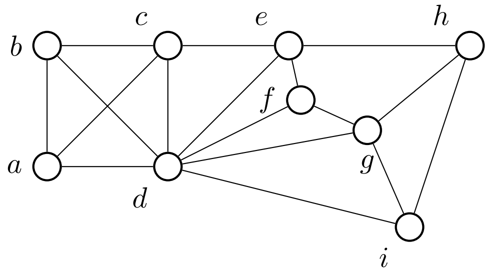

Figure 3.6: A graph of pathwidth 3

Definition 3.6 (Path Decomposition)**.** Let $G = (V, E)$ be a graph. A *path decomposition* is set of subsets of V, $P = (P_1, P_2, \ldots, P_n)$ such that $\bigcup_i P_i = V$ and P has the following properties.

(i) For all edges $(u, v) \in E$ there exists an i such that such $u, v \in P_i$

(ii) If there exists an $i \leq j$ and vertices u, v such that $v \in P_i$ and, $v \in P_j$ then for all $i < k < j$, $v \in P_k$

The *width* of a path decomposition is one less the size if the largest set in P.

Definition 3.7 (Pathwidth)**.** The *pathwidth* of a graph G is the minimum width of a path decomposition of G. A graph of *bounded pathwidth* k is a graph with a pathwidth less than or equal to k.

As an example consider the graphs in figures 3.5 and 3.6. The graph in figure 3.5 has pathwidth 1 and the path decomposition

$$(\{a, b\}, \{b, d\}, \{b, e\}, \{b, f\}, \{b, g\}, \{b, c\}, \{c, h\}, \{h, i\}, \{h, j\})$$

And the graph in figure 3.6 has pathwidth 3 and the path decomposition

$$(\{a, b, c, d\}, \{c, d, e, f\}, \{d, e, f, h\}, \{e, d, f, g\}, \{d, f, g, h\}, \{d, g, h, i\})$$

A maximal graph, G, of pathwidth k is a graph in which we cannot add any more edges to without increasing its pathwidth. In such a graph every element in its path decomposition is a $(k+1)$-clique. These graphs are known as *k-paths*.

Definition 3.8 (k-path)**.** A *k-path* is a maximal graph of pathwidth k.

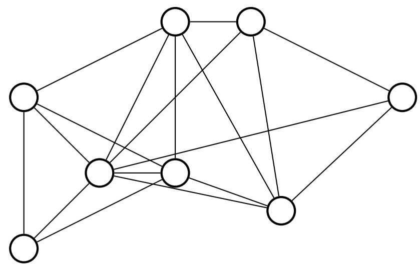

Figure 3.7: A k-path with width 3

A k-path G contains a $k+1$ clique and so there is no proper colouring with less than $k+1$ colours. Hence, when playing the colouring game on a k-path Bob will always win if there are k or less colours.

Theorem 3.9. *For G a k-path,*

$$k+1 \leq \chi_g(G)$$

Theorem 3.10. *Let $\mathcal{P}_k$ be the class of graphs with bounded pathwidth k. If we have $b \geq 1$ then*

$$(b+1)k + \left\lceil \frac{b}{2} \right\rceil \leq \chi_g(\mathcal{P}_k; 1, b)$$

Theorem 3.10 first appeared in Fragile, Kern, Kierstead, Trotter 1993 [7] as a lower bound for the $(1,1)$-colouring game. In [7] the authors do not give a formal proof, rather they sketch a proof by providing the graph on which the game is played. Here we fill in the details by proving a new proof. At the same time we extend the ideas to the $(1,b)$-colouring game.

Proof. It suffices to show that there exists a graph in $\mathcal{P}_k$ for which Bob has a winning strategy with $m = (b+1)k + \lceil \frac{b}{2} \rceil - 1$ colours. Let the set of available colours be $C = \{c_1, c_2, \ldots, c_m\}$.

We define the graph G as follows. Start with a k-clique, K_k, then take $n = 2|C|+1$ vertices, $v_1, \ldots, v_n$ and connect each v_i to each vertex in K_k. Note that for each v_i, $K_k \cup \{v_i\}$ forms a $(k+1)$-clique. Now copy this graph, and connect the copies at any vertex not in K_k. We now have the graph, G, as in figure 3.8. Note that $v_1 = t_1$.

Note that G has the path decomposition

$$\big\{K_k \cup \{v_n\}, K_k \cup \{v_{n-1}\}, \ldots, K_k \cup \{v_1\}, K_k' \cup \{v_1\}, K_k' \cup \{t_2\}, \ldots, K_k' \cup \{t_n\}\big\}$$

and therefore has pathwidth k.

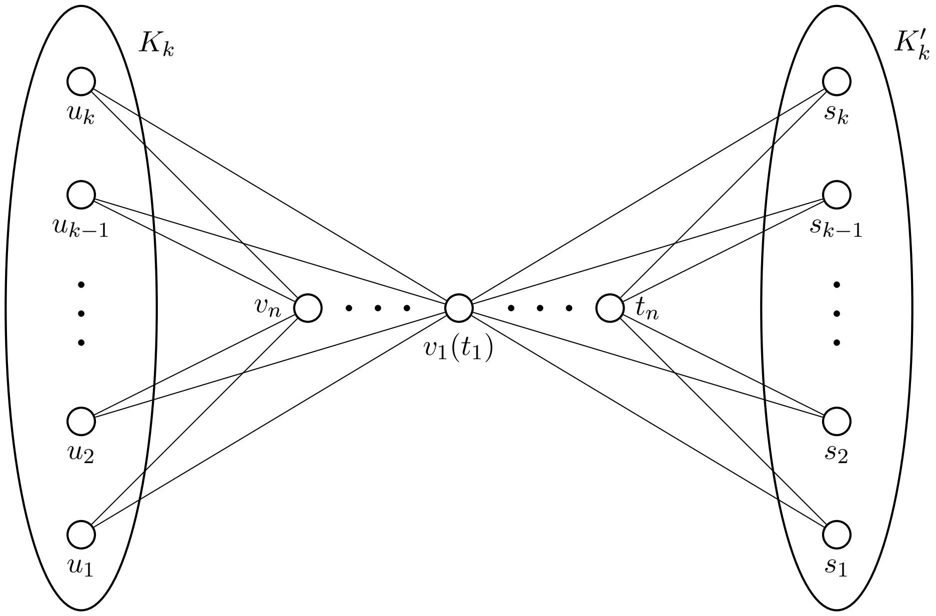

Figure 3.8: Graph G

Consider two disjoint copies of G, G_1 and G_2. On Alices first turn she will colour a vertex in exactly one of G_1 and G_2. On Bob's first turn he can colour a vertex in whichever copy Alice didn't, say G_1, an only play in G_1. Then, without loss of generality we can assume that Bob has the first move in G.

On Bobs first turn he colours v_1 with c_1 and $v_2, t_2, v_3, t_3, \ldots, v_{b/2}, t_{b/2}$ with unique colours. By the symmetry of G we can assume Alice colours one of $t_2, \ldots, t_n, s_1, \ldots, s_k$.

We consider the subgraph, H, in figure 3.9, in which Alice has not yet coloured a vertex.

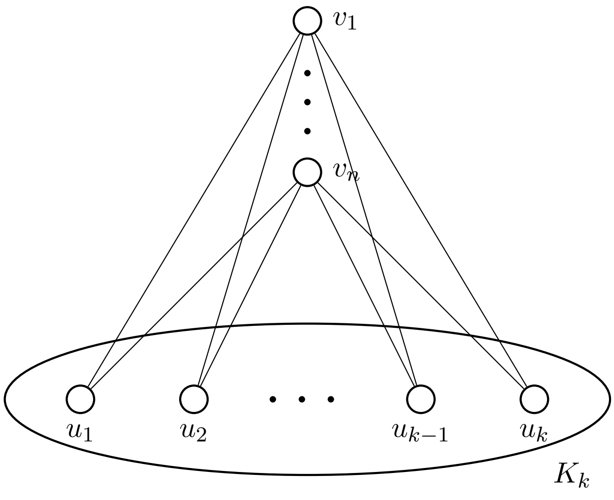

Figure 3.9: Subgraph H of G in figure 3.8

Bob's strategy is to always colour b uncoloured vertices not in K_k with colours not used in H. We keep playing until either K_k is fully coloured bar one, or Bob runs out of new colours. Note that as $n > m$ Bob cannot run out of vertices to colour before he runs out of colours. We consider each case separately.

First, suppose K_k is fully coloured bar one. Let the uncoloured vertex be x. On Bob's first turn he coloured b vertices, but only $\lceil \frac{b}{2} \rceil$ of these vertices are in H. In the second round Alice coloured no vertex in H and bob colured b vertices. Adding b many colours. As each coloured vertex in K_k must have been coloured by Alice there have been at least $k-1$ rounds after the first, and in each round Bob coloured b vertices. Thus $b(k-1)$ many colours have been used in H. Finally, when each vertex in K_k was coloured it must have been coloured differently than the ones before it. This adds $(k-1)$ many colours. Therefore the total number of unique coloured neighbours of x is

$$\left\lceil \frac{b}{2} \right\rceil + b + b(k-1) + (k-1) = (b+1)k + \left\lceil \frac{b}{2} \right\rceil - 1 = |C|$$

Therefore x cannot be coloured, and Bob wins.

Next, suppose Bob has run out of colours. Let y be an uncoloured vertex in K_k. As y is connected to every vertex in $V(H)$, y has $(b+1)k + \lceil \frac{b}{2} \rceil - 1$ many uniquely coloured neighbours. Hence y cannot be coloured. Thus Bob has won. $\square$

3.3 Marking Game

Think back to the Dinner Party Problem. Originally, Alice fed everyone buffet style. But, she finally caught on to Bob's strategy and fired him. To cut down on costs she decides everyone gets a set plate. Now everyone will get a plate, and will always be fed. However, there is a new problem. The guests get upset if too many of their immediate neighbours are fed before them. Consider a graph G with guests as vertices and an edge between a pair of guests if they care about each other being fed. Note that the graph here is different to the one used in the Dinner Party Problem. For each guest v let $d^+(v)$ denote the number of immediate neighbours that are fed before v. Alice wants to make the party go as smoothly as possible. So, she wants a strategy that minimises the value $\Delta^+ = \max_{v \in V(G)} d^+(v)$.

To help hand out the plates, Alice enlists the help of her friend Kate. Kate is known as Bob to her friends. So, to unify this problem with the others we will henceforth refer of Kate as Bob. Alice and Bob will take turns passing out plates. There is a snag. Alice's guest list includes some high ranking government ministers and Bob is a foreign spy. Bob's mission is to ruin Alice's party. In doing so Bob will instil distrust between the ministers. To do this he will attempt to pass out plates that maximises Δ^+. What is the highest value of Δ^+ that Bob can force? This is a specific instance of the marking game.

Introduced by Zhu 1999 [22], the marking game is a simplified version of the colouring game. In the marking game the players simply pick vertices. They don't colour the vertices.

The marking game is played as follows. Let $G = (V, E)$ be a graph, and $t \geq 1$ a target score. Starting with Alice, Alice and Bob take turns choosing unchosen vertices in V. The order in which the vertices were chosen forms a linear order, L. For a vertex v, let $d^+(v)$ denote the number of neighbours of v that are L-less than v. The score of the game is $\max_{v \in V(G)} d^+(v)$. Alice wins if the score is strictly less than t and Bob wins otherwise.

Definition 3.11 (Game Colouring Number)**.** Let G be a graph. The *game colouring number* $\mathrm{col}_g(G)$ is the minimum target score in the marking game such that Alice has a winning strategy.

The marking game is interesting in its own right. But, it has another property that is advantageous. If Alice has a winning strategy in the marking game with target score t then she has a winning strategy in the colouring game with t colours. The marking game is commonly used to find upper bounds for the game chromatic number. For example, the current upper bound for the class of planar graphs was found using a strategy for the marking game [23].

There is simple strategy for Alice that takes a strategy for the marking game and converts it to a strategy for the colouring game. This strategy is called *first fit*. Fix $C = \{c_1, c_2, \ldots, c_k\}$ a set of colours. Suppose Alice has chosen a vertex v in the marking game. In the colouring game she colours v with the least i such that c_i is a valid colour in the colouring game. Let the score of marking game be s. At no point will Alice try to colour a vertex with more than s coloured neighbours. Hence by using first fit Alice has a winning strategy for the colouring game if $|C| \leq s+1$. Note that Alice wins if the target score is equal to the number of colours. Hence the game colouring number bounds the game chromatic number. That is

$$\chi_g(G) \leq \mathrm{col_g}(G)$$

3.3.1 Activation Strategy

Faigle, Kern, Kierstead, and Trotter 1993 [7] introduced a winning strategy for Alice on the class of trees with four colours.

Theorem 3.12 (Faigle, Kern, Kierstead, and Trotter 1993 [7]). *For T a tree $\chi_g(T) \leq 4$*

We present the strategy here, modified for the marking game. This strategy is the Activation Strategy for trees. This example misses some nuance of the full Activation Strategy. However, it provides good motivation for the full strategy.

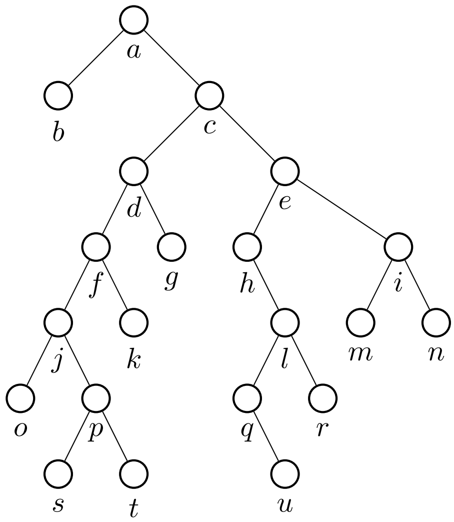

Figure 3.10: A tree, T

Consider the tree T in figure 3.10. T has the vertex set $\{a, b, c, \ldots, t, u\}$. Fix a target score $t = 4$. We consider T to have the root a. Part of Alice's strategy is to keep track of a set of activated vertices, A. Alice starts by marking a and adding a to A. Suppose Bob marks the vertex j. Let P be the path starting with j and traversing up the tree until it reaches a vertex in A. In this case $P = j, f, d, c, a$. Alice adds all the vertices in P to A. Alice's strategy is as follows:

1. If the end of the path P is not marked she marks it.
2. If the last vertex in P is coloured then she colours the second to last vertex.

3. If both the last and second to last vertices are coloured she colours any vertex whose parent is coloured.

The last vertex in P is marked, so by the second rule she marks c. At this stage $A = \{a, c, d, f, g\}$. Bobs next move is to mark k. As before Alice traverses upwards forming a path $P_2 = k, f$. The last vertex in P_2 is f and f is not marked. So by the first rule she marks f. Bob marks g. Alice adds g to A and marks d for the same reason as her last turn. The current game state is represented in figure 3.11, with superscripts representing the order the vertices were chosen.

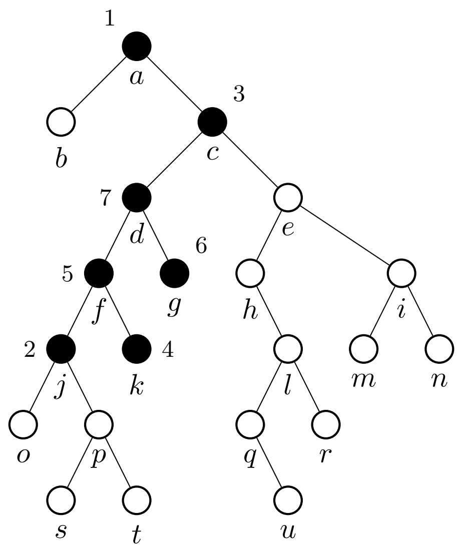

Figure 3.11: The tree T after 6 turns

Some final example plays are as follows. Bob marks q. Alice adds q, l, h, e to A and marks e. Bob marks r. Alice adds r to A and marks l. Bob marks h. By the third rule Alice marks b. The game proceeds in this manner until all the vertices are marked.

When using this strategy Alice will win if the target score is at least 4 (for a proof see section 3.3.3). The full Activation Strategy was introduced by Kirstead 2000 [10]. The Activation Strategy is an extension of the strategy just described to the marking game for arbitrary graphs.

3.3.2 Summary of the Activation Strategy

Before we formally define the Activation Strategy, we need the concept of an *induced direction* in a graph. Let $G = (V, E)$ be a graph and fix a linear order L on V. G_L is the directed graph induced by L on the graph G as follows. We have the directed edge $(u, v) \in G_L$ if and only if $u >_L v$ and $(u, v) \in E$. We call G_L the direction induced by L on G. This ordering may be the reverse of what you might expect. The reason for this is when using the Activation Strategy Alice wants to traverse the vertices from biggest to smallest in L. Hence, we want the edges directed from biggest to smallest. The less than symbol, $>_L$, can be thought of as an arrow pointing to the next element in the directed graph.

We need a little more notation. Let v be a vertex in $V(G)$ and $N(v)$ be the set of neighbours of v. Then with respect to v we have the following

- The *out-neighbours* are $N^+_{G_L}(v) = \{u \in N(v) : v >_L u\}$

- The *in-neighbours* are $N^-_{G_L}(v) = \{u \in N(v) : v <_L u\}$
- The *out-degree* is $d^+_{G_L}(v) = |N^+_{G_L}(v)|$
- The *in-degree* is $d^-_{G_L}(v) = |N^-_{G_L}(v)|$
- $V^+_{G_L}(v) = \{u \in V(g) : v >_L u\}$
- $V^-_{G_L}(v) = \{u \in V(g) : v <_L u\}$

In G_L the maximum out-degree is $\Delta^+_{G_L}(G) = \max_{v\in V(G)} N^+_{G_L}(v)$, and the maximum in-degree is $\Delta^-_{G_L}(G) = \max_{v\in V(G)} N^-_{G_L}(v)$.

Finally,

- $N^+_{G_L}[v] = N^+_{G_L}(v) \cup \{v\}$
- $N^-_{G_L}[v] = N^-_{G_L}(v) \cup \{v\}$
- $V^+_{G_L}[v] = V^+_{G_L}(v) \cup \{v\}$
- $V^-_{G_L}[v] = V^-_{G_L}(v) \cup \{v\}$

When it is clear from context which directed graph we are referring to, we will drop the G_L subscript. A simple way to think about this notation is to consider $+$ as before in the linear order and $-$ as after. For example $N^+(v)$ is the set of neighbours before v.

We can now give a formal description of the Activation Strategy. The Activation Strategy can be summarised as follows.

1. Alice starts by marking the least v in L.
2. On her next turn let u be the last vertex marked by Bob. Alice starts at u *activates* it and moves to w the least unmarked neighbour of u in L.
3. If w is activated or has no unmarked neighbours then Alice marks w. If not Alice repeats step (2) on w until she either finds an active vertex that is activated or has no unmarked neighbours.

Algorithm 3.1 Activation strategy

$x \leftarrow b$
while $x \notin A$ **do**
    $A := A \cup \{x\}$
    $s(x) = \min_L(N^+[x] \cap (U \cup \{b\}))$
    $x \leftarrow s(x)$
if $x \neq b$ **then**
    play x
else
    $y \leftarrow \min_L U$
    if $y \neq A$ **then**
        $A \leftarrow A \cup \{y\}$
    play y

Definition 3.13 (Activation strategy [10])**.** Let G be a graph and L a linear ordering on $V(G)$. We define the Activation Strategy $S(L, G)$ as follows:

Let U denote the set of unmarked vertices. Alice maintains a subset $A \subset V(G)$ of active vertices. Initially $A = \emptyset$. We activate a vertex x by adding it to A. On her first turn Alice activates and marks the least vertex in the ordering L. Now suppose that Bob has just marked the vertex b. Alice uses algorithm 3.1 to update A and choose the next vertex to play.

There are multiple ways of finding bounds for the game colouring number using the Activation Strategy. One such method is to use the concept of matchings. This is the original method undertaken by Kierstead 2000 [10].

Definition 3.14 (Matching)**.** Let $G = (V, E)$ be a graph. A *matching* $M \subset E$ is a set of independent edges, that is a set of edges that share no common vertices. We say M is a matching from X to Y (and is denoted $M: X \to Y$) if $X, Y \subseteq V$ and every vertex in X is joined by an edge in M to some vertex in Y. That is for all $u \in X$ there is $(u, v) \in M$ such that $v \in Y$. And, we write .

Definition 3.15 (Matching Number)**.** Let $G = (V, E)$ be a graph and L a linear order on V. For $u \in V$ the *matching number*, $m(u, L, G)$, of u with respect to L in G is the size of the largest set $Z \subseteq N^-[u]$ such that there exists a partition $\{X, Y\}$ of Z and there exist matchings M from $X \subset N^-[u]$ to $V^+(u)$ and N from $Y \subseteq N^-(u)$ to $V^+[u]$.

Consider figure 3.12. In this figure, all the in-neighours of u ($N^-(u)$) are on the left and the vertices before u ($V^+(u)$) are on the right. The matching M from $X \subset N^-[u]$ to $V^+(u)$ joins the red regions. The matching N from $Y \subset N^-(u)$ to $V^+[u]$ joins the blue regions. Note that there may be edges joining the in-neighbours to the set $V^+(u)$.

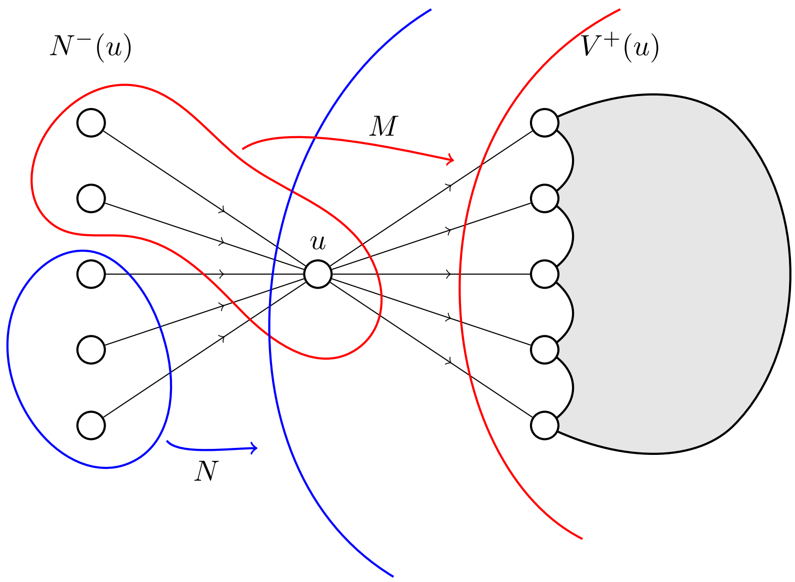

Figure 3.12: The matchings that form the matching number

Definition 3.16 (Graph Rank [10])**.** Let $G = (V, E)$ be a graph, L a linear order on V, and $\Pi(G)$ the set of all linear orders on V. The ranks $r(L, G)$, $r(L, G)$ and $r(G)$ are defined as:

$$r(u, L, G) = d^+_{G_L}(u) + m(u, L, G)$$
$$r(L, G) = \max_{u \in V} r(u, L, G)$$
$$r(G) = \min_{L \in \Pi(G)} r(L, G)$$

Fix some graph $G = (V, E)$, and a linear order L on V. Denote the set of activated vertices A. Every vertex that is marked is immediately activated by Alice. So at the end of her turn any unmarked vertex, u, has at most $|A \cap N(u)|$ many marked neighbours. So the score of the game is at most $|A \cap N(u)|$ for any unmarked vertex u at the end of all Alice's turns. We can partition $|A \cap N(u)|$ into two pieces, the out-neighbours and in-neighbours of u. Let $X = A \cap N^+(u)$ and $Y = A \cap N^-(u)$. $|A \cap N^+(u)| \leq d^+(u)$, this is where the d^+ term comes from in the rank. For any vertex x, let $s(x)$ denote such the least unmarked out-neighbour of x. $s(x)$ is either activated before or after x. So we can partition $A \cap N^-(u)$ into two pieces,

$$\{x \in N^-(u) \cap A : \text{x is activated before } s(x)\}$$
$$\{x \in N^-(u) \cap A : \text{x is activated after } s(x)\}$$

The matching number is the largest possible size of these sets. And so, the matching number bounds the size of $A \cap N^-(u)$.

Most bounds found when using the Activation Strategy involve finding the rank of a graph. Once the rank is known we have a linear order L on which to play the Activation Strategy. When using the activating strategy on such a L we get the following upper bound for the game colouring number.

Theorem 3.17 (Kierstead 2000 [10]). *For any graph $G = (V, E)$ and linear order L on V, if Alice uses the Activation Strategy $S(L, G)$ to play the marking game on G, then the score will be at most $1 + r(L, G)$. In particular,*

$$\mathrm{col_g}(G) \leq 1 + r(G)$$

The following is the original proof from [10] modified to unify it with the definitions in this report. We have also made some minor changes to make it easier to follow.

Proof. Fix $G = (V, E)$ a graph and L a linear order on V. We need to show that on any turn any unchosen vertex u has at most $r(u, L, G)$ many active neighbours. Denote the set of active vertices A and the set of unmarked vertices U. The main task is to show that $|N^-(u) \cap A| \leq m(u, L, G)$. Once this is done we have

$$\begin{aligned}|N(u) \cap A| &\leq d^+(u) + |N^-(u) \cap A| \\ &\leq d^+(u) + m(u, L, G) \\ &= r(u, L, G)\end{aligned}$$

and the result follows.

Let $s(x)$ be the L-least unmarked vertex in $N^+[x]$. Note that $s(x)$ is the same as defined in algorithm 3.1. This means that if x has just been activated $s(x)$ will be activated immediately after x. Define the sets P and Q as follows,

$$P = \{x \in N^-(u) \cap A : \text{x is activated before } s(x)\}$$
$$Q = \{x \in N^-(u) \cap A : \text{x is activated after } s(x)\}$$

We need to show that s is injective when its domain is restricted to either P or Q. Doing this will allow us to form matchings of the form $(x, s(x))$. First, let $x, y \in P$ such that $x \neq y$ and x was activated before y. $s(x)$ is the next vertex activated after x. So either $y = s(x)$ or $s(x)$ was activated before y. In either case $s(y)$ must have been activated after $s(x)$. Thus $s(x) \neq s(y)$.

Next, let $x, y \in Q$ such that $x \neq y$ and x was activated before y. $s(x)$ was activated before x. So $s(x)$ was marked immediately after x. Thus x was marked before y was activated. Hence when y is activated $s(x)$ is not a valid choice for $s(y)$. Thus $s(x) \neq s(y)$.

$\{P, Q\}$ is a partition of $N^-(u) \cap A$. Note that for any vertex $x \in N^-(u) \cap A$, $u \in N^+(x) \cup U$. This means that $s(x) \leq u$ and so $s(x) \in V^+[u]$. Thus $S_P = \{(x, s(x)) : x \in P)\}$ and $S_Q = \{(x, s(x)) : x \in Q)\}$ are matchings from $N^-(u)$ to $V^+[u]$.

There is a problem with these matchings. They both go from $N^-(u)$ to $V^+[u]$. To match the definition of $m(r, L, G)$ we need one of S_P and S_Q to go from $N^-[u]$ to $V^+(u)$. If there are no vertices $x \in P, y \in Q$ such that $s(x) = u = s(y)$ then both matchings are from $N^-(u)$ to $V^+(u)$ and we are done.

If there are vertices $x \in P, y \in Q$ such that $s(x) = u = s(y)$ then fix some such x and y. Since u is unchosen it must be the case that $u \in N^-(s(u))$.

If $u \in P$ then set $X = (P' \setminus x) \cup \{u\}$ and $Y = Q$.

If $u \notin P$ then set $X = (Q' \setminus y) \cup \{u\}$ and $Y = P$.

In either case $S_X = \{(x, s(x)) : x \in X)\}$ is a matching from $N^-[u]$ to $V^+(u)$ and $S_Y = \{(x, s(x)) : x \in Y)\}$ is a matching from $N^-(u)$ to $V^+[u]$. So we have our matchings as desired. □

3.3.3 Upper Bounds Using the Activation Strategy

The proofs in this section follow a pattern. First, we give a linear order for all the graphs in the class. Then we find the maximum possible in-degree. Finally, we bound the maximum size of the two matchings the form the matching number.

Recall that theorem 3.12 states that for any tree T

$$\chi_g(T) \leq 4$$

To demonstrate the basic pattern and idea behind the proofs in this section we prove theorem 3.12 using the Activation Strategy. This proof appears in [10], but few details are given. Here we give a full proof.

Proof of Theorem 3.12. Fix a tree $T = (V, E)$. By theorem 3.17 $\mathrm{col_g}(T) \leq 1 + r(T)$. So it suffices to show that the rank of T is less than or equal to 3. We define a linear order L on V by using *breath first traversal*. First we pick some vertex r to be the root of our tree. r becomes the least element in L. We then add all the children of r then all of their children and so on. For an example see figure 3.14.

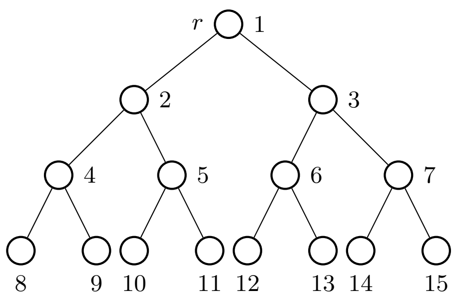

Figure 3.13: Breath first traversal

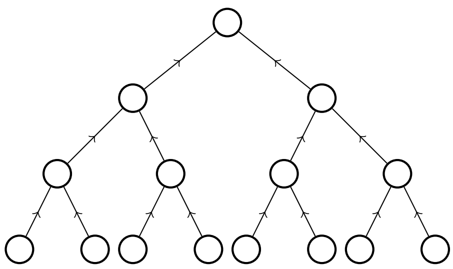

Figure 3.14: Directed graph induced by breadth first traversal

Fix some $v \in V$. We show $r(v, L, T) \leq 3$. Every vertex $u \in V$ apart from r has $d^+(u) = 1$. For r, $d^+(r) = 0$. In other words every vertex has most a single parent. Hence $d^+(v) \leq 1$.

Let $Z \subseteq N^-[v]$ and $\{X, Y\}$ a partition of Z such that $X \subseteq N^-[v]$ and $Y \subseteq N^-(v)$.

It remains to show that if there exist matchings $M : X \to V^+(v)$ and $N : Y \to V^+[v]$ then $|Z| = |X| + |Y| \leq 2$. This is the crux of the entire proof. In short, we bound X and Y by assuming they are a part of the matchings M and N.

Assume there exist such matchings M and N.

Assume for a contradiction that there is an edge (x, y) between $N^-(v)$ and $V^+(v)$. Thus, $x \in N^-(v)$ and $y \in V^+(v)$. As $x >_L v >_L y$, $x \in N^-(y)$. Thus $y \in N^+(x)$. We also know that $v \in N^+(x)$. And thus as $d^+(x) = |N^+(x)| = 1$, $y = v$. This contradicts the fact that $v \notin V^+(v)$. Therefore there are no edges between $N^-(v)$ and $V^+(v)$

Therefore M can contain at most the single edge (v, u) for some $u \in N^+(v)$. Thus the only vertex X can contain is v. The same is true for N. N can contain at most the single edge (u, v) for some $u \in N^+(v)$. Thus the only vertex Y can contain is u. Therefore $|Z| = |X| + |Y| \leq 2$. Then we have

$$r(v, L, T) = d^+(v) + m(v, L, T) \leq 1 + 2 = 3$$

as desired. □

Recall that theorem 3.5 states that for the class of trees $\mathcal{T}$, $\chi_g(\mathcal{T}; 1, b) \geq b + 3$. When $b = 1$ we get the lower bound for the game domination number on the class of trees. Further, when combined with theorem 3.12 we get that $4 \leq \chi_g(\mathcal{T}) \leq 4$. Hence we get corollary 3.18 as an immediate consequence.

Corollary 3.18. *For the class of trees $\mathcal{T}$*

$$\chi_g(\mathcal{T}) = 4$$

Graphs of Bounded Pathwidth

Interval graphs are graphs defined from a series of closed intervals in the real numbers. The vertices represent the intervals and two vertices are connected with an edge if the associated intervals overlap. For an example see figure 3.15. The *clique width* of an interval graph is the

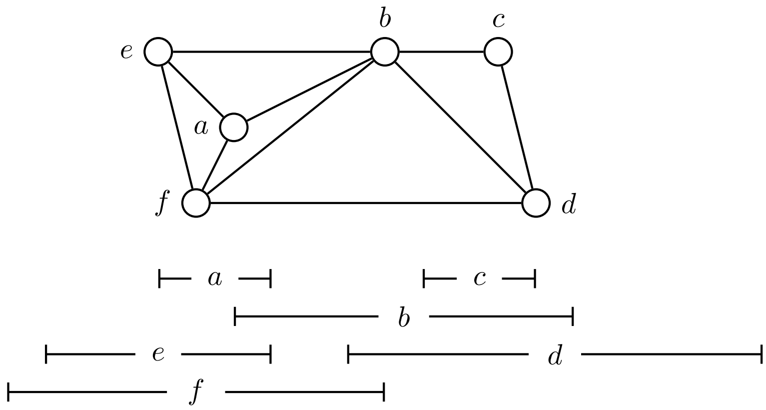

Figure 3.15: An interval graph with clique width 4

size of its maximum clique. The *interval width* of a graph G is the minimum clique width of all interval graphs that contain G as a subgraph. It is shown in Faigle, Kern, Kierstead, and Trotter 1993 [7] that for any graph with interval width w, $\chi_g(G) \leq 3w-2$. Building on this result, Kierstead 2000 [10] uses the Activation Strategy to prove the same result. The pathwidth of a graph is one less than its interval width [4]. This then gives us theorem 3.19.

Theorem 3.19. *Let G be a graph of pathwidth k. Then,*

$$\chi_g(G) \leq 3k+1$$

The proof that we present here is new and based directly on our definition of pathwidth. Rather than the previous results ([7, 10]) whose proofs are based on properties of interval graphs.

Proof. Let $G=(V,E)$ be a graph with pathwidth k, and $P=\{P_1,\ldots,P_n\}$ a path decomposition of width k.

Consider a linear order L on V such that for all $i<j$, all elements in P_i are less than all elements in $P_j \setminus P_i$.

Without loss of generality assume G is maximal, that is all P_i are cliques.

Claim. *For all $v \in V$, $d^+(v) \leq k$.*

Proof of Claim. Fix some $v \in V$. Let i be the least such that $v \in P_i$. Note that $|P_i \setminus v| \leq k$.

Let $x \in N^+(v)$. As there is an edge $(x,v) \in E$ there must be some j such that $x, v \in P_j$.

If $j > i$ then by construction of L, $v <_L x$. But this contradicts the fact that $x \in N^+(v)$. Therefore $j \leq i$. Then by the minimality of i, $i=j$. Therefore $x \in P_i$.

Thus $N^+(v) \subseteq P_i \setminus v$. Hence $d^+(v) = |N^+(v)| \leq |P_i \setminus v| \leq k$. ■

$\chi_g(P) \leq \mathrm{col}_g(P)$ and $\mathrm{col}_g(G) \leq 1 + r(G)$. So, by theorem 3.17 it suffices to show that for any vertex $v \in V$, $r(v,L,G) \leq 3k$.

Let v be any vertex in V. Let $Z \subset N^-[v]$ and $\{X,Y\}$ a partition of Z such that $X \subset N^-[v]$ and $Y \subset N^-(v)$.

Consider a matching $M\colon X \to V^+(v)$. Let (a,b) be an edge in M. $v \in N^+(a)$ and $N^+(a)$ is a clique as $N^+(a) \subseteq P_i$ for some i. Therefore b is adjacent to v that is $b \in N^+(v)$. Hence every edge in M goes from X to $N^+(v)$. Thus $|X| \leq |N^+(v)| \leq k$.

Consider a matching $N\colon Y \to V^+[v]$. Let (a,b) be an edge in N. Let i,j be the least such that $a \in P_i$ and $b \in P_j$. As (a,b) is an edge, by the definition of path decomposition, $b \in P_i$ or $a \in P_j$. If $a \in P_j$ then $a \in V^+[v]$, this contradicts the fact that $a \in N^-(v)$. Therefore $b \in P_i$. As $v \in P_i$ and P_i is a clique b is adjacent to v, that is $b \in N^+(v)$. Hence every edge in N goes from N to $N^+(v)$. Thus $|Y| \leq |N^+(v)| \leq k$.

$m(v,L,G) \leq |Z| = |X| + |Y|$. Therefore by the definition of rank,

$$r(v,L,G) = d^+_{G_L}(v) + m(v,L,G) \leq k + |X| + |Y| \leq 3k$$ □

Graphs of Bounded Treewidth

Definition 3.20 (Tree decomposition)**.** A tree decomposition (X, T) of a graph $G = (V, E)$ is a tree, T, along a collection of subsets of V, $X = \{X_1, \ldots, X_n\}$, indexed by vertices in T, such that $= \bigcup_i X_i = V$ and X obeys the following properties.

(i) For all edges $(u, v) \in E$ there exists an i such that $u, v \in X_i$

(ii) If there exists an $x, y \in V(T)$ and vertices u, v such that $v \in X_x$ and $v \in X_y$ then for all l on the path from x to $y, v \in P_l$.

The *width* of a tree decomposition is the maximum value of $|X_i| - 1$ for all $i \in V(T)$.

Definition 3.21 (Treewidth)**.** The *treewidth* of a graph G is the minimum width of a tree decomposition of G. A graph of *bounded treewidth* k is a graph with treewidth less than or equal to k.

Just as pathwidth is a measure of a graphs "path-ness", treewidth is a measure of how "tree like" a graph is. Though treewidth had been treated informally before, the concept of treewidth was formally introduced by Robertson and Seymour during their seminal work on graph minors [17]. The initial use was to show that if G is a planar graph then there is a k, depending solely on G, such that any graph H with no minor isomorphic to G has tree width at most k. Since then, extensive work as been done using the concept of treewidth. For example, treewidth can be used to solve hard decision problems on graphs restricted to bounded tree width in polynomial time [2].

There is a relationship between our definitions of treewidth and pathwidth. The graphs of bounded pathwidth are exactly the graphs of bounded treewidth whose underlying tree structure forms a path. Thus, every graph of bounded pathwidth k is a graph of bounded treewidth k.

To find an upper bound for graphs of bounded treewidth we use algorithm 3.2 to generate a linear order on the vertices of a graph, G. Such an ordering is based on a minimal tree decomposition, (X, T). Algorithm 3.2 traverses T using breadth first traversal and adds any vertex from X_i to the linear order if it is the first time we had encountered the vertex. In a graph of treewidth 2 (i.e. a tree) algorithm 3.2 is just breath first traversal.

Algorithm 3.2 Linear order in tree decomposition

Require: $(X = \{X_i : i \in V(T)\}, T)$ is the tree decomposition of a graph $G = (V, E)$. r is the root of T
Ensure: L is a linear order on V
function LOInTree((X, T), r)
    $L \leftarrow \emptyset$ ▷ our linear order for V
    let Q be a FIFO queue
    Q.enqueue(r)
    mark r as visited
    while Q is not empty **do**
        $v \leftarrow Q$.dequeue()
        $L \leftarrow L \cup \{v \setminus L\}$ ▷ add all elements in $V(G)$ not already in L
        for all $U \in N(v)$ s.t. U is unvisited **do**
            Q.enqueue(U)
            mark U as visited
    return L

Lemma 3.22 (Askes)**.** *Let $G = (V, E)$ be a graph of treewidth k and $(\{X_1, X_2, \ldots, X_n\}, T)$ a tree decomposition of G with width k. If L is a linear order on V generated by algorithm 3.2, then for any vertex v in V*

$$d^+(v) \leq k$$

Proof. Fix a graph $G = (V, E)$. Let L be the linear ordering on V generated by algorithm 3.2. Let v be any vertex in V.

Suppose that algorithm 3.2 has just traversed $X_i \in X$ and added v to L. All the neighbours of v that are L-less than v must have already been traversed. By the definition of tree decomposition all the neighbours of v that have been traversed are in X_i. Thus $N^+(v) \subseteq X_i$. Note that $|X_i \setminus v| \leq k$. Thus there are at most k neighbours of v in L. □

A k-tree is a maximal graph with treewidth k. This characterization of k-trees means that a graph with treewidth k is a subgraph of some k-tree. Therefore, to find a bound for the class of trees with bounded treewidth we only need to find a bound for the class of k-trees. Theorem 3.23 was first introduced in Wu, Zhu 2008 [20]. In [20] the authors found a bound for k-trees using a concept they call pseudo-partial k-trees. We give a new proof based on the Activation Strategy rather than pseudo-partial k-trees.

Theorem 3.23. *For G a graph of treewidth k,*

$$\chi_g(G) \leq 3k + 2$$

Proof. Suppose G is a graph of treewidth k. Without loss of generality assume that G is maximal, that is G is a k-tree. Let $(\{X_1, X_2, \ldots, X_n\}, T)$ be a tree decomposition of G with width k. As G is a k-tree each X_i is a clique such that $|X_i| = k + 1$.

Let v be a vertex in G. It suffices to show $r(v, L, G) \leq 3k + 1$.

Note that by lemma 3.22 for all $v \in V(G)$, $d^+(v) \leq k$.

Let $Z \subset N^-[v]$ and $\{X, Y\}$ a partition of Z such that $X \subset N^-[v]$ and $Y \subset N^-(v)$.

Consider a matching $M\colon X \to V^+(v)$. Let (a, b) be an edge in M. $N^+(a)$ is a clique and $v \in N^+(a)$. Thus b is adjacent to v. Therefore $X = \mathrm{rng}(M) \subset N^+(v)$. Therefore $|X| \leq |N^+(v)| \leq k$.

Consider a matching $N\colon Y \to V^+[v]$. Let (a, b) be an edge in N. Note that $N^+(a)$ is a clique and $v \in N^+(a)$, thus b is adjacent to v. Therefore $Y = \mathrm{rng}(N) \subset N^+[v]$. Therefore $|Y| \leq |N^+[v]| \leq k + 1$.

$m(v, L, G) \leq |Z| = |X| + |Y|$. Therefore by definition of rank

$$r(v, L, G) = d^+(v) + m(v, L, G) \leq 3k + 1$$ □

3.3.4 Extending the Activation Strategy

The Activation Strategy works well for the $(1, 1)$-marking game but needs modifying to be used in the more general (a, b)-marking game. We will look at two different extensions. The first was introduced by Kierstead and Yang 2005 [13]. Their strategy is called the *Harmonious Strategy.* The second extension we will consider was introduced by Yang and Zhu 2008 [21]. This strategy is called the *Asymmetric Activation Strategy* and is built on work stemming from the Harmonious Strategy.

For a graph $G = (V, E)$ let $\Pi(G)$ be the set of all linear orders of V, and let $\Delta^*(G) = \min_{L\in\Pi(G)} \Delta^+(G_L)$. That is $\Delta^*(G)$ is the smallest maximum out-degree in all the induced directions of G. In a directed graph $G = (V, E)$ for any vertex v we have,

- The set of *out-edges*, $E^+(v) = \{(v, u) \in E\}$
- The set of *in-edges* $E^-(v) = \{(u, v) \in E\}$

The Harmonious Strategy presented here has been modified to make it more like the Activation Strategy. As stated in Kierstead and Yang 2005 [13] the Harmonious Strategy involves tracking contributions between vertices. We have modified it so that we track activated edges. By doing this we can more clearly see that the Harmonious Strategy is an extension of the Activation Strategy. We summarise the main ideas of the Harmonious Strategy as follows.

As part of the Harmonious Strategy Alice keeps of track of a set of activated edges A. A edge is activated by adding it to A. Fix a linear ordering L on the vertices on a graph $G = (V, E)$ and denote the directed graph induced by L on G, G_L. suppose that Bob has just marked a vertex u. Alice plays the Harmonious Strategy by performing the following steps a times.

1. Alice selects, y, the least unmarked out-neighbor of u such that the edge $(u, y) \in E(G_L)$ has not yet been activated. She then activates the edge (u, y). If there is no such vertex then she marks the L-least unmarked vertex.
2. Alice repeats step (1) on u until she reaches at a vertex z that either has no unmarked out-neighbours or no unactivated out-edges.
3. Alice marks z

The Harmonious Strategy is used in the case the game is very asymmetric, that is $\Delta^*(G) \leq a/b$. In the case that $a/b < \Delta^*(G)$ the Harmonious Strategy is limited, as Alice may not be able to mark all the out-neighbours of a vertex. This means that there are graphs and linear orders that the strategy does not work on.

The Harmonious Strategy is used to find upper bounds for the classes of general graphs ($\mathcal{G}$), planar graphs ($\mathcal{P}$), and outerplanar graphs ($\mathcal{Q}$). Let $\mathcal{G}_k = \{G \in \mathcal{C} : \Delta^*(G) \leq k\}$. Some bounds found using the Harmonious Strategy are as follows.

- If $a < k$ then $\mathrm{col_g}(\mathcal{G}_k; a, b) = \infty$
- If $3 < a$ then $7 \leq \mathrm{col_g}(\mathcal{P}; a, 1) \leq 8$
- If $2 < a$ then $5 \leq \mathrm{col_g}(\mathcal{Q}; a, b) \leq 6$

More generally, we have the following bound for when Alice uses the Harmonious Strategy.

Theorem 3.24 (Kierstead and Yang 2005 [13])**.** *Let a, b be positive integers and G a graph with $\Delta^*(G) = k \leq \frac{a}{b}$. Then if Alice uses the Harmonious Strategy,*

$$\mathrm{col_g}(G; a, b) \leq 2k + b + 1$$

.

The proof presented here is a modified version of the original proof in [13]. We have modified the proof to coincide with our modified definition of the Harmonious Strategy.

Proof. Consider any time when Alice has just marked a vertex v and on Bob's previous turn he marked $x_0, \ldots, x_b$. Let u be an arbitrary unmarked vertex. We note, without proof, the following facts,

(1) Any unmarked vertex u has the same number of activated in-edges and activated out-edges.

(2) v has no unactivated out-edges.

(3) At the end of Alice's turn, every vertex x satisfies the following; all the vertices adjacent to an activated out-edge of x are marked.

If it still Alice's turn, then $x_0, \ldots, x_b$ may be adjacent to u and might not satisfy the facts (1), (2), and (3). If it is Bobs turn then he may be about to mark b vertices neighbouring u. Thus it suffices to show that u has at most $2k$ many marked neighbours other than $x_0, \ldots, x_b$.

By a combination of (2) and (3) for each marked in-neighbour y of u other than $x_0, \ldots, x_b$ the edge (y, u) is activated. By (1) the number of activated out-edges of u is the same as the number of marked in-neighbours. Thus, as u has at most k out-neighbours, u has at most $2k$ many activated neighbours other than $x_0, \ldots, x_b$. □

The bound we get when using Harmonious Strategy only works on some graphs and linear orders. This is a problem, but can be fixed. We could modify the strategy for a more general bound, one such example is the *Limited Harmonious Strategy* in [13]. Instead, we consider the Asymmetric Activation Strategy, which does not have this problem. The Asymmetric Activation Strategy was introduced by Yang and Zhu 2008 [21] as a strategy for the $(a, 1)$–marking game. This strategy is a combination of the Harmonious Strategy and the Activation Strategy.

Informally we summarise the Asymmetric Activation Strategy as follows. Fix a graph $G = (V, E)$ and a linear order L on V. Suppose that Bob has just marked a vertex u, Alice repeats the following steps a times to mark a vertices.

1. Alice activates u. If u has no out-neighbours then Alice marks the L-least vertex in V. Otherwise let $z = u$
2. Let v be the L-least out-neighbour of z. Alice activates v if v has been activated less than a times, otherwise she marks v.
3. If v has no out-neighbours then she marks v. Otherwise she lets $z = v$ and repeats 2 until she marks a vertex.

In summary, the primary difference between the original Activation Strategy and the Asymmetric Activation Strategy is that each vertex is activated a times before it is marked.

Algorithm 3.3 Asymmetric Activation Strategy

for i from 1 to a and $U \neq \emptyset$ **do**
  if $((N^+ \cap U \neq \emptyset) \wedge (t_u > 0))$ **then**
    $v \leftarrow \min_L(N^+(u) \cap U)$
    $t_u \leftarrow t_u - 1$
  else
    $v \leftarrow \min_L U$
  while $((N^+(v) \cap U \neq \emptyset) \wedge (t_v > 0))$ **do**
    $t \leftarrow \min_L(N^+(v) \cap U)$
    $t_v \leftarrow t_v - 1$
    $v \leftarrow t$
  $U \leftarrow U \setminus v$
  Mark the vertex v

Definition 3.25 (Asymmetric Activation Strategy, [21]). Let $G = (V, E)$ be a graph, U denote the set of unmarked vertices, and for all $v \in V$ let $t_v = a$, track the number of activations of v. If v has been activated a times then v will have $t_v = 0$. Let L be a linear order on V, and let G_L be the directed graph induced by L on G. For simplicity we consider the equivalent $(a, 1)$-marking game where Bob goes first and marks a vertex with no neighbours. Suppose that Bob has just activated a vertex u, Alice uses algorithm 3.3 to update the graph and mark vertices.

Consider a graph $G = (V, E)$ and a linear order L on V. We define a *loose out-neighbour* of a vertex v as a vertex u such that either $u \in N^+(v)$ or there exists a vertex z such that $u, v \in N^-(z)$ and $u <_L v$.

By using the Asymmetric Activation Strategy on the $(a, 1)$-marking game we get the following upper bound for $\mathrm{col_g}(G; a, 1)$.

Theorem 3.26 (Yang and Zhu 2008 [21]). *Fix a graph $G = (V, E)$ and a linear order L on V. Let G_L be the directed graph induced by the linear order L on the graph G such that $\Delta^+(G_L) = k > a$. Let r be the maximum number of loose out-neighbours of any vertex in V. If Alice uses the Asymmetric Activation Strategy then*

$$\mathrm{col_g}(G; a, 1) \leq k + \left\lfloor \left(1 + \frac{1}{a}\right) r \right\rfloor + 2$$

The following proof is the original from [21]. We present it here in its entirety, unified with our definitions.

Proof. Fix a graph $G = (V, E)$ and L a linear order on V such that $\Delta^+(G_L) = k > a$. Let r be the maximum number of loose out-neighbours of any vertex in V. Suppose that Bob has just marked a vertex x. Let M denote the current set of marked vertices and U the unmarked vertices. Let $u \in U$ be an arbitrary unmarked vertex. We need to show that u has no more than $k + \left\lfloor \left(1 + \frac{1}{a}\right) r \right\rfloor + 2$ marked neighbours other than x.

Let $S = (N(u) \cap M) \setminus x$. S is the set of marked neighbours of u excluding x. We partition S into two sets Q and R. $Q = N^+(u) \cap S$ is the set of marked out-neighbours of u excluding x. $R = N^-(u) \cap S$ is the set marked in-neighbours of u excluding x.

Notice that $|Q| \leq k = \Delta^+(G_L)$.

Fix $y \in R$. Either Alice marked y or Bob did. Alice only marks vertices with $t_y = 0$.

Suppose that Bob marked y. There are two times when we activate vertices, line 3 and line 9 in algorithm 3.3. At line 3, we activate the vertex that Bob has marked. We do this for each unmarked out-neighbour of y or until y has been activated a times. And, as every vertex has $k > a$ out-neighbours, x must have $t_y = 0$.

So, regardless of who marked y, $t_y = 0$.

Let $z = \min_L(N^+(y) \cap U)$, be the next vertex activated after y. Note that $u \in N^+(y) \cap U$ and so z is either a loose out-neighbour of u or $u = z$. Note that if u is activated immediately after y the next activated vertex is in $N^+(u)$, which is a set of loose out-neighbours of u. Hence the next activated vertex is a loose out-neighbour of u. Let n_u be the number of loose out-neigbours of u. Any vertex can be activated at most $a + 1$ many times. So we have

$$a|R| \leq (a + 1)n_u \leq (a + 1)r$$

Therefore,

$$|S| = |Q| + |R| \leq k + |R| \leq k + (1 + 1/a)r$$

□

3.3.5 Refined Activation Strategy

The Refined Activation Strategy is a refinement on the Activation Strategy introduced by Xuding Zhu in [23]. The Refined Activation Strategy is used to find the current best upper bound for the $(1,1)$-marking game for the class of planar graphs.

Theorem 3.27 (Theorem 1, Zhu 2008 [23])**.** *If G is a planar graph, then*

$$\mathrm{col_g}(G) \leq 17$$

We will not provide a full proof of this theorem, but rather will we describe the broad strokes behind the Refined Activation Strategy for a graph G.

The refined Activation Strategy applies the same basic strategy as the standard Activation Strategy but with two main differences. First, instead of a linear ordering on $V(G)$ we use directed graph, L_0, on $V(G)$ as our ordering. This is only a partial ordering, and so we partition $V(G)$ into blocks $B_1, B_2, \ldots, B_i$, where if $x \in B_i$, $y \in B_j$ and $i < j$ then the edge (x, y) is in L_0. The ordering may not be a linear ordering in each block. But, if we ignore what happens in each block we get a linear ordering between blocks.

The second change is the inclusion of a preference function ρ that maps each $y \in V(G)$ to a non-empty subset of the L_0-out-neigbours of y that are not in the same block as y. The preference function determines which unmarked out-neighbour of y is least. The preference function also determines if and when the edges in L_0 change direction. Suppose we have just activated a vertex v. Let w be the L_0-least out-neighbour of v. If there is a $u \in V(G)$ such that there is a directed edge $(u, w) \in L_0$ where w is in $\rho(u) \cap B_i$ for some i then we reverse the direction of the edge (u, v) in L_0. So, within the blocks the ordering changes as the marking game is played.

In summary the refined Activation Strategy is played as follows. During the strategy Alice tracks a set of activated vertices, A, and a preference function ρ.

1. Alice starts by marking the least v in L.
2. On her next turn let u be the last vertex marked by Bob. Alice starts at u *activates* it and moves to w the L_0-least unmarked neighbour of u determined by ρ.
3. Alice reverses an edge $(v, w) \in L_0$ as determined above.
4. If w is activated or has no unmarked neighbours then Alice marks w. If not Alice repeats step (2) on w until she either finds an active vertex.

Chapter 4

Conclusion

4.1 Summary

In this report we have studied a variety of different games and strategies. In our study of the dominating game we found both upper and lower bounds for the game dominating number. The upper bound was found by the way of a simple strategy for Alice. The lower bound was found by exploiting a property of tight bounds for the dominating number. For the class of trees we demonstrated two different strategies for Alice and their relationship with the 3/5-conjecture (conjecture 2.11). We concluded our study of the dominating game with a look at the (a, b) variant. In which we extended a previous result to get a new upper bound of for the (a, b)-dominating number. To conclude chapter 2 we studied the independent dominating game. While slightly shorter than the other sections, it touched on most areas in the literature.

Our look at the colouring game was divided into two main parts; lower bounds for the colouring game, and upper bounds by the way of the marking game. The lower bounds for which came virtue of some new extensions of previous results to the (a, b)-colouring game. The marking game is the largest section in this report. Most of this time was spent looking at the Activation Strategy and variations thereof. As part of our studies we provided two new proofs for graphs of bounded treewidth and pathwidth. We then showed how the activation strategy can be modified for the (a, b)-marking game. We concluded our study of the marking game with a brief look at the refined activation strategy.

4.2 A Java Implementation of The Colouring Game

As part of the background of, and to improve our understanding of, we implemented a version of the colouring game in Java [1]. This implementation took the form of an actual game played against the computer. The computer takes on the role of Alice and the user plays as Bob. The game opens by asking the user about the graph they wish to play on; namely, the number of vertices in the graph, how many colours are available, the width of the graph, and the type of graph (bounded treewidth or bounded pathwidth). A graph with the specified properties is then randomly generated. Once the game starts, the computer uses the activation strategy in an attempt to win the colouring game.

Applications of this game include as a study tool and a proper game. Because of the nature of the colouring game if the user does not give themselves enough colours the game is unwinnable. This unwinnable nature serves as a useful learning tool. By losing to the computer we are able to get a better understanding of the strategy as a whole. Conversely, an understanding of the activation strategy allows the user to win (assuming there are not too many colours). This is because the user can strategically pick vertices that force the computer to make certain moves. A desire to win would motivate the user to gain further understanding of the Activation Strategy. A useful property in a study tool.

This implementation could easily be extended into a full fledged application and released to the public. Such an extension would be a game that secretly teaches people mathematics, and more specify graph theory. There is potential here to include other strategies and other graph games. The other games are strategies studied in this report would make excellent additions. As an example, the Refined Activation Strategy for planar graphs would introduce a unique challenge to the game.

4.3 Other Games

We conclude this report by taking a brief look at some games that we would have loved to include, but did not have the time to. They are the total domination game, the perfect code game, and online colouring graphs of bounded pathwidth.

The *total domination game* is another variation of the domination game. Alice and Bob take turns building a total dominating set D in a graph $G = (V, E)$. A total dominating set is a dominating set D such that every vertex in D is adjacent to another vertex in D. So on their turn Alice and Bob add a vertex v to D such that $N[D]$ increase is size and v is adjacent to at least one vertex in $D \setminus v$. The game ends when D forms a dominating set. The total dominating game is a relatively recent creation. The introductory paper [9] was only published in 2015. As every total dominating set is a dominating set the total dominating game bounds the dominating game. This gives some relation between the two versions. However, the two game do differ in many ways. Exactly how the games differ is something we would have loved to include, but didn't have the time to.

A *perfect code* in a graph $G = (V, E)$ is an independent subset C of V such that every vertex in V is either in C or adjacent to exactly one vertex in C. In the *perfect code game* Alice and Bob take turns adding a vertex to C such that C forms a partial perfect code. The game continues until C forms a perfect code, or cannot be enlarged further. If at the end on the game C is a perfect code then Alice wins, and Bob wins if C is not a perfect code. Some graphs do not admit a perfect code. On such graphs Alice will never win. There are other graphs that admit perfect codes but on which Bob will always win. This raises the question, on which classes of graphs can Alice win? We can also ask, how small can Alice force the perfect code to be? These are the sorts of questions that could motivate further research.

When *online colouring* a graph, a single vertex is revealed along with its relation to all previously revealed vertices. The revealed vertex is then assigned a colour. How many colours do we need to online colour a graph? A graph of pathwidth k can be online coloured using $3k + 1$ colours [11]. We also have, when playing the colouring game on an interval graph with width k Alice will always win if the number of available colours is at least $3k + 1$ (theorem 3.19). It is quite a coincidence that these two concepts have the same bound on the same class of graphs. It gets even weirder when you consider the fact that the bound for the colouring game was not found using the colouring game. Exactly why these bounds are the same is unknown.

Bibliography

[1] Matthew Askes, *The Graph Colouring Game*, `https://github.com/askes5/graphColouringGame/releases/tag/1.0`, 2020, [Online].

[2] Hans L. Bodlaender, *Dynamic programming on graphs with bounded treewidth*, International Colloquium on Automata, Languages, and Programming, Springer, 1988, pp. 105–118.

[3] ———, *On the complexity of some coloring games*, Internat. J. Found. Comput. Sci. **2** (1991), no. 2, 133–147. MR 1143920

[4] ———, *A partial k-arboretum of graphs with bounded treewidth*, Theoret. Comput. Sci. **209** (1998), no. 1-2, 1–45. MR 1647486

[5] Boštjan Brešar, Sandi Klavžar, and Douglas F. Rall, *Domination game and an imagination strategy*, SIAM J. Discrete Math. **24** (2010), no. 3, 979–991. MR 2680228

[6] Csilla Bujtás, *Domination game on forests*, Discrete Math. **338** (2015), no. 12, 2220–2228. MR 3373324

[7] U. Faigle, U. Kern, H. A. Kierstead, and W. T. Trotter, *On the game chromatic number of some classes of graphs*, Ars Combin. **35** (1993), 143–150. MR 1220515

[8] Wayne Goddard and Michael A. Henning, *The competition-independence game in trees*, J. Combin. Math. Combin. Comput. **104** (2018), 161–170. MR 3753936

[9] Michael A. Henning, Sandi Klavžar, and Douglas F. Rall, *Total version of the domination game*, Graphs and Combinatorics **31** (2015), no. 5, 1453–1462.

[10] H. A. Kierstead, *A simple competitive graph coloring algorithm*, Journal of Combinatorial Theory, Series B **78** (2000), no. 1, 57–68.

[11] H. A. Kierstead and William T. Trotter, *An extremal problem in recursive combinatorics*, Congressus Numerantium **33** (1981), no. 143-153, 98.

[12] ———, *Planar graph coloring with an uncooperative partner*, Journal of Graph Theory **18** (1994), no. 6, 569–584.

[13] H. A. Kierstead and Daqing Yang, *Very asymmetric marking games*, Order **22** (2005), no. 2, 93–107 (2006). MR 2207192

[14] William B. Kinnersley, Douglas B. West, and Reza Zamani, *Extremal problems for game domination number*, SIAM J. Discrete Math. **27** (2013), no. 4, 2090–2107. MR 3143826

[15] Oystein Ore, *Theory of graphs (colloquium publications (amer mathematical soc))*, American Mathematical Society, dec 1962.

[16] James B. Phillips and Peter J. Slater, *An introduction to graph competition independence and enclaveless parameters*, Graph Theory Notes N. Y. **41** (2001), 37–41. MR 1869475

[17] Neil Robertson and Paul D. Seymour, *Graph minors. v. excluding a planar graph*, Journal of Combinatorial Theory, Series B **41** (1986), no. 1, 92–114.

[18] Robin Thomas, *An update on the four-color theorem*, Notices of the AMS **45** (1998), no. 7, 848–859.

[19] C. Worawannotai and W. Ruksasakchai, *Competition-independence game and domination game*, Mathematics **8** (2020), 359.

[20] Jiaojiao Wu and Xuding Zhu, *Lower bounds for the game colouring number of partial k-trees and planar graphs*, Discrete Math. **308** (2008), no. 12, 2637–2642. MR 2410475

[21] Daqing Yang and Xuding Zhu, *Activation strategy for asymmetric marking games*, European J. Combin. **29** (2008), no. 5, 1123–1132. MR 2419216

[22] Xuding Zhu, *The game coloring number of planar graphs*, J. Combin. Theory Ser. B **75** (1999), no. 2, 245–258. MR 1676891

[23] ———, *Refined activation strategy for the marking game*, J. Combin. Theory Ser. B **98** (2008), no. 1, 1–18. MR 2368019